

# GDCM

## 2.4.2

Generated by Doxygen 1.8.7

Fri Jul 18 2014 07:12:17



# Contents

<b>1</b>	<b>GDCM Documentation</b>	<b>1</b>
<b>2</b>	<b>off-screen rendering of DICOM images</b>	<b>3</b>
2.1	SYNOPSIS . . . . .	3
2.2	DESCRIPTION . . . . .	3
2.3	PARAMETERS . . . . .	3
2.4	OPTIONS . . . . .	3
2.4.1	OPTIONS . . . . .	3
2.4.2	general options . . . . .	3
2.5	Simple usage . . . . .	4
2.6	SEE ALSO . . . . .	4
2.7	COPYRIGHT . . . . .	4
<b>3</b>	<b>Convert a file supported by VTK into DICOM.</b>	<b>5</b>
3.1	SYNOPSIS . . . . .	5
3.2	DESCRIPTION . . . . .	5
3.3	PARAMETERS . . . . .	5
3.4	OPTIONS . . . . .	5
3.4.1	OPTIONS . . . . .	5
3.4.2	compression options . . . . .	6
3.4.3	general options . . . . .	6
3.4.4	environment variable . . . . .	6
3.5	DESCRIPTION . . . . .	6
3.5.1	CONVERT MetalImage (mhd, mha) . . . . .	6
3.5.2	CONVERT MHA/MHD . . . . .	7
3.5.3	CONVERT VTI . . . . .	7
3.5.4	CONVERT VTK . . . . .	7
3.6	CONVERT DICOM . . . . .	7
3.7	RoundTrip DICOM to MHD to DICOM . . . . .	7

3.8	gdcm2vtk notes	7
3.9	SEE ALSO	8
3.10	COPYRIGHT	8
<b>4</b>	<b>Tool to anonymize a DICOM file.</b>	<b>9</b>
4.1	SYNOPSIS	9
4.2	DESCRIPTION	9
4.3	PARAMETERS	9
4.4	OPTIONS	10
4.4.1	Required parameters	10
4.4.2	OPTIONS	10
4.4.3	encryption options	10
4.4.4	dumb mode options	10
4.4.5	general options	10
4.4.6	environment variable	11
4.5	Typical usage	11
4.5.1	De-identification (anonymization, encrypt)	11
4.5.2	Re-identification (de-anonymization,decrypt)	11
4.5.3	Multiple files caveat	11
4.5.4	Dumb mode	11
4.5.4.1	Irreversible Anonymization	12
4.6	OpenSSL	12
4.6.1	Generating a Private Key	12
4.6.2	Generating a Certificate	13
4.7	DICOM Standard:	13
4.8	Warnings	13
4.9	SEE ALSO	13
4.10	COPYRIGHT	13
<b>5</b>	<b>Tool to convert DICOM to DICOM.</b>	<b>15</b>
5.1	SYNOPSIS	15
5.2	DESCRIPTION	15
5.3	PARAMETERS	15
5.4	OPTIONS	15
5.4.1	PARAMETERS	15
5.4.2	OPTIONS	15
5.4.3	image options	16
5.4.4	JPEG options	16



5.4.5	JPEG-LS options	16
5.4.6	J2K options	16
5.4.7	general options	16
5.4.8	special options	16
5.4.9	environment variable	17
5.5	Simple usage	17
5.6	Typical usage	17
5.6.1	File Meta Header	17
5.6.2	Conversion to Explicit Transfer Syntax	18
5.6.3	Compressing to lossless JPEG	18
5.6.4	Compressing to lossy JPEG	18
5.6.5	Compressing to lossless JPEG-LS	18
5.6.6	Compressing to lossy JPEG-LS	18
5.6.7	Compressing to lossless J2K	18
5.6.8	Compressing to lossy J2K	18
5.6.9	Compressing to lossless RLE	19
5.6.10	Split encapsulated DICOM:	19
5.6.11	Forcing (re)compression	19
5.6.12	Decompressing a Compressed DICOM	19
5.6.13	Compressing an uncompressed Icon	19
5.6.14	Generating an Icon	20
5.6.15	Changing the planar Configuration	20
5.7	Lossless Conversion	20
5.8	Quality Control	20
5.8.1	DCMTK / dicom3tools	20
5.8.2	VIM: vimdiff	21
5.8.3	vbindiff	21
5.9	SEE ALSO	21
5.10	COPYRIGHT	21
<b>6</b>	<b>dumps differences of two DICOM files</b>	<b>23</b>
6.1	SYNOPSIS	23
6.2	DESCRIPTION	23
6.3	PARAMETERS	23
6.4	OPTIONS	23
6.4.1	OPTIONS	23
6.4.2	general options	23

6.5	Simple usage	24
6.6	SEE ALSO	24
6.7	COPYRIGHT	24
<b>7</b>	<b>dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.</b>	<b>25</b>
7.1	SYNOPSIS	25
7.2	DESCRIPTION	25
7.3	PARAMETERS	25
7.4	OPTIONS	25
7.4.1	OPTIONS	25
7.4.2	general options	26
7.4.3	special options	26
7.5	Typical usage	26
7.5.1	Printing Implicit Transfer Syntax	26
7.5.2	Print Private Attributes	27
7.5.3	SIEMENS CSA Header	27
7.5.4	GEMS Protocol Data Block	27
7.5.5	ELSCINT Protocol Information	28
7.5.6	VEPRO Protocol Information	28
7.5.7	Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)	29
7.5.8	Encapsulated ASN1 Structure	30
7.6	SEE ALSO	31
7.7	COPYRIGHT	31
<b>8</b>	<b>Tool to generate a DICOMDIR file from a File-Set.</b>	<b>33</b>
8.1	SYNOPSIS	33
8.2	DESCRIPTION	33
8.3	PARAMETERS	33
8.4	OPTIONS	33
8.4.1	Parameters	33
8.4.2	OPTIONS	33
8.4.3	general options	33
8.4.4	environment variable	34
8.5	Typical usage	34
8.6	NOTE	34
8.7	SEE ALSO	34
8.8	COPYRIGHT	34

<b>9 Manipulate DICOM image file.</b>	<b>35</b>
9.1 SYNOPSIS . . . . .	35
9.2 DESCRIPTION . . . . .	35
9.3 PARAMETERS . . . . .	35
9.4 OPTIONS . . . . .	35
9.4.1 PARAMETERS . . . . .	35
9.4.2 OPTIONS . . . . .	36
9.4.3 fill options . . . . .	36
9.4.4 general options . . . . .	36
9.4.5 environment variable . . . . .	36
9.5 Supported File Format (appropriate file extension) <code>gdcmimg</code> . . . . .	36
9.6 Typical usage . . . . .	37
9.6.1 Remove a rectangular part of the image . . . . .	37
9.6.2 Convert RAW to DICOM . . . . .	37
9.6.3 Convert PGM/PNM/PPM to DICOM . . . . .	38
9.6.4 Convert RLE to DICOM . . . . .	38
9.6.5 Convert JPEG to DICOM . . . . .	38
9.6.6 Convert J2K to DICOM . . . . .	38
9.6.7 Specifying a SOP Class UID . . . . .	38
9.7 Multiple Files . . . . .	38
9.8 Start Offset . . . . .	39
9.9 Warning . . . . .	39
9.10 SEE ALSO . . . . .	39
9.11 COPYRIGHT . . . . .	39
 <b>10 Display meta info about the input DICOM file.</b>	 <b>41</b>
10.1 SYNOPSIS . . . . .	41
10.2 DESCRIPTION . . . . .	41
10.3 PARAMETERS . . . . .	41
10.4 OPTIONS . . . . .	41
10.4.1 OPTIONS . . . . .	41
10.4.2 general options . . . . .	41
10.4.3 environment variable . . . . .	42
10.5 Simple usage . . . . .	42
10.5.1 <code>gdcmData</code> . . . . .	42
10.5.2 Davie Clunie datasets: . . . . .	42
10.5.3 Checking the md5sum of the Pixel Data . . . . .	43

10.5.4 Checking if Pixel Data is lossless . . . . .	43
10.6 SEE ALSO . . . . .	43
10.7 COPYRIGHT . . . . .	43
<b>11 Tool to convert PYPYRUS 3.0 to DICOM.</b>	<b>45</b>
11.1 SYNOPSIS . . . . .	45
11.2 DESCRIPTION . . . . .	45
11.3 PARAMETERS . . . . .	45
11.4 OPTIONS . . . . .	45
11.4.1 PARAMETERS . . . . .	45
11.4.2 OPTIONS . . . . .	45
11.4.3 general options . . . . .	45
11.4.4 environment variable . . . . .	46
11.5 Simple usage . . . . .	46
11.6 SEE ALSO . . . . .	46
11.7 COPYRIGHT . . . . .	46
<b>12 Tool to convert PDF to PDF/DICOM.</b>	<b>47</b>
12.1 SYNOPSIS . . . . .	47
12.2 DESCRIPTION . . . . .	47
12.3 PARAMETERS . . . . .	47
12.4 OPTIONS . . . . .	47
12.4.1 general options . . . . .	47
12.5 Usage Example . . . . .	48
12.6 PDF Info Mapping . . . . .	48
12.7 SEE ALSO . . . . .	49
12.8 COPYRIGHT . . . . .	49
<b>13 Extract Data Element Value Field.</b>	<b>51</b>
13.1 SYNOPSIS . . . . .	51
13.2 DESCRIPTION . . . . .	51
13.3 PARAMETERS . . . . .	51
13.4 OPTIONS . . . . .	51
13.4.1 PARAMETERS . . . . .	51
13.4.2 OPTIONS . . . . .	51
13.4.3 general options . . . . .	51
13.5 Typical usage . . . . .	52
13.5.1 Copy Attribute Value to file . . . . .	52

13.5.2 Extract Pixel Data . . . . .	52
13.5.3 Encapsulated Syntax . . . . .	52
13.5.4 Extract fragments as single file . . . . .	53
13.6 Footnote about JPEG files . . . . .	54
13.7 SEE ALSO . . . . .	54
13.8 COPYRIGHT . . . . .	54
<b>14 Scan a directory containing DICOM files.</b>	<b>55</b>
14.1 SYNOPSIS . . . . .	55
14.2 DESCRIPTION . . . . .	55
14.2.1 PARAMETERS . . . . .	55
14.2.2 OPTIONS . . . . .	55
14.2.3 general options . . . . .	55
14.3 Typical usage . . . . .	56
14.4 Simple usage . . . . .	56
14.5 Complex usage . . . . .	56
14.6 SEE ALSO . . . . .	56
14.7 COPYRIGHT . . . . .	56
<b>15 Tool to execute a DICOM Query/Retrieve operation</b>	<b>57</b>
15.1 SYNOPSIS . . . . .	57
15.2 DESCRIPTION . . . . .	57
15.3 PARAMETERS . . . . .	57
15.4 OPTIONS . . . . .	57
15.4.1 OPTIONS . . . . .	57
15.4.2 mode options . . . . .	57
15.4.3 C-STORE options . . . . .	58
15.4.4 C-FIND/C-MOVE options . . . . .	58
15.4.5 C-MOVE options . . . . .	58
15.4.6 general options . . . . .	58
15.4.7 environment variable . . . . .	58
15.5 C-ECHO usage . . . . .	59
15.6 C-STORE usage . . . . .	59
15.7 C-FIND usage . . . . .	59
15.8 C-MOVE usage . . . . .	60
15.9 patientroot notes . . . . .	60
15.10 Debugging . . . . .	60
15.11 Port Warning . . . . .	60

15.12C-STORE Warnings . . . . .	61
15.13C-MOVE Warnings . . . . .	61
15.14C-FIND IMAGE level (Composite Object Instance) . . . . .	61
15.15Storing the Query . . . . .	61
15.16DICOM Public Servers . . . . .	62
15.17SEE ALSO . . . . .	62
15.18COPYRIGHT . . . . .	62
<b>16 Concatenate/Extract DICOM files. . . . .</b>	<b>63</b>
16.1 SYNOPSIS . . . . .	63
16.2 DESCRIPTION . . . . .	63
16.3 PARAMETERS . . . . .	63
16.4 OPTIONS . . . . .	63
16.4.1 OPTIONS . . . . .	63
16.4.2 general options . . . . .	63
16.4.3 environment variable . . . . .	64
16.5 Typical usage . . . . .	64
16.5.1 SIEMENS Mosaic . . . . .	64
16.6 SEE ALSO . . . . .	65
16.7 COPYRIGHT . . . . .	65
<b>17 Simple DICOM viewer. . . . .</b>	<b>67</b>
17.1 SYNOPSIS . . . . .	67
17.2 DESCRIPTION . . . . .	67
17.3 PARAMETERS . . . . .	67
17.4 OPTIONS . . . . .	67
17.4.1 OPTIONS . . . . .	67
17.4.2 general options . . . . .	67
17.5 Typical usage . . . . .	68
17.6 Simple usage . . . . .	68
17.7 Wiki Link . . . . .	68
17.8 SEE ALSO . . . . .	68
17.9 COPYRIGHT . . . . .	68
<b>18 provides a tool to convert a DICOM file into a XML infoset and vice-versa. . . . .</b>	<b>69</b>
18.1 SYNOPSIS . . . . .	69
18.2 DESCRIPTION . . . . .	69
18.3 PARAMETERS . . . . .	69

18.4	OPTIONS	69
18.4.1	PARAMETERS	69
18.4.2	Options for DICOM to XML:	69
18.4.3	Options for XML to DICOM:	70
18.4.4	general options	70
18.5	SEE ALSO	70
18.6	COPYRIGHT	70
<b>19</b>	<b>Todo List</b>	<b>71</b>
<b>20</b>	<b>Deprecated List</b>	<b>73</b>
<b>21</b>	<b>Bug List</b>	<b>75</b>
<b>22</b>	<b>Namespace Index</b>	<b>77</b>
22.1	Namespace List	77
<b>23</b>	<b>Hierarchical Index</b>	<b>79</b>
23.1	Class Hierarchy	79
<b>24</b>	<b>Class Index</b>	<b>87</b>
24.1	Class List	87
<b>25</b>	<b>File Index</b>	<b>103</b>
25.1	File List	103
<b>26</b>	<b>Namespace Documentation</b>	<b>111</b>
26.1	gdcm Namespace Reference	111
26.1.1	Detailed Description	125
26.1.2	Typedef Documentation	126
26.1.2.1	AEComp	126
26.1.2.2	ASComp	126
26.1.2.3	BOOL_FUNCTION_PFILE_PFILE_POINTER	126
26.1.2.4	CSComp	126
26.1.2.5	DAComp	126
26.1.2.6	DTComp	126
26.1.2.7	FileList	126
26.1.2.8	IconImage	126
26.1.2.9	LOComp	126
26.1.2.10	LTComp	126

26.1.2.11 MacroEntry . . . . .	126
26.1.2.12 NestedMacroEntries . . . . .	126
26.1.2.13 PNComp . . . . .	126
26.1.2.14 SHComp . . . . .	126
26.1.2.15 STComp . . . . .	126
26.1.2.16 TMComp . . . . .	126
26.1.2.17 UIComp . . . . .	126
26.1.2.18 UTComp . . . . .	126
26.1.3 Enumeration Type Documentation . . . . .	126
26.1.3.1 CompOperators . . . . .	126
26.1.3.2 ECharSet . . . . .	127
26.1.3.3 EQueryLevel . . . . .	127
26.1.3.4 EQueryType . . . . .	127
26.1.3.5 ERootType . . . . .	128
26.1.3.6 LodModeType . . . . .	128
26.1.4 Function Documentation . . . . .	128
26.1.4.1 backslash . . . . .	128
26.1.4.2 GetVRFromTag . . . . .	128
26.1.4.3 operator"! = . . . . .	128
26.1.4.4 operator"! = . . . . .	128
26.1.4.5 operator<< . . . . .	128
26.1.4.6 operator<< . . . . .	128
26.1.4.7 operator<< . . . . .	128
26.1.4.8 operator<< . . . . .	128
26.1.4.9 operator<< . . . . .	128
26.1.4.10 operator<< . . . . .	129
26.1.4.11 operator<< . . . . .	129
26.1.4.12 operator<< . . . . .	129
26.1.4.13 operator<< . . . . .	129
26.1.4.14 operator<< . . . . .	129
26.1.4.15 operator<< . . . . .	129
26.1.4.16 operator<< . . . . .	129
26.1.4.17 operator<< . . . . .	129
26.1.4.18 operator<< . . . . .	129
26.1.4.19 operator<< . . . . .	129
26.1.4.20 operator<< . . . . .	129
26.1.4.21 operator<< . . . . .	129



26.1.4.22 operator<<	129
26.1.4.23 operator<<	129
26.1.4.24 operator<<	130
26.1.4.25 operator<<	130
26.1.4.26 operator<<	130
26.1.4.27 operator<<	130
26.1.4.28 operator<<	130
26.1.4.29 operator<<	130
26.1.4.30 operator<<	130
26.1.4.31 operator<<	130
26.1.4.32 operator<<	130
26.1.4.33 operator<<	130
26.1.4.34 operator<<	130
26.1.4.35 operator<<	130
26.1.4.36 operator<<	130
26.1.4.37 operator<<	130
26.1.4.38 operator<<	130
26.1.4.39 operator<<	130
26.1.4.40 operator<<	130
26.1.4.41 operator<<	131
26.1.4.42 operator<<	131
26.1.4.43 operator<<	131
26.1.4.44 operator<<	131
26.1.4.45 operator<<	131
26.1.4.46 operator<<	131
26.1.4.47 operator<<	131
26.1.4.48 operator<<	131
26.1.4.49 operator<<	131
26.1.4.50 operator<<	131
26.1.4.51 operator<<	131
26.1.4.52 operator<<	132
26.1.4.53 operator<<	132
26.1.4.54 operator<<	132
26.1.4.55 operator<<	132
26.1.4.56 operator<<	132
26.1.4.57 operator<<	132
26.1.4.58 operator<<	132

26.1.4.59 operator==	132
26.1.4.60 operator>>	132
26.1.4.61 operator>>	132
26.1.4.62 operator>>	132
26.1.4.63 to_string	132
26.1.4.64 TYPETOENCODING	133
26.1.5 Variable Documentation	133
26.1.5.1 GlobalInstance	133
26.1.5.2 VRINARY	133
26.2 gdcm::network Namespace Reference	133
26.2.1 Enumeration Type Documentation	137
26.2.1.1 EEventID	137
26.2.1.2 EStateID	137
26.2.2 Function Documentation	138
26.2.2.1 GetStateIndex	138
26.2.3 Variable Documentation	138
26.2.3.1 cMaxEventID	138
26.2.3.2 cMaxStateID	138
26.3 gdcm::SegmentHelper Namespace Reference	138
26.4 gdcm::terminal Namespace Reference	138
26.4.1 Detailed Description	139
26.4.2 Enumeration Type Documentation	139
26.4.2.1 Attribute	139
26.4.2.2 Color	139
26.4.2.3 Mode	140
26.4.3 Function Documentation	140
26.4.3.1 setattribute	140
26.4.3.2 setbgcolor	140
26.4.3.3 setfgcolor	140
26.4.3.4 setmode	140
<b>27 Class Documentation</b>	<b>141</b>
27.1 gdcm::network::AAbortPDU Class Reference	141
27.1.1 Detailed Description	142
27.1.2 Constructor & Destructor Documentation	142
27.1.2.1 AAbortPDU	142
27.1.3 Member Function Documentation	142

27.1.3.1	IsLastFragment	142
27.1.3.2	Print	142
27.1.3.3	Read	142
27.1.3.4	SetReason	143
27.1.3.5	SetSource	143
27.1.3.6	Size	143
27.1.3.7	Write	143
27.2	gdcmm::network::AAssociateACPDU Class Reference	143
27.2.1	Detailed Description	144
27.2.2	Member Typedef Documentation	145
27.2.2.1	SizeType	145
27.2.3	Constructor & Destructor Documentation	145
27.2.3.1	AAssociateACPDU	145
27.2.4	Member Function Documentation	145
27.2.4.1	AddPresentationContextAC	145
27.2.4.2	GetNumberOfPresentationContextAC	145
27.2.4.3	GetPresentationContextAC	145
27.2.4.4	GetUserInformation	145
27.2.4.5	InitFromRQ	145
27.2.4.6	IsLastFragment	145
27.2.4.7	Print	145
27.2.4.8	Read	145
27.2.4.9	SetCalledAETitle	145
27.2.4.10	SetCallingAETitle	145
27.2.4.11	Size	145
27.2.4.12	Write	145
27.2.5	Friends And Related Function Documentation	145
27.2.5.1	AAssociateRQPDU	146
27.3	gdcmm::network::AAssociateRJPDU Class Reference	146
27.3.1	Detailed Description	147
27.3.2	Constructor & Destructor Documentation	147
27.3.2.1	AAssociateRJPDU	147
27.3.3	Member Function Documentation	147
27.3.3.1	IsLastFragment	147
27.3.3.2	Print	147
27.3.3.3	Read	147
27.3.3.4	Size	147

27.3.3.5	Write	147
27.4	gdcm::network::AAssociateRQPDU Class Reference	147
27.4.1	Detailed Description	149
27.4.2	Member Typedef Documentation	149
27.4.2.1	PresentationContextArrayType	149
27.4.2.2	SizeType	149
27.4.3	Constructor & Destructor Documentation	149
27.4.3.1	AAssociateRQPDU	149
27.4.3.2	AAssociateRQPDU	149
27.4.4	Member Function Documentation	150
27.4.4.1	AddPresentationContext	150
27.4.4.2	GetCalledAETitle	150
27.4.4.3	GetCallingAETitle	150
27.4.4.4	GetNumberOfPresentationContext	150
27.4.4.5	GetPresentationContext	150
27.4.4.6	GetPresentationContextByAbstractSyntax	150
27.4.4.7	GetPresentationContextByID	150
27.4.4.8	GetPresentationContexts	150
27.4.4.9	GetReserved43_74	150
27.4.4.10	GetUserInformation	150
27.4.4.11	IsAETitleValid	150
27.4.4.12	IsLastFragment	150
27.4.4.13	Print	150
27.4.4.14	Read	150
27.4.4.15	SetCalledAETitle	150
27.4.4.16	SetCallingAETitle	151
27.4.4.17	SetUserInformation	151
27.4.4.18	Size	151
27.4.4.19	Write	151
27.4.5	Friends And Related Function Documentation	151
27.4.5.1	AAssociateACPDU	151
27.5	gdcm::AbortEvent Class Reference	151
27.6	gdcm::network::AbstractSyntax Class Reference	152
27.6.1	Detailed Description	152
27.6.2	Constructor & Destructor Documentation	153
27.6.2.1	AbstractSyntax	153
27.6.3	Member Function Documentation	153

27.6.3.1	<a href="#">GetAsDataElement</a>	153
27.6.3.2	<a href="#">GetName</a>	153
27.6.3.3	<a href="#">operator==</a>	153
27.6.3.4	<a href="#">Print</a>	153
27.6.3.5	<a href="#">Read</a>	153
27.6.3.6	<a href="#">SetName</a>	153
27.6.3.7	<a href="#">SetNameFromUID</a>	153
27.6.3.8	<a href="#">Size</a>	153
27.6.3.9	<a href="#">Write</a>	153
27.7	<a href="#">gdcm::AnonymizeEvent Class Reference</a>	153
27.7.1	<a href="#">Detailed Description</a>	155
27.7.2	<a href="#">Member Typedef Documentation</a>	155
27.7.2.1	<a href="#">Self</a>	155
27.7.2.2	<a href="#">Superclass</a>	155
27.7.3	<a href="#">Constructor &amp; Destructor Documentation</a>	155
27.7.3.1	<a href="#">AnonymizeEvent</a>	155
27.7.3.2	<a href="#">~AnonymizeEvent</a>	155
27.7.3.3	<a href="#">AnonymizeEvent</a>	155
27.7.4	<a href="#">Member Function Documentation</a>	155
27.7.4.1	<a href="#">CheckEvent</a>	155
27.7.4.2	<a href="#">GetEventName</a>	155
27.7.4.3	<a href="#">GetTag</a>	155
27.7.4.4	<a href="#">MakeObject</a>	155
27.7.4.5	<a href="#">SetTag</a>	155
27.8	<a href="#">gdcm::Anonymizer Class Reference</a>	156
27.8.1	<a href="#">Detailed Description</a>	157
27.8.2	<a href="#">Constructor &amp; Destructor Documentation</a>	158
27.8.2.1	<a href="#">Anonymizer</a>	158
27.8.2.2	<a href="#">~Anonymizer</a>	158
27.8.3	<a href="#">Member Function Documentation</a>	158
27.8.3.1	<a href="#">BALCPPProtect</a>	159
27.8.3.2	<a href="#">BasicApplicationLevelConfidentialityProfile</a>	159
27.8.3.3	<a href="#">CanEmptyTag</a>	159
27.8.3.4	<a href="#">ClearInternalUIDs</a>	159
27.8.3.5	<a href="#">Empty</a>	159
27.8.3.6	<a href="#">GetBasicApplicationLevelConfidentialityProfileAttributes</a>	159
27.8.3.7	<a href="#">GetCryptographicMessageSyntax</a>	159

27.8.3.8	<a href="#">GetFile</a>	159
27.8.3.9	<a href="#">New</a>	159
27.8.3.10	<a href="#">RecurseDataSet</a>	159
27.8.3.11	<a href="#">Remove</a>	159
27.8.3.12	<a href="#">RemoveGroupLength</a>	160
27.8.3.13	<a href="#">RemovePrivateTags</a>	160
27.8.3.14	<a href="#">RemoveRetired</a>	160
27.8.3.15	<a href="#">Replace</a>	160
27.8.3.16	<a href="#">Replace</a>	160
27.8.3.17	<a href="#">SetCryptographicMessageSyntax</a>	160
27.8.3.18	<a href="#">SetFile</a>	160
27.9	<a href="#">gdcm::AnyEvent Class Reference</a>	161
27.10	<a href="#">gdcm::network::ApplicationContext Class Reference</a>	162
27.10.1	<a href="#">Detailed Description</a>	162
27.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	163
27.10.2.1	<a href="#">ApplicationContext</a>	163
27.10.3	<a href="#">Member Function Documentation</a>	163
27.10.3.1	<a href="#">GetName</a>	163
27.10.3.2	<a href="#">Print</a>	163
27.10.3.3	<a href="#">Read</a>	163
27.10.3.4	<a href="#">SetName</a>	163
27.10.3.5	<a href="#">Size</a>	163
27.10.3.6	<a href="#">Write</a>	163
27.11	<a href="#">gdcm::ApplicationEntity Class Reference</a>	163
27.11.1	<a href="#">Detailed Description</a>	164
27.11.2	<a href="#">Member Function Documentation</a>	164
27.11.2.1	<a href="#">IsValid</a>	164
27.11.2.2	<a href="#">Print</a>	164
27.11.2.3	<a href="#">SetBlob</a>	164
27.11.2.4	<a href="#">Squeeze</a>	164
27.11.3	<a href="#">Member Data Documentation</a>	164
27.11.3.1	<a href="#">Internal</a>	164
27.11.3.2	<a href="#">MaxLength</a>	164
27.11.3.3	<a href="#">MaxNumberOfComponents</a>	164
27.11.3.4	<a href="#">Padding</a>	165
27.11.3.5	<a href="#">Separator</a>	165
27.12	<a href="#">gdcm::network::AReleaseRPPDU Class Reference</a>	165

27.12.1 Detailed Description	166
27.12.2 Constructor & Destructor Documentation	166
27.12.2.1 AReleaseRPPDU	166
27.12.3 Member Function Documentation	166
27.12.3.1 IsLastFragment	166
27.12.3.2 Print	166
27.12.3.3 Read	166
27.12.3.4 Size	166
27.12.3.5 Write	166
27.13gdcmm::network::AReleaseRQPDU Class Reference	166
27.13.1 Detailed Description	167
27.13.2 Constructor & Destructor Documentation	167
27.13.2.1 AReleaseRQPDU	168
27.13.3 Member Function Documentation	168
27.13.3.1 IsLastFragment	168
27.13.3.2 Print	168
27.13.3.3 Read	168
27.13.3.4 Size	168
27.13.3.5 Write	168
27.14gdcmm::network::ARTIMTimer Class Reference	168
27.14.1 Detailed Description	169
27.14.2 Constructor & Destructor Documentation	169
27.14.2.1 ARTIMTimer	169
27.14.3 Member Function Documentation	169
27.14.3.1 GetElapsedTime	169
27.14.3.2 GetHasExpired	169
27.14.3.3 GetTimeout	169
27.14.3.4 SetTimeout	169
27.14.3.5 Start	169
27.14.3.6 Stop	169
27.15gdcmm::ASN1 Class Reference	169
27.15.1 Detailed Description	170
27.15.2 Constructor & Destructor Documentation	170
27.15.2.1 ASN1	170
27.15.2.2 ~ASN1	170
27.15.3 Member Function Documentation	170
27.15.3.1 ParseDump	170

27.15.3.2 ParseDumpFile . . . . .	170
27.15.3.3 TestPBKDF2 . . . . .	170
27.16gdcmm::network::AsynchronousOperationsWindowSub Class Reference . . . . .	170
27.16.1 Detailed Description . . . . .	170
27.16.2 Constructor & Destructor Documentation . . . . .	171
27.16.2.1 AsynchronousOperationsWindowSub . . . . .	171
27.16.3 Member Function Documentation . . . . .	171
27.16.3.1 Print . . . . .	171
27.16.3.2 Read . . . . .	171
27.16.3.3 Size . . . . .	171
27.16.3.4 Write . . . . .	171
27.17gdcmm::Attribute< Group, Element, TVR, TVM > Class Template Reference . . . . .	171
27.17.1 Detailed Description . . . . .	173
27.17.2 Member Typedef Documentation . . . . .	173
27.17.2.1 ArrayType . . . . .	173
27.17.3 Member Enumeration Documentation . . . . .	173
27.17.3.1 anonymous enum . . . . .	173
27.17.4 Member Function Documentation . . . . .	173
27.17.4.1 GDCM_STATIC_ASSERT . . . . .	173
27.17.4.2 GDCM_STATIC_ASSERT . . . . .	173
27.17.4.3 GDCM_STATIC_ASSERT . . . . .	173
27.17.4.4 GetAsDataElement . . . . .	174
27.17.4.5 GetDictVM . . . . .	174
27.17.4.6 GetDictVR . . . . .	174
27.17.4.7 GetNumberOfValues . . . . .	174
27.17.4.8 GetTag . . . . .	174
27.17.4.9 GetValue . . . . .	175
27.17.4.10GetValue . . . . .	175
27.17.4.11GetValues . . . . .	175
27.17.4.12GetVM . . . . .	175
27.17.4.13GetVR . . . . .	175
27.17.4.14operator"!=" . . . . .	175
27.17.4.15operator< . . . . .	176
27.17.4.16operator== . . . . .	176
27.17.4.17operator[] . . . . .	176
27.17.4.18operator[] . . . . .	176
27.17.4.19Print . . . . .	176



27.17.4.20Set	176
27.17.4.21SetByteValue	176
27.17.4.22SetByteValueNoSwap	177
27.17.4.23SetFromDataElement	177
27.17.4.24SetFromDataSet	177
27.17.4.25SetValue	177
27.17.4.26SetValues	177
27.17.5 Member Data Documentation	178
27.17.5.1 Internal	178
27.18gdcmm::Attribute< Group, Element, TVR, VM::VM1 > Class Template Reference	178
27.18.1 Member Typedef Documentation	180
27.18.1.1 ArrayType	180
27.18.2 Member Enumeration Documentation	180
27.18.2.1 anonymous enum	180
27.18.3 Member Function Documentation	180
27.18.3.1 GDCM_STATIC_ASSERT	180
27.18.3.2 GDCM_STATIC_ASSERT	180
27.18.3.3 GDCM_STATIC_ASSERT	180
27.18.3.4 GDCM_STATIC_ASSERT	180
27.18.3.5 GetAsDataElement	180
27.18.3.6 GetDictVM	181
27.18.3.7 GetDictVR	181
27.18.3.8 GetNumberOfValues	181
27.18.3.9 GetTag	181
27.18.3.10GetValue	181
27.18.3.11GetValue	181
27.18.3.12GetValues	181
27.18.3.13GetVM	181
27.18.3.14GetVR	181
27.18.3.15operator!=	181
27.18.3.16operator<	181
27.18.3.17operator==	182
27.18.3.18Print	182
27.18.3.19Set	182
27.18.3.20SetByteValue	182
27.18.3.21SetByteValueNoSwap	182
27.18.3.22SetFromDataElement	182

27.18.3.23SetFromDataSet . . . . .	182
27.18.3.24SetValue . . . . .	182
27.18.4 Member Data Documentation . . . . .	183
27.18.4.1 Internal . . . . .	183
27.19gdcmm::Attribute< Group, Element, TVR, VM::VM1_3 > Class Template Reference . . . . .	183
27.19.1 Member Function Documentation . . . . .	184
27.19.1.1 GetVM . . . . .	184
27.20gdcmm::Attribute< Group, Element, TVR, VM::VM1_8 > Class Template Reference . . . . .	184
27.20.1 Member Function Documentation . . . . .	185
27.20.1.1 GetVM . . . . .	185
27.21gdcmm::Attribute< Group, Element, TVR, VM::VM1_n > Class Template Reference . . . . .	185
27.21.1 Member Typedef Documentation . . . . .	187
27.21.1.1 ArrayType . . . . .	187
27.21.2 Constructor & Destructor Documentation . . . . .	187
27.21.2.1 Attribute . . . . .	187
27.21.2.2 ~Attribute . . . . .	187
27.21.3 Member Function Documentation . . . . .	187
27.21.3.1 GDCM_STATIC_ASSERT . . . . .	187
27.21.3.2 GDCM_STATIC_ASSERT . . . . .	187
27.21.3.3 GDCM_STATIC_ASSERT . . . . .	187
27.21.3.4 GetAsDataElement . . . . .	187
27.21.3.5 GetDictVM . . . . .	187
27.21.3.6 GetDictVR . . . . .	187
27.21.3.7 GetNumberOfValues . . . . .	188
27.21.3.8 GetTag . . . . .	188
27.21.3.9 GetValue . . . . .	188
27.21.3.10GetValue . . . . .	188
27.21.3.11GetValues . . . . .	188
27.21.3.12GetVM . . . . .	188
27.21.3.13GetVR . . . . .	188
27.21.3.14operator[] . . . . .	188
27.21.3.15operator[] . . . . .	188
27.21.3.16Print . . . . .	188
27.21.3.17Set . . . . .	189
27.21.3.18SetByteValue . . . . .	189
27.21.3.19SetFromDataElement . . . . .	189
27.21.3.20SetFromDataSet . . . . .	189

27.21.3.21SetNumberOfValues . . . . .	189
27.21.3.22SetValue . . . . .	189
27.21.3.23SetValue . . . . .	189
27.21.3.24SetValues . . . . .	190
27.22gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . . .	190
27.22.1 Member Function Documentation . . . . .	191
27.22.1.1 GetVM . . . . .	191
27.23gdcmm::Attribute< Group, Element, TVR, VM::VM2_n > Class Template Reference . . . . .	191
27.23.1 Member Function Documentation . . . . .	192
27.23.1.1 GetVM . . . . .	193
27.24gdcmm::Attribute< Group, Element, TVR, VM::VM3_3n > Class Template Reference . . . . .	193
27.24.1 Member Function Documentation . . . . .	194
27.24.1.1 GetVM . . . . .	194
27.25gdcmm::Attribute< Group, Element, TVR, VM::VM3_n > Class Template Reference . . . . .	194
27.25.1 Member Function Documentation . . . . .	195
27.25.1.1 GetVM . . . . .	196
27.26gdcmm::AudioCodec Class Reference . . . . .	196
27.26.1 Detailed Description . . . . .	197
27.26.2 Constructor & Destructor Documentation . . . . .	197
27.26.2.1 AudioCodec . . . . .	197
27.26.2.2 ~AudioCodec . . . . .	197
27.26.3 Member Function Documentation . . . . .	197
27.26.3.1 CanCode . . . . .	197
27.26.3.2 CanDecode . . . . .	198
27.26.3.3 Decode . . . . .	198
27.27gdcmm::Base64 Class Reference . . . . .	198
27.27.1 Detailed Description . . . . .	198
27.27.2 Member Function Documentation . . . . .	198
27.27.2.1 Decode . . . . .	198
27.27.2.2 Encode . . . . .	199
27.27.2.3 GetDecodeLength . . . . .	199
27.27.2.4 GetEncodeLength . . . . .	199
27.28gdcmm::network::BaseCompositeMessage Class Reference . . . . .	199
27.28.1 Detailed Description . . . . .	200
27.28.2 Constructor & Destructor Documentation . . . . .	201
27.28.2.1 ~BaseCompositeMessage . . . . .	201
27.28.3 Member Function Documentation . . . . .	201

27.28.3.1 ConstructPDV . . . . .	201
27.29gdcmm::network::BasePDU Class Reference . . . . .	201
27.29.1 Detailed Description . . . . .	202
27.29.2 Constructor & Destructor Documentation . . . . .	203
27.29.2.1 ~BasePDU . . . . .	203
27.29.3 Member Function Documentation . . . . .	203
27.29.3.1 IsLastFragment . . . . .	203
27.29.3.2 Print . . . . .	203
27.29.3.3 Read . . . . .	203
27.29.3.4 Size . . . . .	203
27.29.3.5 Write . . . . .	203
27.30gdcmm::BaseRootQuery Class Reference . . . . .	203
27.30.1 Detailed Description . . . . .	205
27.30.2 Constructor & Destructor Documentation . . . . .	205
27.30.2.1 BaseRootQuery . . . . .	205
27.30.2.2 ~BaseRootQuery . . . . .	205
27.30.3 Member Function Documentation . . . . .	206
27.30.3.1 AddQueryDataSet . . . . .	206
27.30.3.2 Construct . . . . .	206
27.30.3.3 GetAbstractSyntaxUID . . . . .	206
27.30.3.4 GetQueryDataSet . . . . .	206
27.30.3.5 GetQueryDataSet . . . . .	206
27.30.3.6 GetQueryLevelFromQueryRoot . . . . .	206
27.30.3.7 GetQueryLevelFromString . . . . .	206
27.30.3.8 GetQueryLevelString . . . . .	206
27.30.3.9 GetTagListByLevel . . . . .	206
27.30.3.10InitializeDataSet . . . . .	206
27.30.3.11Print . . . . .	206
27.30.3.12SetSearchParameter . . . . .	206
27.30.3.13SetSearchParameter . . . . .	206
27.30.3.14SetSearchParameter . . . . .	207
27.30.3.15ValidateQuery . . . . .	207
27.30.3.16WriteHelpFile . . . . .	207
27.30.3.17WriteQuery . . . . .	207
27.30.4 Friends And Related Function Documentation . . . . .	207
27.30.4.1 QueryFactory . . . . .	207
27.30.5 Member Data Documentation . . . . .	207

27.30.5.1 mDataSet . . . . .	207
27.30.5.2 mHelpDescription . . . . .	207
27.30.5.3 mImage . . . . .	207
27.30.5.4 mPatient . . . . .	207
27.30.5.5 mRootType . . . . .	207
27.30.5.6 mSeries . . . . .	207
27.30.5.7 mStudy . . . . .	207
27.31gdcm::SegmentHelper::BasicCodedEntry Struct Reference . . . . .	207
27.31.1 Detailed Description . . . . .	209
27.31.2 Constructor & Destructor Documentation . . . . .	209
27.31.2.1 BasicCodedEntry . . . . .	209
27.31.2.2 BasicCodedEntry . . . . .	209
27.31.2.3 BasicCodedEntry . . . . .	209
27.31.3 Member Function Documentation . . . . .	209
27.31.3.1 IsEmpty . . . . .	209
27.31.4 Member Data Documentation . . . . .	209
27.31.4.1 CM . . . . .	209
27.31.4.2 CSD . . . . .	209
27.31.4.3 CSV . . . . .	209
27.31.4.4 CV . . . . .	210
27.32gdcm::BasicOffsetTable Class Reference . . . . .	210
27.32.1 Detailed Description . . . . .	211
27.32.2 Constructor & Destructor Documentation . . . . .	211
27.32.2.1 BasicOffsetTable . . . . .	211
27.32.3 Member Function Documentation . . . . .	212
27.32.3.1 Read . . . . .	212
27.32.4 Friends And Related Function Documentation . . . . .	212
27.32.4.1 operator<< . . . . .	212
27.33gdcm::Bitmap Class Reference . . . . .	212
27.33.1 Detailed Description . . . . .	215
27.33.2 Member Typedef Documentation . . . . .	215
27.33.2.1 LUTPtr . . . . .	215
27.33.3 Constructor & Destructor Documentation . . . . .	215
27.33.3.1 Bitmap . . . . .	215
27.33.3.2 ~Bitmap . . . . .	215
27.33.4 Member Function Documentation . . . . .	215
27.33.4.1 AreOverlaysInPixelData . . . . .	215

27.33.4.2 Clear	215
27.33.4.3 ComputeLossyFlag	215
27.33.4.4 GetBuffer	215
27.33.4.5 GetBuffer2	215
27.33.4.6 GetBufferLength	215
27.33.4.7 GetColumns	216
27.33.4.8 GetDataElement	216
27.33.4.9 GetDataElement	216
27.33.4.10 GetDimension	216
27.33.4.11 GetDimensions	216
27.33.4.12 GetLUT	216
27.33.4.13 GetLUT	216
27.33.4.14 GetNeedByteSwap	216
27.33.4.15 GetNumberOfDimensions	216
27.33.4.16 GetPhotometricInterpretation	216
27.33.4.17 GetPixelFormat	217
27.33.4.18 GetPixelFormat	217
27.33.4.19 GetPlanarConfiguration	217
27.33.4.20 GetRows	217
27.33.4.21 GetTransferSyntax	217
27.33.4.22 IsEmpty	217
27.33.4.23 IsLossy	217
27.33.4.24 IsTransferSyntaxCompatible	217
27.33.4.25 Print	217
27.33.4.26 SetColumns	217
27.33.4.27 SetDataElement	217
27.33.4.28 SetDimension	218
27.33.4.29 SetDimensions	218
27.33.4.30 SetLossyFlag	218
27.33.4.31 SetLUT	218
27.33.4.32 SetNeedByteSwap	218
27.33.4.33 SetNumberOfDimensions	218
27.33.4.34 SetPhotometricInterpretation	218
27.33.4.35 SetPixelFormat	218
27.33.4.36 SetPlanarConfiguration	218
27.33.4.37 SetRows	219
27.33.4.38 SetTransferSyntax	219

27.33.4.39TryJPEG2000Codec . . . . .	219
27.33.4.40TryJPEG2000Codec2 . . . . .	219
27.33.4.41TryJPEGCodec . . . . .	219
27.33.4.42TryJPEGCodec2 . . . . .	219
27.33.4.43TryJPEGLSCodec . . . . .	219
27.33.4.44TryKAKADUCodec . . . . .	219
27.33.4.45TryPVRGCodec . . . . .	219
27.33.4.46TryRAWCodec . . . . .	219
27.33.4.47TryRLECodec . . . . .	219
27.33.5 Friends And Related Function Documentation . . . . .	219
27.33.5.1 ImageChangeTransferSyntax . . . . .	219
27.33.5.2 PixmapReader . . . . .	219
27.33.6 Member Data Documentation . . . . .	219
27.33.6.1 Dimensions . . . . .	219
27.33.6.2 LossyFlag . . . . .	219
27.33.6.3 LUT . . . . .	219
27.33.6.4 NeedByteSwap . . . . .	219
27.33.6.5 NumberOfDimensions . . . . .	219
27.33.6.6 PF . . . . .	219
27.33.6.7 PI . . . . .	219
27.33.6.8 PixelData . . . . .	219
27.33.6.9 PlanarConfiguration . . . . .	220
27.33.6.10TS . . . . .	220
27.34gdcm::BitmapToBitmapFilter Class Reference . . . . .	220
27.34.1 Detailed Description . . . . .	221
27.34.2 Constructor & Destructor Documentation . . . . .	221
27.34.2.1 BitmapToBitmapFilter . . . . .	221
27.34.2.2 ~BitmapToBitmapFilter . . . . .	222
27.34.3 Member Function Documentation . . . . .	222
27.34.3.1 GetOutput . . . . .	222
27.34.3.2 GetOutputAsBitmap . . . . .	222
27.34.3.3 SetInput . . . . .	222
27.34.4 Member Data Documentation . . . . .	222
27.34.4.1 Input . . . . .	222
27.34.4.2 Output . . . . .	222
27.35gdcm::BoxRegion Class Reference . . . . .	222
27.35.1 Detailed Description . . . . .	224

27.35.2 Constructor & Destructor Documentation	224
27.35.2.1 BoxRegion	224
27.35.2.2 ~BoxRegion	224
27.35.2.3 BoxRegion	224
27.35.3 Member Function Documentation	224
27.35.3.1 Area	224
27.35.3.2 BoundingBox	224
27.35.3.3 Clone	224
27.35.3.4 ComputeBoundingBox	224
27.35.3.5 Empty	224
27.35.3.6 GetXMax	224
27.35.3.7 GetXMin	224
27.35.3.8 GetYMax	225
27.35.3.9 GetYMin	225
27.35.3.10 GetZMax	225
27.35.3.11 GetZMin	225
27.35.3.12 IsValid	225
27.35.3.13 operator=	225
27.35.3.14 Print	225
27.35.3.15 SetDomain	225
27.36 gdcmm::ByteBuffer Class Reference	225
27.36.1 Detailed Description	225
27.36.2 Constructor & Destructor Documentation	226
27.36.2.1 ByteBuffer	226
27.36.3 Member Function Documentation	226
27.36.3.1 Get	226
27.36.3.2 GetStart	226
27.36.3.3 ShiftEnd	226
27.36.3.4 UpdatePosition	226
27.37 gdcmm::ByteSwap< T > Class Template Reference	226
27.37.1 Detailed Description	226
27.37.2 Member Function Documentation	227
27.37.2.1 Swap	227
27.37.2.2 SwapFromSwapCodeIntoSystem	227
27.37.2.3 SwapRange	227
27.37.2.4 SwapRangeFromSwapCodeIntoSystem	227
27.37.2.5 SystemIsBigEndian	227



27.37.2.6 SystemIsLittleEndian . . . . .	227
27.38gdcmm::ByteSwapFilter Class Reference . . . . .	227
27.38.1 Detailed Description . . . . .	227
27.38.2 Constructor & Destructor Documentation . . . . .	228
27.38.2.1 ByteSwapFilter . . . . .	228
27.38.2.2 ~ByteSwapFilter . . . . .	228
27.38.3 Member Function Documentation . . . . .	228
27.38.3.1 ByteSwap . . . . .	228
27.38.3.2 SetByteSwapTag . . . . .	228
27.39gdcmm::ByteValue Class Reference . . . . .	228
27.39.1 Detailed Description . . . . .	230
27.39.2 Constructor & Destructor Documentation . . . . .	230
27.39.2.1 ByteValue . . . . .	230
27.39.2.2 ByteValue . . . . .	230
27.39.2.3 ~ByteValue . . . . .	230
27.39.3 Member Function Documentation . . . . .	230
27.39.3.1 Clear . . . . .	230
27.39.3.2 ComputeLength . . . . .	230
27.39.3.3 Fill . . . . .	231
27.39.3.4 GetBuffer . . . . .	231
27.39.3.5 GetLength . . . . .	231
27.39.3.6 GetPointer . . . . .	231
27.39.3.7 IsEmpty . . . . .	231
27.39.3.8 IsPrintable . . . . .	231
27.39.3.9 operator const std::vector< char > & . . . . .	232
27.39.3.10operator= . . . . .	232
27.39.3.11operator== . . . . .	232
27.39.3.12operator== . . . . .	232
27.39.3.13Print . . . . .	232
27.39.3.14PrintASCII . . . . .	232
27.39.3.15PrintASCIIXML . . . . .	232
27.39.3.16PrintGroupLength . . . . .	232
27.39.3.17PrintHex . . . . .	232
27.39.3.18PrintHexXML . . . . .	232
27.39.3.19PrintPNXML . . . . .	232
27.39.3.20Read . . . . .	232
27.39.3.21Read . . . . .	232

27.39.3.22SetLength . . . . .	232
27.39.3.23SetLengthOnly . . . . .	232
27.39.3.24Write . . . . .	232
27.39.3.25Write . . . . .	233
27.39.3.26WriteBuffer . . . . .	233
27.40gdcmm::CAPICryptoFactory Class Reference . . . . .	233
27.40.1 Constructor & Destructor Documentation . . . . .	234
27.40.1.1 CAPICryptoFactory . . . . .	234
27.40.2 Member Function Documentation . . . . .	234
27.40.2.1 CreateCMSProvider . . . . .	234
27.41gdcmm::CAPICryptographicMessageSyntax Class Reference . . . . .	234
27.41.1 Constructor & Destructor Documentation . . . . .	235
27.41.1.1 CAPICryptographicMessageSyntax . . . . .	235
27.41.1.2 ~CAPICryptographicMessageSyntax . . . . .	235
27.41.2 Member Function Documentation . . . . .	235
27.41.2.1 Decrypt . . . . .	235
27.41.2.2 Encrypt . . . . .	236
27.41.2.3 GetCipherType . . . . .	236
27.41.2.4 GetInitialized . . . . .	236
27.41.2.5 ParseCertificateFile . . . . .	236
27.41.2.6 ParseKeyFile . . . . .	236
27.41.2.7 SetCipherType . . . . .	236
27.41.2.8 SetPassword . . . . .	236
27.42gdcmm::network::CEchoRQ Class Reference . . . . .	236
27.42.1 Detailed Description . . . . .	237
27.42.2 Member Function Documentation . . . . .	238
27.42.2.1 ConstructPDV . . . . .	238
27.42.3 Member Data Documentation . . . . .	238
27.42.3.1 AffectedSOPClassUID . . . . .	238
27.42.3.2 MessageID . . . . .	238
27.43gdcmm::network::CEchoRSP Class Reference . . . . .	238
27.43.1 Detailed Description . . . . .	239
27.43.2 Member Function Documentation . . . . .	239
27.43.2.1 ConstructPDVByDataSet . . . . .	239
27.44gdcmm::network::CFind Class Reference . . . . .	239
27.44.1 Detailed Description . . . . .	239
27.45gdcmm::network::CFindCancelIRQ Class Reference . . . . .	240

27.45.1 Detailed Description	240
27.45.2 Member Function Documentation	241
27.45.2.1 ConstructPDVByDataSet	241
27.46gdcm::network::CFindRQ Class Reference	241
27.46.1 Detailed Description	242
27.46.2 Member Function Documentation	242
27.46.2.1 ConstructPDV	242
27.47gdcm::network::CFindRSP Class Reference	242
27.47.1 Detailed Description	243
27.47.2 Member Function Documentation	243
27.47.2.1 ConstructPDVByDataSet	243
27.48gdcm::network::CMoveCancelRq Class Reference	243
27.48.1 Member Function Documentation	244
27.48.1.1 ConstructPDVByDataSet	244
27.49gdcm::network::CMoveRQ Class Reference	245
27.49.1 Detailed Description	245
27.49.2 Member Function Documentation	246
27.49.2.1 ConstructPDV	246
27.50gdcm::network::CMoveRSP Class Reference	246
27.50.1 Detailed Description	247
27.50.2 Member Function Documentation	247
27.50.2.1 ConstructPDVByDataSet	247
27.51gdcm::Codec Class Reference	247
27.51.1 Detailed Description	248
27.52gdcm::Coder Class Reference	248
27.52.1 Detailed Description	249
27.52.2 Constructor & Destructor Documentation	249
27.52.2.1 ~Coder	249
27.52.3 Member Function Documentation	249
27.52.3.1 CanCode	249
27.52.3.2 Code	250
27.52.3.3 InternalCode	250
27.53gdcm::CodeString Class Reference	250
27.53.1 Detailed Description	251
27.53.2 Member Typedef Documentation	251
27.53.2.1 const_iterator	251
27.53.2.2 const_reference	251

27.53.2.3 <code>const_reverse_iterator</code> . . . . .	251
27.53.2.4 <code>difference_type</code> . . . . .	251
27.53.2.5 <code>iterator</code> . . . . .	251
27.53.2.6 <code>pointer</code> . . . . .	251
27.53.2.7 <code>reference</code> . . . . .	251
27.53.2.8 <code>reverse_iterator</code> . . . . .	251
27.53.2.9 <code>size_type</code> . . . . .	252
27.53.2.10 <code>value_type</code> . . . . .	252
27.53.3 Constructor & Destructor Documentation . . . . .	252
27.53.3.1 <code>CodeString</code> . . . . .	252
27.53.3.2 <code>CodeString</code> . . . . .	252
27.53.3.3 <code>CodeString</code> . . . . .	252
27.53.3.4 <code>CodeString</code> . . . . .	252
27.53.4 Member Function Documentation . . . . .	252
27.53.4.1 <code>GetAsString</code> . . . . .	252
27.53.4.2 <code>IsValid</code> . . . . .	252
27.53.4.3 <code>Size</code> . . . . .	252
27.53.4.4 <code>TrimInternal</code> . . . . .	252
27.53.5 Friends And Related Function Documentation . . . . .	252
27.53.5.1 <code>operator!=</code> . . . . .	252
27.53.5.2 <code>operator&lt;&lt;</code> . . . . .	252
27.53.5.3 <code>operator==</code> . . . . .	252
27.54 <code>gdcmm::Command</code> Class Reference . . . . .	252
27.54.1 Detailed Description . . . . .	254
27.54.2 Constructor & Destructor Documentation . . . . .	254
27.54.2.1 <code>Command</code> . . . . .	254
27.54.2.2 <code>~Command</code> . . . . .	254
27.54.3 Member Function Documentation . . . . .	254
27.54.3.1 <code>Execute</code> . . . . .	254
27.54.3.2 <code>Execute</code> . . . . .	254
27.55 <code>gdcmm::CommandDataSet</code> Class Reference . . . . .	254
27.55.1 Detailed Description . . . . .	256
27.55.2 Constructor & Destructor Documentation . . . . .	256
27.55.2.1 <code>CommandDataSet</code> . . . . .	256
27.55.2.2 <code>~CommandDataSet</code> . . . . .	256
27.55.3 Member Function Documentation . . . . .	256
27.55.3.1 <code>Insert</code> . . . . .	256

27.55.3.2 Read . . . . .	256
27.55.3.3 Replace . . . . .	256
27.55.3.4 Write . . . . .	256
27.55.4 Friends And Related Function Documentation . . . . .	256
27.55.4.1 operator<< . . . . .	256
27.56gdcmm::network::CompositeMessageFactory Class Reference . . . . .	256
27.56.1 Detailed Description . . . . .	257
27.56.2 Member Function Documentation . . . . .	257
27.56.2.1 ConstructCEchoRQ . . . . .	257
27.56.2.2 ConstructCFindRQ . . . . .	257
27.56.2.3 ConstructCMoveRQ . . . . .	257
27.56.2.4 ConstructCStoreRQ . . . . .	257
27.56.2.5 ConstructCStoreRSP . . . . .	257
27.57gdcmm::CompositeNetworkFunctions Class Reference . . . . .	257
27.57.1 Detailed Description . . . . .	258
27.57.2 Member Typedef Documentation . . . . .	258
27.57.2.1 KeyValuePairArrayType . . . . .	258
27.57.2.2 KeyValuePairType . . . . .	259
27.57.3 Member Function Documentation . . . . .	259
27.57.3.1 CEcho . . . . .	259
27.57.3.2 CFind . . . . .	259
27.57.3.3 CMove . . . . .	259
27.57.3.4 ConstructQuery . . . . .	260
27.57.3.5 ConstructQuery . . . . .	260
27.57.3.6 CStore . . . . .	260
27.58gdcmm::ConstCharWrapper Class Reference . . . . .	261
27.58.1 Detailed Description . . . . .	261
27.58.2 Constructor & Destructor Documentation . . . . .	261
27.58.2.1 ConstCharWrapper . . . . .	261
27.58.3 Member Function Documentation . . . . .	261
27.58.3.1 operator const char * . . . . .	261
27.59gdcmm::CP246ExplicitDataElement Class Reference . . . . .	261
27.59.1 Detailed Description . . . . .	262
27.59.2 Member Function Documentation . . . . .	262
27.59.2.1 GetLength . . . . .	262
27.59.2.2 Read . . . . .	263
27.59.2.3 ReadPreValue . . . . .	263

27.59.2.4 ReadValue . . . . .	263
27.59.2.5 ReadWithLength . . . . .	263
27.60gdcmm::CryptoFactory Class Reference . . . . .	263
27.60.1 Detailed Description . . . . .	264
27.60.2 Member Enumeration Documentation . . . . .	264
27.60.2.1 CryptoLib . . . . .	264
27.60.3 Constructor & Destructor Documentation . . . . .	264
27.60.3.1 CryptoFactory . . . . .	264
27.60.3.2 CryptoFactory . . . . .	264
27.60.3.3 ~CryptoFactory . . . . .	264
27.60.4 Member Function Documentation . . . . .	264
27.60.4.1 CreateCMSProvider . . . . .	264
27.60.4.2 GetFactoryInstance . . . . .	264
27.61gdcmm::CryptographicMessageSyntax Class Reference . . . . .	265
27.61.1 Detailed Description . . . . .	265
27.61.2 Member Enumeration Documentation . . . . .	266
27.61.2.1 CipherTypes . . . . .	266
27.61.3 Constructor & Destructor Documentation . . . . .	266
27.61.3.1 CryptographicMessageSyntax . . . . .	266
27.61.3.2 ~CryptographicMessageSyntax . . . . .	266
27.61.4 Member Function Documentation . . . . .	266
27.61.4.1 Decrypt . . . . .	266
27.61.4.2 Encrypt . . . . .	266
27.61.4.3 GetCipherType . . . . .	266
27.61.4.4 ParseCertificateFile . . . . .	266
27.61.4.5 ParseKeyFile . . . . .	266
27.61.4.6 SetCipherType . . . . .	267
27.61.4.7 SetPassword . . . . .	267
27.62gdcmm::CSAElement Class Reference . . . . .	267
27.62.1 Detailed Description . . . . .	269
27.62.2 Member Typedef Documentation . . . . .	269
27.62.2.1 DataPtr . . . . .	269
27.62.3 Constructor & Destructor Documentation . . . . .	269
27.62.3.1 CSAElement . . . . .	269
27.62.3.2 CSAElement . . . . .	269
27.62.4 Member Function Documentation . . . . .	269
27.62.4.1 GetByteValue . . . . .	269

27.62.4.2 GetKey . . . . .	269
27.62.4.3 GetName . . . . .	269
27.62.4.4 GetNoOfItems . . . . .	269
27.62.4.5 GetSyngoDT . . . . .	270
27.62.4.6 GetValue . . . . .	270
27.62.4.7 GetValue . . . . .	270
27.62.4.8 GetVM . . . . .	270
27.62.4.9 GetVR . . . . .	270
27.62.4.10 IsEmpty . . . . .	270
27.62.4.11 operator< . . . . .	270
27.62.4.12 operator= . . . . .	270
27.62.4.13 operator== . . . . .	270
27.62.4.14 SetByteValue . . . . .	270
27.62.4.15 SetKey . . . . .	271
27.62.4.16 SetName . . . . .	271
27.62.4.17 SetNoOfItems . . . . .	271
27.62.4.18 SetSyngoDT . . . . .	271
27.62.4.19 SetValue . . . . .	271
27.62.4.20 SetVM . . . . .	271
27.62.4.21 SetVR . . . . .	271
27.62.5 Friends And Related Function Documentation . . . . .	271
27.62.5.1 operator<< . . . . .	271
27.62.6 Member Data Documentation . . . . .	271
27.62.6.1 DataField . . . . .	271
27.62.6.2 KeyField . . . . .	271
27.62.6.3 NameField . . . . .	271
27.62.6.4 NoOfItemsField . . . . .	271
27.62.6.5 SyngoDTField . . . . .	271
27.62.6.6 ValueMultiplicityField . . . . .	271
27.62.6.7 VRField . . . . .	271
27.63gdcm::CSAHeader Class Reference . . . . .	272
27.63.1 Detailed Description . . . . .	273
27.63.2 Member Enumeration Documentation . . . . .	273
27.63.2.1 CSAHeaderType . . . . .	273
27.63.3 Constructor & Destructor Documentation . . . . .	274
27.63.3.1 CSAHeader . . . . .	274
27.63.3.2 ~CSAHeader . . . . .	274

27.63.4 Member Function Documentation	274
27.63.4.1 FindCSAElementByName	274
27.63.4.2 GetCSADataInfo	274
27.63.4.3 GetCSAEnd	274
27.63.4.4 GetCSAElementByName	274
27.63.4.5 GetCSAImageHeaderInfoTag	274
27.63.4.6 GetCSASeriesHeaderInfoTag	275
27.63.4.7 GetDataSet	275
27.63.4.8 GetFormat	275
27.63.4.9 GetInterfile	275
27.63.4.10 LoadFromDataElement	275
27.63.4.11 Print	275
27.63.4.12 Read	275
27.63.4.13 Write	275
27.63.5 Friends And Related Function Documentation	275
27.63.5.1 operator<<	275
27.64 gdcmm::CSAHeaderDict Class Reference	276
27.64.1 Detailed Description	276
27.64.2 Member Typedef Documentation	276
27.64.2.1 ConstIterator	276
27.64.2.2 Iterator	276
27.64.2.3 MapCSAHeaderDictEntry	276
27.64.3 Constructor & Destructor Documentation	277
27.64.3.1 CSAHeaderDict	277
27.64.4 Member Function Documentation	277
27.64.4.1 AddCSAHeaderDictEntry	277
27.64.4.2 Begin	277
27.64.4.3 End	277
27.64.4.4 GetCSAHeaderDictEntry	277
27.64.4.5 IsEmpty	277
27.64.4.6 LoadDefault	277
27.64.5 Friends And Related Function Documentation	277
27.64.5.1 Dicts	277
27.64.5.2 operator<<	277
27.65 gdcmm::CSAHeaderDictEntry Class Reference	277
27.65.1 Detailed Description	278
27.65.2 Constructor & Destructor Documentation	278



27.65.2.1 CSAHeaderDictEntry . . . . .	278
27.65.3 Member Function Documentation . . . . .	278
27.65.3.1 GetDescription . . . . .	278
27.65.3.2 GetName . . . . .	278
27.65.3.3 GetVM . . . . .	278
27.65.3.4 GetVR . . . . .	279
27.65.3.5 operator< . . . . .	279
27.65.3.6 SetDescription . . . . .	279
27.65.3.7 SetName . . . . .	279
27.65.3.8 SetVM . . . . .	279
27.65.3.9 SetVR . . . . .	279
27.65.4 Friends And Related Function Documentation . . . . .	279
27.65.4.1 operator<< . . . . .	279
27.66gdcm::CSAHeaderDictException Class Reference . . . . .	279
27.67gdcm::network::CStoreRQ Class Reference . . . . .	280
27.67.1 Detailed Description . . . . .	281
27.67.2 Member Function Documentation . . . . .	281
27.67.2.1 ConstructPDV . . . . .	281
27.68gdcm::network::CStoreRSP Class Reference . . . . .	281
27.68.1 Detailed Description . . . . .	282
27.68.2 Member Function Documentation . . . . .	282
27.68.2.1 ConstructPDV . . . . .	282
27.69gdcm::Curve Class Reference . . . . .	283
27.69.1 Detailed Description . . . . .	284
27.69.2 Constructor & Destructor Documentation . . . . .	284
27.69.2.1 Curve . . . . .	284
27.69.2.2 ~Curve . . . . .	284
27.69.2.3 Curve . . . . .	284
27.69.3 Member Function Documentation . . . . .	284
27.69.3.1 Decode . . . . .	285
27.69.3.2 GetAsPoints . . . . .	285
27.69.3.3 GetCurveDataDescriptor . . . . .	285
27.69.3.4 GetDataValueRepresentation . . . . .	285
27.69.3.5 GetDimensions . . . . .	285
27.69.3.6 GetGroup . . . . .	285
27.69.3.7 GetNumberOfCurves . . . . .	285
27.69.3.8 GetNumberOfPoints . . . . .	285

27.69.3.9 GetTypeOfData . . . . .	285
27.69.3.10GetTypeOfDataDescription . . . . .	285
27.69.3.11IsEmpty . . . . .	285
27.69.3.12Print . . . . .	285
27.69.3.13SetCoordinateStartValue . . . . .	285
27.69.3.14SetCoordinateStepValue . . . . .	285
27.69.3.15SetCurve . . . . .	285
27.69.3.16SetCurveDataDescriptor . . . . .	285
27.69.3.17SetCurveDescription . . . . .	285
27.69.3.18SetDataValueRepresentation . . . . .	285
27.69.3.19SetDimensions . . . . .	285
27.69.3.20SetGroup . . . . .	285
27.69.3.21SetNumberOfPoints . . . . .	285
27.69.3.22SetTypeOfData . . . . .	285
27.69.3.23Update . . . . .	285
27.70gdcmm::DataElement Class Reference . . . . .	286
27.70.1 Detailed Description . . . . .	288
27.70.2 Member Typedef Documentation . . . . .	289
27.70.2.1 ValuePtr . . . . .	289
27.70.3 Constructor & Destructor Documentation . . . . .	289
27.70.3.1 DataElement . . . . .	289
27.70.3.2 DataElement . . . . .	289
27.70.4 Member Function Documentation . . . . .	289
27.70.4.1 Clear . . . . .	289
27.70.4.2 Empty . . . . .	289
27.70.4.3 GetByteValue . . . . .	289
27.70.4.4 GetLength . . . . .	289
27.70.4.5 GetSequenceOfFragments . . . . .	289
27.70.4.6 GetSequenceOfFragments . . . . .	290
27.70.4.7 GetTag . . . . .	290
27.70.4.8 GetTag . . . . .	290
27.70.4.9 GetValue . . . . .	290
27.70.4.10GetValue . . . . .	290
27.70.4.11GetValueAsSQ . . . . .	290
27.70.4.12GetVL . . . . .	291
27.70.4.13GetVL . . . . .	291
27.70.4.14GetVR . . . . .	291

27.70.4.15	IsEmpty	291
27.70.4.16	IsUndefinedLength	291
27.70.4.17	operator<	291
27.70.4.18	operator=	291
27.70.4.19	operator==	292
27.70.4.20	Read	292
27.70.4.21	ReadOrSkip	292
27.70.4.22	ReadPreValue	292
27.70.4.23	ReadValue	292
27.70.4.24	ReadValueWithLength	292
27.70.4.25	ReadWithLength	292
27.70.4.26	SetByteValue	292
27.70.4.27	SetTag	292
27.70.4.28	SetValue	293
27.70.4.29	SetValueFieldLength	293
27.70.4.30	SetVL	293
27.70.4.31	SetVLToUndefined	293
27.70.4.32	SetVR	293
27.70.4.33	Write	294
27.70.5	Friends And Related Function Documentation	294
27.70.5.1	operator<<	294
27.70.6	Member Data Documentation	294
27.70.6.1	TagField	294
27.70.6.2	ValueField	294
27.70.6.3	ValueLengthField	294
27.70.6.4	VRField	294
27.71	gdcm::DataElementException Class Reference	294
27.72	gdcm::DataEvent Class Reference	295
27.72.1	Detailed Description	296
27.72.2	Member Typedef Documentation	296
27.72.2.1	Self	296
27.72.2.2	Superclass	296
27.72.3	Constructor & Destructor Documentation	297
27.72.3.1	DataEvent	297
27.72.3.2	~DataEvent	297
27.72.3.3	DataEvent	297
27.72.4	Member Function Documentation	297

27.72.4.1 CheckEvent . . . . .	297
27.72.4.2 GetData . . . . .	297
27.72.4.3 GetDataLength . . . . .	297
27.72.4.4 GetEventName . . . . .	297
27.72.4.5 MakeObject . . . . .	297
27.72.4.6 SetData . . . . .	297
27.73gdcm::DataSet Class Reference . . . . .	297
27.73.1 Detailed Description . . . . .	300
27.73.2 Member Typedef Documentation . . . . .	300
27.73.2.1 ConstIterator . . . . .	300
27.73.2.2 DataElementSet . . . . .	300
27.73.2.3 Iterator . . . . .	300
27.73.2.4 SizeType . . . . .	300
27.73.3 Member Function Documentation . . . . .	300
27.73.3.1 Begin . . . . .	300
27.73.3.2 Begin . . . . .	300
27.73.3.3 Clear . . . . .	300
27.73.3.4 ComputeDataElement . . . . .	301
27.73.3.5 ComputeGroupLength . . . . .	301
27.73.3.6 End . . . . .	301
27.73.3.7 End . . . . .	301
27.73.3.8 FindDataElement . . . . .	301
27.73.3.9 FindDataElement . . . . .	301
27.73.3.10FindNextDataElement . . . . .	301
27.73.3.11GetDataElement . . . . .	301
27.73.3.12GetDataElement . . . . .	302
27.73.3.13GetDEEnd . . . . .	302
27.73.3.14GetDES . . . . .	302
27.73.3.15GetDES . . . . .	302
27.73.3.16GetLength . . . . .	302
27.73.3.17GetMediaStorage . . . . .	302
27.73.3.18GetPrivateCreator . . . . .	302
27.73.3.19Insert . . . . .	302
27.73.3.20InsertDataElement . . . . .	302
27.73.3.21IsEmpty . . . . .	303
27.73.3.22operator() . . . . .	303
27.73.3.23operator= . . . . .	303

27.73.3.24operator[]	303
27.73.3.25Print	303
27.73.3.26Read	303
27.73.3.27ReadNested	303
27.73.3.28ReadSelectedPrivateTags	303
27.73.3.29ReadSelectedPrivateTagsWithLength	303
27.73.3.30ReadSelectedTags	303
27.73.3.31ReadSelectedTagsWithLength	303
27.73.3.32ReadUpToTag	303
27.73.3.33ReadUpToTagWithLength	303
27.73.3.34ReadWithLength	303
27.73.3.35Remove	303
27.73.3.36Replace	303
27.73.3.37ReplaceEmpty	304
27.73.3.38Size	304
27.73.3.39Write	304
27.73.4 Friends And Related Function Documentation	304
27.73.4.1 CSAHeader	304
27.73.4.2 operator<<	304
27.74gdcm::DataSetEvent Class Reference	304
27.74.1 Detailed Description	306
27.74.2 Member Typedef Documentation	306
27.74.2.1 Self	306
27.74.2.2 Superclass	306
27.74.3 Constructor & Destructor Documentation	306
27.74.3.1 DataSetEvent	306
27.74.3.2 ~DataSetEvent	306
27.74.3.3 DataSetEvent	306
27.74.4 Member Function Documentation	306
27.74.4.1 CheckEvent	306
27.74.4.2 GetDataSet	306
27.74.4.3 GetEventName	306
27.74.4.4 MakeObject	306
27.75gdcm::DataSetHelper Class Reference	307
27.75.1 Detailed Description	307
27.75.2 Member Function Documentation	307
27.75.2.1 ComputeVR	307

27.76gdcmm::Decoder Class Reference . . . . .	307
27.76.1 Detailed Description . . . . .	308
27.76.2 Constructor & Destructor Documentation . . . . .	308
27.76.2.1 ~Decoder . . . . .	308
27.76.3 Member Function Documentation . . . . .	308
27.76.3.1 CanDecode . . . . .	308
27.76.3.2 Decode . . . . .	308
27.76.3.3 DecodeByStreams . . . . .	308
27.77gdcmm::DefinedTerms Class Reference . . . . .	309
27.77.1 Detailed Description . . . . .	309
27.77.2 Constructor & Destructor Documentation . . . . .	309
27.77.2.1 DefinedTerms . . . . .	309
27.78gdcmm::Defs Class Reference . . . . .	309
27.78.1 Detailed Description . . . . .	310
27.78.2 Constructor & Destructor Documentation . . . . .	310
27.78.2.1 Defs . . . . .	310
27.78.2.2 ~Defs . . . . .	310
27.78.3 Member Function Documentation . . . . .	310
27.78.3.1 GetIODFromFile . . . . .	310
27.78.3.2 GetIODNameFromMediaStorage . . . . .	310
27.78.3.3 GetIODs . . . . .	311
27.78.3.4 GetIODs . . . . .	311
27.78.3.5 GetMacros . . . . .	311
27.78.3.6 GetMacros . . . . .	311
27.78.3.7 GetModules . . . . .	311
27.78.3.8 GetModules . . . . .	311
27.78.3.9 GetTypeFromTag . . . . .	311
27.78.3.10IsEmpty . . . . .	311
27.78.3.11LoadDefaults . . . . .	311
27.78.3.12LoadFromFile . . . . .	311
27.78.3.13Verify . . . . .	311
27.78.3.14Verify . . . . .	311
27.78.4 Friends And Related Function Documentation . . . . .	311
27.78.4.1 Global . . . . .	311
27.79gdcmm::DeltaEncodingCodec Class Reference . . . . .	311
27.79.1 Detailed Description . . . . .	313
27.79.2 Constructor & Destructor Documentation . . . . .	313

27.79.2.1 DeltaEncodingCodec . . . . .	313
27.79.2.2 ~DeltaEncodingCodec . . . . .	313
27.79.3 Member Function Documentation . . . . .	313
27.79.3.1 CanDecode . . . . .	313
27.79.3.2 Decode . . . . .	313
27.79.3.3 Decode . . . . .	313
27.80gdcm::DICOMDIR Class Reference . . . . .	313
27.80.1 Detailed Description . . . . .	313
27.80.2 Constructor & Destructor Documentation . . . . .	313
27.80.2.1 DICOMDIR . . . . .	313
27.80.2.2 DICOMDIR . . . . .	314
27.81gdcm::DICOMDIRGenerator Class Reference . . . . .	314
27.81.1 Detailed Description . . . . .	314
27.81.2 Member Typedef Documentation . . . . .	315
27.81.2.1 FilenamesType . . . . .	315
27.81.2.2 FilenameType . . . . .	315
27.81.3 Constructor & Destructor Documentation . . . . .	315
27.81.3.1 DICOMDIRGenerator . . . . .	315
27.81.3.2 ~DICOMDIRGenerator . . . . .	315
27.81.4 Member Function Documentation . . . . .	315
27.81.4.1 AddImageDirectoryRecord . . . . .	315
27.81.4.2 AddPatientDirectoryRecord . . . . .	315
27.81.4.3 AddSeriesDirectoryRecord . . . . .	315
27.81.4.4 AddStudyDirectoryRecord . . . . .	315
27.81.4.5 Generate . . . . .	315
27.81.4.6 GetFile . . . . .	315
27.81.4.7 GetScanner . . . . .	315
27.81.4.8 SetDescriptor . . . . .	316
27.81.4.9 SetFile . . . . .	316
27.81.4.10SetFilenames . . . . .	316
27.81.4.11SetRootDirectory . . . . .	316
27.82gdcm::Dict Class Reference . . . . .	316
27.82.1 Detailed Description . . . . .	317
27.82.2 Member Typedef Documentation . . . . .	317
27.82.2.1 ConstIterator . . . . .	317
27.82.2.2 Iterator . . . . .	317
27.82.2.3 MapDictEntry . . . . .	317

27.82.3 Constructor & Destructor Documentation . . . . .	317
27.82.3.1 Dict . . . . .	317
27.82.4 Member Function Documentation . . . . .	317
27.82.4.1 AddDictEntry . . . . .	317
27.82.4.2 Begin . . . . .	317
27.82.4.3 End . . . . .	317
27.82.4.4 GetDictEntry . . . . .	318
27.82.4.5 GetDictEntryByKeyword . . . . .	318
27.82.4.6 GetDictEntryByName . . . . .	318
27.82.4.7 GetKeywordFromTag . . . . .	318
27.82.4.8 IsEmpty . . . . .	318
27.82.4.9 LoadDefault . . . . .	318
27.82.5 Friends And Related Function Documentation . . . . .	318
27.82.5.1 Dicts . . . . .	318
27.82.5.2 operator<< . . . . .	318
27.83gdcmm::DictConverter Class Reference . . . . .	318
27.83.1 Detailed Description . . . . .	319
27.83.2 Member Enumeration Documentation . . . . .	320
27.83.2.1 OutputTypes . . . . .	320
27.83.3 Constructor & Destructor Documentation . . . . .	320
27.83.3.1 DictConverter . . . . .	320
27.83.3.2 ~DictConverter . . . . .	320
27.83.4 Member Function Documentation . . . . .	320
27.83.4.1 AddGroupLength . . . . .	320
27.83.4.2 Convert . . . . .	320
27.83.4.3 ConvertToCXX . . . . .	320
27.83.4.4 ConvertToXML . . . . .	320
27.83.4.5 GetDictName . . . . .	320
27.83.4.6 GetInputFilename . . . . .	320
27.83.4.7 GetOutputFilename . . . . .	320
27.83.4.8 GetOutputType . . . . .	320
27.83.4.9 Readuint16 . . . . .	320
27.83.4.10ReadVM . . . . .	320
27.83.4.11ReadVR . . . . .	320
27.83.4.12SetDictName . . . . .	320
27.83.4.13SetInputFileName . . . . .	320
27.83.4.14SetOutputFileName . . . . .	320



27.83.4.15SetOutputType . . . . .	320
27.83.4.16WriteFooter . . . . .	320
27.83.4.17WriteHeader . . . . .	321
27.84gdcmm::DictEntry Class Reference . . . . .	321
27.84.1 Detailed Description . . . . .	322
27.84.2 Constructor & Destructor Documentation . . . . .	322
27.84.2.1 DictEntry . . . . .	322
27.84.3 Member Function Documentation . . . . .	322
27.84.3.1 GetKeyword . . . . .	322
27.84.3.2 GetName . . . . .	322
27.84.3.3 GetRetired . . . . .	322
27.84.3.4 GetVM . . . . .	322
27.84.3.5 GetVR . . . . .	323
27.84.3.6 IsUnique . . . . .	323
27.84.3.7 SetElementXX . . . . .	323
27.84.3.8 SetGroupXX . . . . .	323
27.84.3.9 SetKeyword . . . . .	323
27.84.3.10SetName . . . . .	323
27.84.3.11SetRetired . . . . .	323
27.84.3.12SetVM . . . . .	323
27.84.3.13SetVR . . . . .	323
27.84.4 Friends And Related Function Documentation . . . . .	323
27.84.4.1 Dict . . . . .	323
27.84.4.2 operator<< . . . . .	323
27.85gdcmm::DictPrinter Class Reference . . . . .	323
27.85.1 Detailed Description . . . . .	325
27.85.2 Constructor & Destructor Documentation . . . . .	325
27.85.2.1 DictPrinter . . . . .	325
27.85.2.2 ~DictPrinter . . . . .	325
27.85.3 Member Function Documentation . . . . .	325
27.85.3.1 Print . . . . .	325
27.85.3.2 PrintDataElement2 . . . . .	325
27.85.3.3 PrintDataSet2 . . . . .	325
27.86gdcmm::Dicts Class Reference . . . . .	325
27.86.1 Detailed Description . . . . .	326
27.86.2 Member Enumeration Documentation . . . . .	326
27.86.2.1 ConstructorType . . . . .	326

27.86.3 Constructor & Destructor Documentation	326
27.86.3.1 Dicts	326
27.86.3.2 ~Dicts	326
27.86.4 Member Function Documentation	326
27.86.4.1 GetConstructorString	327
27.86.4.2 GetCSAHeaderDict	327
27.86.4.3 GetDictEntry	327
27.86.4.4 GetDictEntry	327
27.86.4.5 GetPrivateDict	327
27.86.4.6 GetPrivateDict	327
27.86.4.7 GetPublicDict	327
27.86.4.8 IsEmpty	327
27.86.4.9 LoadDefaults	327
27.86.5 Friends And Related Function Documentation	327
27.86.5.1 Global	327
27.86.5.2 operator<<	327
27.87gdcmm::network::DIMSE Class Reference	328
27.87.1 Detailed Description	328
27.87.2 Member Enumeration Documentation	328
27.87.2.1 CommandTypes	328
27.88gdcmm::DirectionCosines Class Reference	329
27.88.1 Detailed Description	330
27.88.2 Constructor & Destructor Documentation	330
27.88.2.1 DirectionCosines	330
27.88.2.2 DirectionCosines	330
27.88.2.3 ~DirectionCosines	330
27.88.3 Member Function Documentation	330
27.88.3.1 ComputeDistAlongNormal	330
27.88.3.2 Cross	330
27.88.3.3 CrossDot	330
27.88.3.4 Dot	330
27.88.3.5 IsValid	330
27.88.3.6 Normalize	331
27.88.3.7 operator const double *	331
27.88.3.8 Print	331
27.88.3.9 SetFromString	331
27.89gdcmm::Directory Class Reference	331

27.89.1 Detailed Description . . . . .	332
27.89.2 Member Typedef Documentation . . . . .	332
27.89.2.1 FilenamesType . . . . .	332
27.89.2.2 FilenameType . . . . .	332
27.89.3 Constructor & Destructor Documentation . . . . .	332
27.89.3.1 Directory . . . . .	332
27.89.3.2 ~Directory . . . . .	332
27.89.4 Member Function Documentation . . . . .	332
27.89.4.1 Explore . . . . .	332
27.89.4.2 GetDirectories . . . . .	333
27.89.4.3 GetFilenames . . . . .	333
27.89.4.4 GetToplevel . . . . .	333
27.89.4.5 Load . . . . .	333
27.89.4.6 Print . . . . .	333
27.89.5 Friends And Related Function Documentation . . . . .	333
27.89.5.1 operator<< . . . . .	333
27.90gdcmm::DirectoryHelper Class Reference . . . . .	334
27.90.1 Detailed Description . . . . .	334
27.90.2 Member Function Documentation . . . . .	334
27.90.2.1 GetCTImageSeriesUIDs . . . . .	334
27.90.2.2 GetFilenamesFromSeriesUIDs . . . . .	334
27.90.2.3 GetFrameOfReference . . . . .	334
27.90.2.4 GetMRImageSeriesUIDs . . . . .	334
27.90.2.5 GetRTStructSeriesUIDs . . . . .	335
27.90.2.6 GetSeriesUIDsBySOPClassUID . . . . .	335
27.90.2.7 GetSOPClassUID . . . . .	335
27.90.2.8 GetStringValueFromTag . . . . .	335
27.90.2.9 LoadImageFromFiles . . . . .	335
27.90.2.10RetrieveSOPInstanceUIDFromIndex . . . . .	335
27.90.2.11RetrieveSOPInstanceUIDFromZPosition . . . . .	335
27.91gdcmm::DummyValueGenerator Class Reference . . . . .	335
27.91.1 Detailed Description . . . . .	335
27.91.2 Member Function Documentation . . . . .	335
27.91.2.1 Generate . . . . .	336
27.92gdcmm::Dumper Class Reference . . . . .	336
27.92.1 Detailed Description . . . . .	337
27.92.2 Constructor & Destructor Documentation . . . . .	337

27.92.2.1 Dumper	337
27.92.2.2 ~Dumper	337
27.93gdcmm::Element< TVR, TVM > Class Template Reference	338
27.93.1 Detailed Description	340
27.93.2 Member Typedef Documentation	340
27.93.2.1 Type	340
27.93.3 Member Function Documentation	340
27.93.3.1 GetAsDataElement	340
27.93.3.2 GetLength	340
27.93.3.3 GetValue	340
27.93.3.4 GetValue	340
27.93.3.5 GetValues	340
27.93.3.6 GetVM	340
27.93.3.7 GetVR	340
27.93.3.8 operator[]	340
27.93.3.9 Print	340
27.93.3.10Read	341
27.93.3.11Set	341
27.93.3.12SetFromDataElement	341
27.93.3.13SetNoSwap	341
27.93.3.14SetValue	341
27.93.3.15Write	341
27.93.4 Member Data Documentation	341
27.93.4.1 Internal	341
27.94gdcmm::Element< TVR, VM::VM1_2 > Class Template Reference	341
27.94.1 Member Typedef Documentation	342
27.94.1.1 Parent	342
27.94.2 Member Function Documentation	342
27.94.2.1 SetLength	342
27.95gdcmm::Element< TVR, VM::VM1_n > Class Template Reference	342
27.95.1 Member Typedef Documentation	344
27.95.1.1 Type	344
27.95.2 Constructor & Destructor Documentation	344
27.95.2.1 Element	344
27.95.2.2 ~Element	344
27.95.2.3 Element	344
27.95.3 Member Function Documentation	344

27.95.3.1 GetAsDataElement . . . . .	344
27.95.3.2 GetLength . . . . .	344
27.95.3.3 GetValue . . . . .	344
27.95.3.4 GetValue . . . . .	344
27.95.3.5 GetVM . . . . .	344
27.95.3.6 GetVR . . . . .	344
27.95.3.7 operator= . . . . .	344
27.95.3.8 operator[] . . . . .	344
27.95.3.9 Print . . . . .	344
27.95.3.10Read . . . . .	344
27.95.3.11Set . . . . .	344
27.95.3.12SetArray . . . . .	345
27.95.3.13SetFromDataElement . . . . .	345
27.95.3.14SetLength . . . . .	345
27.95.3.15SetNoSwap . . . . .	345
27.95.3.16SetValue . . . . .	345
27.95.3.17Write . . . . .	345
27.95.3.18WriteASCII . . . . .	345
27.96gdcm::Element< TVR, VM::VM2_2n > Class Template Reference . . . . .	345
27.96.1 Member Typedef Documentation . . . . .	347
27.96.1.1 Parent . . . . .	347
27.96.2 Member Function Documentation . . . . .	347
27.96.2.1 SetLength . . . . .	347
27.97gdcm::Element< TVR, VM::VM2_n > Class Template Reference . . . . .	347
27.97.1 Member Typedef Documentation . . . . .	348
27.97.1.1 Parent . . . . .	348
27.97.2 Member Function Documentation . . . . .	348
27.97.2.1 SetLength . . . . .	348
27.98gdcm::Element< TVR, VM::VM3_3n > Class Template Reference . . . . .	348
27.98.1 Member Typedef Documentation . . . . .	350
27.98.1.1 Parent . . . . .	350
27.98.2 Member Function Documentation . . . . .	350
27.98.2.1 SetLength . . . . .	350
27.99gdcm::Element< TVR, VM::VM3_n > Class Template Reference . . . . .	350
27.99.1 Member Typedef Documentation . . . . .	351
27.99.1.1 Parent . . . . .	351
27.99.2 Member Function Documentation . . . . .	351

27.99.2.1 SetLength . . . . .	351
27.100dcm::Element< VR::AS, VM::VM5 > Class Template Reference . . . . .	351
27.100.1Member Function Documentation . . . . .	352
27.100.1.1GetLength . . . . .	352
27.100.1.2Print . . . . .	352
27.100.2Member Data Documentation . . . . .	352
27.100.2.1Internal . . . . .	352
27.101dcm::Element< VR::OB, VM::VM1 > Class Template Reference . . . . .	352
27.102dcm::Element< VR::OW, VM::VM1 > Class Template Reference . . . . .	353
27.103dcm::ElementDisableCombinations< TVR, TVM > Class Template Reference . . . . .	355
27.103.1Detailed Description . . . . .	355
27.104dcm::ElementDisableCombinations< VR::OB, VM::VM1_n > Class Template Reference . . . . .	356
27.105dcm::ElementDisableCombinations< VR::OW, VM::VM1_n > Class Template Reference . . . . .	356
27.106dcm::EncapsulatedDocument Class Reference . . . . .	356
27.106.1Detailed Description . . . . .	356
27.106.2Constructor & Destructor Documentation . . . . .	356
27.106.2.1EncapsulatedDocument . . . . .	356
27.107dcm::EncodingImplementation< T > Class Template Reference . . . . .	357
27.107.1Detailed Description . . . . .	357
27.108dcm::EncodingImplementation< VR::VRASCII > Class Template Reference . . . . .	357
27.108.1Member Function Documentation . . . . .	357
27.108.1.1Read . . . . .	357
27.108.1.2ReadComputeLength . . . . .	358
27.108.1.3ReadNoSwap . . . . .	358
27.108.1.4Write . . . . .	358
27.108.1.5Write . . . . .	358
27.108.1.6Write . . . . .	358
27.109dcm::EncodingImplementation< VR::VRBINARY > Class Template Reference . . . . .	358
27.109.1Member Function Documentation . . . . .	358
27.109.1.1Read . . . . .	358
27.109.1.2ReadComputeLength . . . . .	359
27.109.1.3ReadNoSwap . . . . .	359
27.109.1.4Write . . . . .	359
27.110dcm::EndEvent Class Reference . . . . .	359
27.111dcm::EnumeratedValues Class Reference . . . . .	360
27.111.1Detailed Description . . . . .	360
27.111.2Constructor & Destructor Documentation . . . . .	361

27.111.2.1EnumeratedValues . . . . .	361
27.112dcm::Event Class Reference . . . . .	361
27.112.1Detailed Description . . . . .	363
27.112.2Constructor & Destructor Documentation . . . . .	363
27.112.2.1Event . . . . .	363
27.112.2.2Event . . . . .	363
27.112.2.3~Event . . . . .	363
27.112.3Member Function Documentation . . . . .	363
27.112.3.1CheckEvent . . . . .	363
27.112.3.2GetEventName . . . . .	363
27.112.3.3MakeObject . . . . .	363
27.112.3.4Print . . . . .	363
27.113dcm::Exception Class Reference . . . . .	364
27.113.1Detailed Description . . . . .	365
27.113.2Constructor & Destructor Documentation . . . . .	365
27.113.2.1Exception . . . . .	365
27.113.2.2~Exception . . . . .	365
27.113.3Member Function Documentation . . . . .	365
27.113.3.1GetDescription . . . . .	365
27.113.3.2what . . . . .	365
27.114dcm::ExitEvent Class Reference . . . . .	365
27.115dcm::ExplicitDataElement Class Reference . . . . .	367
27.115.1Detailed Description . . . . .	368
27.115.2Member Function Documentation . . . . .	368
27.115.2.1GetLength . . . . .	368
27.115.2.2Read . . . . .	368
27.115.2.3ReadPreValue . . . . .	368
27.115.2.4ReadValue . . . . .	368
27.115.2.5ReadWithLength . . . . .	368
27.115.2.6Write . . . . .	368
27.116dcm::ExplicitImplicitDataElement Class Reference . . . . .	368
27.116.1Detailed Description . . . . .	370
27.116.2Member Function Documentation . . . . .	370
27.116.2.1GetLength . . . . .	370
27.116.2.2Read . . . . .	370
27.116.2.3ReadPreValue . . . . .	370
27.116.2.4ReadValue . . . . .	370

27.116.2.5	ReadWithLength . . . . .	370
27.117	dcm::Fiducials Class Reference . . . . .	370
27.117.1	Detailed Description . . . . .	370
27.117.2	Constructor & Destructor Documentation . . . . .	371
27.117.2.1	Fiducials . . . . .	371
27.118	dcm::File Class Reference . . . . .	371
27.118.1	Detailed Description . . . . .	372
27.118.2	Constructor & Destructor Documentation . . . . .	373
27.118.2.1	File . . . . .	373
27.118.3	Member Function Documentation . . . . .	373
27.118.3.1	GetDataSet . . . . .	373
27.118.3.2	GetDataSet . . . . .	373
27.118.3.3	GetHeader . . . . .	373
27.118.3.4	GetHeader . . . . .	374
27.118.3.5	Read . . . . .	374
27.118.3.6	SetDataSet . . . . .	374
27.118.3.7	SetHeader . . . . .	374
27.118.3.8	Write . . . . .	374
27.118.4	Friends And Related Function Documentation . . . . .	374
27.118.4.1	operator<< . . . . .	374
27.119	dcm::FileAnonymizer Class Reference . . . . .	374
27.119.1	Detailed Description . . . . .	376
27.119.2	Constructor & Destructor Documentation . . . . .	376
27.119.2.1	FileAnonymizer . . . . .	376
27.119.2.2	~FileAnonymizer . . . . .	376
27.119.3	Member Function Documentation . . . . .	376
27.119.3.1	Empty . . . . .	376
27.119.3.2	Remove . . . . .	376
27.119.3.3	Replace . . . . .	377
27.119.3.4	Replace . . . . .	377
27.119.3.5	SetInputFileName . . . . .	377
27.119.3.6	SetOutputFileName . . . . .	377
27.119.3.7	Write . . . . .	377
27.120	dcm::FileChangeTransferSyntax Class Reference . . . . .	377
27.120.1	Detailed Description . . . . .	379
27.120.2	Constructor & Destructor Documentation . . . . .	379
27.120.2.1	FileChangeTransferSyntax . . . . .	379



27.120.2.2~FileChangeTransferSyntax . . . . .	379
27.120.3 Member Function Documentation . . . . .	379
27.120.3.1Change . . . . .	379
27.120.3.2GetCodec . . . . .	379
27.120.3.3New . . . . .	379
27.120.3.4SetInputFileName . . . . .	380
27.120.3.5SetOutputFileName . . . . .	380
27.120.3.6SetTransferSyntax . . . . .	380
27.121 dcm::FileDerivation Class Reference . . . . .	380
27.121.1 Detailed Description . . . . .	381
27.121.2 Constructor & Destructor Documentation . . . . .	381
27.121.2.1FileDerivation . . . . .	381
27.121.2.2~FileDerivation . . . . .	381
27.121.3 Member Function Documentation . . . . .	381
27.121.3.1AddDerivationDescription . . . . .	381
27.121.3.2AddPurposeOfReferenceCodeSequence . . . . .	381
27.121.3.3AddReference . . . . .	381
27.121.3.4AddSourceImageSequence . . . . .	381
27.121.3.5Derive . . . . .	381
27.121.3.6GetFile . . . . .	382
27.121.3.7GetFile . . . . .	382
27.121.3.8SetDerivationCodeSequenceCodeValue . . . . .	382
27.121.3.9SetDerivationDescription . . . . .	382
27.121.3.10SetFile . . . . .	382
27.121.3.11SetPurposeOfReferenceCodeSequenceCodeValue . . . . .	382
27.122 dcm::FileExplicitFilter Class Reference . . . . .	382
27.122.1 Detailed Description . . . . .	383
27.122.2 Constructor & Destructor Documentation . . . . .	383
27.122.2.1FileExplicitFilter . . . . .	383
27.122.2.2~FileExplicitFilter . . . . .	383
27.122.3 Member Function Documentation . . . . .	383
27.122.3.1Change . . . . .	384
27.122.3.2ChangeFMI . . . . .	384
27.122.3.3GetFile . . . . .	384
27.122.3.4ProcessDataSet . . . . .	384
27.122.3.5SetChangePrivateTags . . . . .	384
27.122.3.6SetFile . . . . .	384

27.122.3.7SetRecomputeItemLength . . . . .	384
27.122.3.8SetRecomputeSequenceLength . . . . .	384
27.122.3.9SetUseVRUN . . . . .	384
27.123.0dcm::FileMetaInformation Class Reference . . . . .	384
27.123.1Detailed Description . . . . .	387
27.123.2Constructor & Destructor Documentation . . . . .	387
27.123.2.1FileMetaInformation . . . . .	387
27.123.2.2FileMetaInformation . . . . .	387
27.123.3Member Function Documentation . . . . .	387
27.123.3.1AppendImplementationClassUID . . . . .	387
27.123.3.2ComputeDataSetMediaStorageSOPClass . . . . .	387
27.123.3.3ComputeDataSetTransferSyntax . . . . .	387
27.123.3.4Default . . . . .	387
27.123.3.5FillFromDataSet . . . . .	387
27.123.3.6GetDataSetTransferSyntax . . . . .	387
27.123.3.7GetFileMetaInformationVersion . . . . .	388
27.123.3.8GetFullLength . . . . .	388
27.123.3.9GetGDCMImplementationClassUID . . . . .	388
27.123.3.10GetGDCMImplementationVersionName . . . . .	388
27.123.3.11GetGDCMSourceApplicationEntityTitle . . . . .	388
27.123.3.12GetImplementationClassUID . . . . .	388
27.123.3.13GetImplementationVersionName . . . . .	388
27.123.3.14GetMediaStorage . . . . .	388
27.123.3.15GetMediaStorageAsString . . . . .	388
27.123.3.16GetMetaInformationTS . . . . .	388
27.123.3.17GetPreamble . . . . .	388
27.123.3.18GetPreamble . . . . .	388
27.123.3.19GetSourceApplicationEntityTitle . . . . .	388
27.123.3.20Insert . . . . .	388
27.123.3.21IsValid . . . . .	388
27.123.3.22Read . . . . .	388
27.123.3.23ReadCompat . . . . .	388
27.123.3.24ReadCompatInternal . . . . .	388
27.123.3.25Replace . . . . .	389
27.123.3.26SetDataSetTransferSyntax . . . . .	389
27.123.3.27SetImplementationClassUID . . . . .	389
27.123.3.28SetImplementationVersionName . . . . .	389

27.123.3.2	SetPreamble	389
27.123.3.3	SetSourceApplicationEntityTitle	389
27.123.3.3	Write	389
27.123.4	Friends And Related Function Documentation	389
27.123.4.1	operator<<	389
27.123.5	Member Data Documentation	389
27.123.5.1	DataSetMS	389
27.123.5.2	DataSetTS	389
27.123.5.3	MetaInformationTS	390
27.124	dcm::Filename Class Reference	390
27.124.1	Detailed Description	390
27.124.2	Constructor & Destructor Documentation	391
27.124.2.1	Filename	391
27.124.3	Member Function Documentation	391
27.124.3.1	EndWith	391
27.124.3.2	GetExtension	391
27.124.3.3	GetFileName	391
27.124.3.4	GetName	391
27.124.3.5	GetPath	391
27.124.3.6	IsEmpty	391
27.124.3.7	IsIdentical	391
27.124.3.8	Join	391
27.124.3.9	operator const char *	391
27.124.3.10	ToUnixSlashes	391
27.124.3.11	ToWindowsSlashes	392
27.125	dcm::FileNameEvent Class Reference	392
27.125.1	Detailed Description	393
27.125.2	Member Typedef Documentation	394
27.125.2.1	Self	394
27.125.2.2	Superclass	394
27.125.3	Constructor & Destructor Documentation	394
27.125.3.1	FileNameEvent	394
27.125.3.2	~FileNameEvent	394
27.125.3.3	FileNameEvent	394
27.125.4	Member Function Documentation	394
27.125.4.1	CheckEvent	394
27.125.4.2	GetEventName	394

27.125.4.3	GetFileName	394
27.125.4.4	MakeObject	394
27.125.4.5	SetFileName	394
27.126	dcmm::FilenameGenerator Class Reference	394
27.126.1	Detailed Description	395
27.126.2	Member Typedef Documentation	395
27.126.2.1	FileNamesType	395
27.126.2.2	FilenameType	395
27.126.2.3	SizeType	395
27.126.3	Constructor & Destructor Documentation	395
27.126.3.1	FilenameGenerator	396
27.126.3.2	~FilenameGenerator	396
27.126.4	Member Function Documentation	396
27.126.4.1	Generate	396
27.126.4.2	GetFilename	396
27.126.4.3	GetFileNames	396
27.126.4.4	GetNumberOfFileNames	396
27.126.4.5	GetPattern	396
27.126.4.6	GetPrefix	396
27.126.4.7	SetNumberOfFileNames	396
27.126.4.8	SetPattern	396
27.126.4.9	SetPrefix	397
27.127	dcmm::FileSet Class Reference	397
27.127.1	Detailed Description	397
27.127.2	Member Typedef Documentation	397
27.127.2.1	FilesType	397
27.127.2.2	FileType	397
27.127.3	Constructor & Destructor Documentation	397
27.127.3.1	FileSet	397
27.127.4	Member Function Documentation	397
27.127.4.1	AddFile	398
27.127.4.2	AddFile	398
27.127.4.3	GetFiles	398
27.127.4.4	SetFiles	398
27.127.5	Friends And Related Function Documentation	398
27.127.5.1	operator<<	398
27.128	dcmm::FileStreamer Class Reference	398

27.128.1Detailed Description . . . . .	400
27.128.2Constructor & Destructor Documentation . . . . .	400
27.128.2.1FileStreamer . . . . .	400
27.128.2.2~FileStreamer . . . . .	400
27.128.3Member Function Documentation . . . . .	400
27.128.3.1AppendToDataElement . . . . .	400
27.128.3.2AppendToGroupDataElement . . . . .	400
27.128.3.3CheckDataElement . . . . .	400
27.128.3.4CheckTemplateFileName . . . . .	400
27.128.3.5New . . . . .	401
27.128.3.6ReserveDataElement . . . . .	401
27.128.3.7ReserveGroupDataElement . . . . .	401
27.128.3.8SetOutputFileName . . . . .	401
27.128.3.9SetTemplateFileName . . . . .	401
27.128.3.10StartDataElement . . . . .	401
27.128.3.11StartGroupDataElement . . . . .	401
27.128.3.12StopDataElement . . . . .	401
27.128.3.13StopGroupDataElement . . . . .	401
27.129dcm::FileWithName Class Reference . . . . .	402
27.129.1Detailed Description . . . . .	403
27.129.2Constructor & Destructor Documentation . . . . .	403
27.129.2.1FileWithName . . . . .	403
27.129.3Member Data Documentation . . . . .	403
27.129.3.1filename . . . . .	403
27.130dcm::FindPatientRootQuery Class Reference . . . . .	403
27.130.1Detailed Description . . . . .	405
27.130.2Constructor & Destructor Documentation . . . . .	405
27.130.2.1FindPatientRootQuery . . . . .	405
27.130.3Member Function Documentation . . . . .	405
27.130.3.1GetAbstractSyntaxUID . . . . .	405
27.130.3.2GetTagListByLevel . . . . .	405
27.130.3.3InitializeDataSet . . . . .	405
27.130.3.4ValidateQuery . . . . .	405
27.130.4Friends And Related Function Documentation . . . . .	405
27.130.4.1QueryFactory . . . . .	405
27.131dcm::FindStudyRootQuery Class Reference . . . . .	406
27.131.1Detailed Description . . . . .	407

27.131.2	Constructor & Destructor Documentation	407
27.131.2.1	FindStudyRootQuery	407
27.131.3	Member Function Documentation	407
27.131.3.1	GetAbstractSyntaxUID	407
27.131.3.2	GetTagListByLevel	407
27.131.3.3	InitializeDataSet	407
27.131.3.4	ValidateQuery	407
27.131.4	Friends And Related Function Documentation	407
27.131.4.1	QueryFactory	407
27.132	dcm::Fragment Class Reference	408
27.132.1	Detailed Description	409
27.132.2	Constructor & Destructor Documentation	409
27.132.2.1	Fragment	409
27.132.3	Member Function Documentation	409
27.132.3.1	ComputeLength	409
27.132.3.2	GetLength	409
27.132.3.3	Read	409
27.132.3.4	ReadBacktrack	409
27.132.3.5	ReadPreValue	410
27.132.3.6	ReadValue	410
27.132.3.7	Write	410
27.132.4	Friends And Related Function Documentation	410
27.132.4.1	operator<<	410
27.133	dcm::Global Class Reference	410
27.133.1	Detailed Description	411
27.133.2	Constructor & Destructor Documentation	411
27.133.2.1	Global	411
27.133.2.2	~Global	411
27.133.3	Member Function Documentation	411
27.133.3.1	Append	411
27.133.3.2	GetDefs	411
27.133.3.3	GetDicts	411
27.133.3.4	GetDicts	412
27.133.3.5	GetInstance	412
27.133.3.6	LoadResourcesFiles	412
27.133.3.7	Locate	412
27.133.3.8	Prepend	412

27.133.4	Friends And Related Function Documentation	. 412
27.133.4.1	operator<<	. 412
27.134	dcm::GroupDict Class Reference	. 412
27.134.1	Detailed Description	. 413
27.134.2	Member Typedef Documentation	. 413
27.134.2.1	GroupStringVector	. 413
27.134.3	Constructor & Destructor Documentation	. 413
27.134.3.1	GroupDict	. 413
27.134.3.2	~GroupDict	. 413
27.134.4	Member Function Documentation	. 413
27.134.4.1	Add	. 413
27.134.4.2	GetAbbreviation	. 413
27.134.4.3	GetName	. 414
27.134.4.4	Insert	. 414
27.134.4.5	Size	. 414
27.134.5	Friends And Related Function Documentation	. 414
27.134.5.1	operator<<	. 414
27.135	dcm::IconImageFilter Class Reference	. 414
27.135.1	Detailed Description	. 415
27.135.2	Constructor & Destructor Documentation	. 415
27.135.2.1	IconImageFilter	. 415
27.135.2.2	~IconImageFilter	. 415
27.135.3	Member Function Documentation	. 415
27.135.3.1	Extract	. 415
27.135.3.2	ExtractIconImages	. 415
27.135.3.3	ExtractVeprolIconImages	. 415
27.135.3.4	GetFile	. 416
27.135.3.5	GetFile	. 416
27.135.3.6	GetIconImage	. 416
27.135.3.7	GetNumberOfIconImages	. 416
27.135.3.8	SetFile	. 416
27.136	dcm::IconImageGenerator Class Reference	. 416
27.136.1	Detailed Description	. 417
27.136.2	Constructor & Destructor Documentation	. 417
27.136.2.1	IconImageGenerator	. 417
27.136.2.2	~IconImageGenerator	. 417
27.136.3	Member Function Documentation	. 417

27.136.3.1AutoPixelMinMax . . . . .	417
27.136.3.2ConvertRGBToPaletteColor . . . . .	417
27.136.3.3Generate . . . . .	418
27.136.3.4GetIconImage . . . . .	418
27.136.3.5GetPixmap . . . . .	418
27.136.3.6GetPixmap . . . . .	418
27.136.3.7SetOutputDimensions . . . . .	418
27.136.3.8SetOutsideValuePixel . . . . .	418
27.136.3.9SetPixelMinMax . . . . .	418
27.136.3.10SetPixmap . . . . .	418
27.137.dcm::ignore_char Struct Reference . . . . .	419
27.137.1Constructor & Destructor Documentation . . . . .	419
27.137.1.1ignore_char . . . . .	419
27.137.2Member Data Documentation . . . . .	419
27.137.2.1m_char . . . . .	419
27.138.dcm::Image Class Reference . . . . .	419
27.138.1Detailed Description . . . . .	421
27.138.2Constructor & Destructor Documentation . . . . .	422
27.138.2.1Image . . . . .	422
27.138.2.2~Image . . . . .	422
27.138.3Member Function Documentation . . . . .	422
27.138.3.1GetDirectionCosines . . . . .	422
27.138.3.2GetDirectionCosines . . . . .	422
27.138.3.3GetIntercept . . . . .	422
27.138.3.4GetOrigin . . . . .	422
27.138.3.5GetOrigin . . . . .	422
27.138.3.6GetSlope . . . . .	422
27.138.3.7GetSpacing . . . . .	422
27.138.3.8GetSpacing . . . . .	422
27.138.3.9Print . . . . .	422
27.138.3.10SetDirectionCosines . . . . .	422
27.138.3.11SetDirectionCosines . . . . .	422
27.138.3.12SetDirectionCosines . . . . .	423
27.138.3.13SetIntercept . . . . .	423
27.138.3.14SetOrigin . . . . .	423
27.138.3.15SetOrigin . . . . .	423
27.138.3.16SetOrigin . . . . .	423



27.138.3.1	SetSlope	423
27.138.3.1	SetSpacing	423
27.138.3.1	SetSpacing	423
27.139	gdcm::ImageApplyLookupTable Class Reference	423
27.139.1	Detailed Description	426
27.139.2	Constructor & Destructor Documentation	426
27.139.2.1	ImageApplyLookupTable	426
27.139.2.2	~ImageApplyLookupTable	426
27.139.3	Member Function Documentation	426
27.139.3.1	Apply	426
27.140	gdcm::ImageChangePhotometricInterpretation Class Reference	426
27.140.1	Detailed Description	429
27.140.2	Constructor & Destructor Documentation	429
27.140.2.1	ImageChangePhotometricInterpretation	429
27.140.2.2	~ImageChangePhotometricInterpretation	429
27.140.3	Member Function Documentation	429
27.140.3.1	Change	429
27.140.3.2	ChangeMonochrome	429
27.140.3.3	GetPhotometricInterpretation	429
27.140.3.4	RGB2YBR	429
27.140.3.5	SetPhotometricInterpretation	429
27.140.3.6	YBR2RGB	430
27.141	gdcm::ImageChangePlanarConfiguration Class Reference	430
27.141.1	Detailed Description	432
27.141.2	Constructor & Destructor Documentation	432
27.141.2.1	ImageChangePlanarConfiguration	432
27.141.2.2	~ImageChangePlanarConfiguration	432
27.141.3	Member Function Documentation	432
27.141.3.1	Change	432
27.141.3.2	GetPlanarConfiguration	432
27.141.3.3	RGBPixelsToRGBPlanes	432
27.141.3.4	RGBPlanesToRGBPixels	432
27.141.3.5	SetPlanarConfiguration	432
27.142	gdcm::ImageChangeTransferSyntax Class Reference	433
27.142.1	Detailed Description	435
27.142.2	Constructor & Destructor Documentation	435
27.142.2.1	ImageChangeTransferSyntax	435

27.142.2.2~ImageChangeTransferSyntax . . . . .	435
27.142.3Member Function Documentation . . . . .	435
27.142.3.1Change . . . . .	435
27.142.3.2GetTransferSyntax . . . . .	435
27.142.3.3SetCompressIconImage . . . . .	436
27.142.3.4SetForce . . . . .	436
27.142.3.5SetTransferSyntax . . . . .	436
27.142.3.6SetUserCodec . . . . .	436
27.142.3.7TryJPEG2000Codec . . . . .	436
27.142.3.8TryJPEGCodec . . . . .	436
27.142.3.9TryJPEGLSCodec . . . . .	436
27.142.3.10TryRAWCodec . . . . .	436
27.142.3.11TryRLECodec . . . . .	436
27.143dcm::ImageCodec Class Reference . . . . .	437
27.143.1Detailed Description . . . . .	439
27.143.2Member Typedef Documentation . . . . .	439
27.143.2.1LUTPtr . . . . .	439
27.143.3Constructor & Destructor Documentation . . . . .	439
27.143.3.1ImageCodec . . . . .	439
27.143.3.2~ImageCodec . . . . .	439
27.143.4Member Function Documentation . . . . .	439
27.143.4.1CanCode . . . . .	439
27.143.4.2CanDecode . . . . .	439
27.143.4.3Clone . . . . .	439
27.143.4.4Decode . . . . .	440
27.143.4.5DecodeByStreams . . . . .	440
27.143.4.6DoByteSwap . . . . .	440
27.143.4.7DoInvertMonochrome . . . . .	440
27.143.4.8DoOverlayCleanup . . . . .	440
27.143.4.9DoPaddedCompositePixelCode . . . . .	440
27.143.4.10DoPlanarConfiguration . . . . .	440
27.143.4.11DoSimpleCopy . . . . .	440
27.143.4.12DoYBR . . . . .	440
27.143.4.13GetDimensions . . . . .	440
27.143.4.14GetHeaderInfo . . . . .	440
27.143.4.15GetLossyFlag . . . . .	440
27.143.4.16GetLUT . . . . .	440

27.143.4.1	<a href="#">GetNeedByteSwap</a>	440
27.143.4.1	<a href="#">GetNumberOfDimensions</a>	440
27.143.4.1	<a href="#">GetPhotometricInterpretation</a>	440
27.143.4.2	<a href="#">GetPixelFormat</a>	441
27.143.4.2	<a href="#">GetPixelFormat</a>	441
27.143.4.2	<a href="#">GetPlanarConfiguration</a>	441
27.143.4.2	<a href="#">Lossy</a>	441
27.143.4.2	<a href="#">Valid</a>	441
27.143.4.2	<a href="#">SetDimensions</a>	441
27.143.4.2	<a href="#">SetDimensions</a>	441
27.143.4.2	<a href="#">SetLossyFlag</a>	441
27.143.4.2	<a href="#">SetLUT</a>	441
27.143.4.2	<a href="#">SetNeedByteSwap</a>	441
27.143.4.3	<a href="#">SetNeedOverlayCleanup</a>	441
27.143.4.3	<a href="#">SetNumberOfDimensions</a>	441
27.143.4.3	<a href="#">SetPhotometricInterpretation</a>	441
27.143.4.3	<a href="#">SetPixelFormat</a>	441
27.143.4.3	<a href="#">SetPlanarConfiguration</a>	442
27.143.5	<a href="#">Friends And Related Function Documentation</a>	442
27.143.5.1	<a href="#">ImageChangePhotometricInterpretation</a>	442
27.143.6	<a href="#">Member Data Documentation</a>	442
27.143.6.1	<a href="#">Dimensions</a>	442
27.143.6.2	<a href="#">LossyFlag</a>	442
27.143.6.3	<a href="#">LUT</a>	442
27.143.6.4	<a href="#">NeedByteSwap</a>	442
27.143.6.5	<a href="#">NeedOverlayCleanup</a>	442
27.143.6.6	<a href="#">NumberOfDimensions</a>	442
27.143.6.7	<a href="#">PF</a>	442
27.143.6.8	<a href="#">PI</a>	442
27.143.6.9	<a href="#">PlanarConfiguration</a>	442
27.143.6.10	<a href="#">RequestPaddedCompositePixelCode</a>	442
27.143.6.11	<a href="#">RequestPlanarConfiguration</a>	442
27.144	<a href="#">dcm::ImageConverter Class Reference</a>	442
27.144.1	<a href="#">Detailed Description</a>	443
27.144.2	<a href="#">Constructor &amp; Destructor Documentation</a>	443
27.144.2.1	<a href="#">ImageConverter</a>	443
27.144.2.2	<a href="#">~ImageConverter</a>	443

27.144.3	Member Function Documentation	443
27.144.3.1	Convert	443
27.144.3.2	GetOutput	443
27.144.3.3	SetInput	443
27.145	gdcm::ImageFragmentSplitter Class Reference	443
27.145.1	Detailed Description	446
27.145.2	Constructor & Destructor Documentation	446
27.145.2.1	ImageFragmentSplitter	446
27.145.2.2	~ImageFragmentSplitter	446
27.145.3	Member Function Documentation	446
27.145.3.1	GetFragmentSizeMax	446
27.145.3.2	SetForce	446
27.145.3.3	SetFragmentSizeMax	446
27.145.3.4	Split	446
27.146	gdcm::ImageHelper Class Reference	446
27.146.1	Detailed Description	447
27.146.2	Member Function Documentation	447
27.146.2.1	ComputeSpacingFromImagePositionPatient	447
27.146.2.2	GetDimensionsValue	448
27.146.2.3	GetDirectionCosinesFromDataSet	448
27.146.2.4	GetDirectionCosinesValue	448
27.146.2.5	GetForcePixelSpacing	448
27.146.2.6	GetForceRescaleInterceptSlope	448
27.146.2.7	GetLUT	448
27.146.2.8	GetOriginValue	448
27.146.2.9	GetPhotometricInterpretationValue	448
27.146.2.10	GetPixelFormatValue	448
27.146.2.11	GetPlanarConfigurationValue	448
27.146.2.12	GetPointerFromElement	448
27.146.2.13	GetRescaleInterceptSlopeValue	448
27.146.2.14	GetSpacingTagFromMediaStorage	449
27.146.2.15	GetSpacingValue	449
27.146.2.16	GetZSpacingTagFromMediaStorage	449
27.146.2.17	SetDimensionsValue	449
27.146.2.18	SetDirectionCosinesValue	449
27.146.2.19	SetForcePixelSpacing	449
27.146.2.20	SetForceRescaleInterceptSlope	449

27.146.2.2	SetOriginValue . . . . .	449
27.146.2.2	SetRescaleInterceptSlopeValue . . . . .	449
27.146.2.2	SetSpacingValue . . . . .	449
27.147	gdcm::ImageReader Class Reference . . . . .	449
27.147.1	Detailed Description . . . . .	452
27.147.2	Constructor & Destructor Documentation . . . . .	452
27.147.2.1	ImageReader . . . . .	452
27.147.2.2	~ImageReader . . . . .	452
27.147.3	Member Function Documentation . . . . .	452
27.147.3.1	GetImage . . . . .	452
27.147.3.2	GetImage . . . . .	452
27.147.3.3	Read . . . . .	452
27.147.3.4	ReadACRNEMAIImage . . . . .	453
27.147.3.5	ReadImage . . . . .	453
27.148	gdcm::ImageRegionReader Class Reference . . . . .	453
27.148.1	Detailed Description . . . . .	455
27.148.2	Constructor & Destructor Documentation . . . . .	455
27.148.2.1	ImageRegionReader . . . . .	455
27.148.2.2	~ImageRegionReader . . . . .	455
27.148.3	Member Function Documentation . . . . .	455
27.148.3.1	ComputeBufferLength . . . . .	455
27.148.3.2	GetRegion . . . . .	455
27.148.3.3	Read . . . . .	455
27.148.3.4	ReadInformation . . . . .	455
27.148.3.5	ReadIntoBuffer . . . . .	456
27.148.3.6	SetRegion . . . . .	456
27.149	gdcm::ImageToImageFilter Class Reference . . . . .	456
27.149.1	Detailed Description . . . . .	457
27.149.2	Constructor & Destructor Documentation . . . . .	458
27.149.2.1	ImageToImageFilter . . . . .	458
27.149.2.2	~ImageToImageFilter . . . . .	458
27.149.3	Member Function Documentation . . . . .	458
27.149.3.1	GetInput . . . . .	458
27.149.3.2	GetOutput . . . . .	458
27.150	gdcm::ImageWriter Class Reference . . . . .	458
27.150.1	Detailed Description . . . . .	460
27.150.2	Constructor & Destructor Documentation . . . . .	460

27.150.2.1ImageWriter . . . . .	460
27.150.2.2~ImageWriter . . . . .	460
27.150.3Member Function Documentation . . . . .	460
27.150.3.1GetImage . . . . .	460
27.150.3.2GetImage . . . . .	460
27.150.3.3Write . . . . .	460
27.151dcm::network::ImplementationClassUIDSub Class Reference . . . . .	461
27.151.1Detailed Description . . . . .	461
27.151.2Constructor & Destructor Documentation . . . . .	461
27.151.2.1ImplementationClassUIDSub . . . . .	461
27.151.3Member Function Documentation . . . . .	461
27.151.3.1Print . . . . .	461
27.151.3.2Read . . . . .	461
27.151.3.3Size . . . . .	461
27.151.3.4Write . . . . .	461
27.152dcm::network::ImplementationUIDSub Class Reference . . . . .	461
27.152.1Detailed Description . . . . .	462
27.152.2Constructor & Destructor Documentation . . . . .	462
27.152.2.1ImplementationUIDSub . . . . .	462
27.152.3Member Function Documentation . . . . .	462
27.152.3.1Write . . . . .	462
27.153dcm::network::ImplementationVersionNameSub Class Reference . . . . .	462
27.153.1Detailed Description . . . . .	462
27.153.2Constructor & Destructor Documentation . . . . .	462
27.153.2.1ImplementationVersionNameSub . . . . .	462
27.153.3Member Function Documentation . . . . .	462
27.153.3.1Print . . . . .	462
27.153.3.2Read . . . . .	462
27.153.3.3Size . . . . .	462
27.153.3.4Write . . . . .	463
27.154dcm::ImplicitDataElement Class Reference . . . . .	463
27.154.1Detailed Description . . . . .	464
27.154.2Member Function Documentation . . . . .	465
27.154.2.1GetLength . . . . .	465
27.154.2.2Read . . . . .	465
27.154.2.3ReadPreValue . . . . .	465
27.154.2.4ReadValue . . . . .	465

27.154.2.5	ReadValueWithLength . . . . .	465
27.154.2.6	ReadWithLength . . . . .	465
27.154.2.7	Write . . . . .	465
27.155	dcm::InitializeEvent Class Reference . . . . .	465
27.156	dcm::IOD Class Reference . . . . .	467
27.156.1	Detailed Description . . . . .	467
27.156.2	Member Typedef Documentation . . . . .	467
27.156.2.1	MapIODEntry . . . . .	467
27.156.2.2	SizeType . . . . .	467
27.156.3	Constructor & Destructor Documentation . . . . .	467
27.156.3.1	IOD . . . . .	467
27.156.4	Member Function Documentation . . . . .	468
27.156.4.1	AddIODEntry . . . . .	468
27.156.4.2	Clear . . . . .	468
27.156.4.3	GetIODEntry . . . . .	468
27.156.4.4	GetNumberOfIODs . . . . .	468
27.156.4.5	GetTypeFromTag . . . . .	468
27.156.5	Friends And Related Function Documentation . . . . .	468
27.156.5.1	operator<< . . . . .	468
27.157	dcm::IODEntry Class Reference . . . . .	468
27.157.1	Detailed Description . . . . .	469
27.157.2	Constructor & Destructor Documentation . . . . .	469
27.157.2.1	IODEntry . . . . .	469
27.157.3	Member Function Documentation . . . . .	469
27.157.3.1	GetIE . . . . .	469
27.157.3.2	GetName . . . . .	469
27.157.3.3	GetRef . . . . .	469
27.157.3.4	GetUsage . . . . .	470
27.157.3.5	GetUsageType . . . . .	470
27.157.3.6	SetIE . . . . .	470
27.157.3.7	SetName . . . . .	470
27.157.3.8	SetRef . . . . .	470
27.157.3.9	SetUsage . . . . .	470
27.157.4	Friends And Related Function Documentation . . . . .	470
27.157.4.1	operator<< . . . . .	470
27.158	dcm::IODs Class Reference . . . . .	470
27.158.1	Detailed Description . . . . .	471

27.158.2	Member Typedef Documentation . . . . .	471
27.158.2.1	IODMapType . . . . .	471
27.158.2.2	IODMapTypeConstIterator . . . . .	471
27.158.2.3	IODName . . . . .	471
27.158.3	Constructor & Destructor Documentation . . . . .	471
27.158.3.1	IODs . . . . .	471
27.158.4	Member Function Documentation . . . . .	471
27.158.4.1	AddIOD . . . . .	471
27.158.4.2	Begin . . . . .	471
27.158.4.3	Clear . . . . .	471
27.158.4.4	End . . . . .	471
27.158.4.5	GetIOD . . . . .	471
27.158.5	Friends And Related Function Documentation . . . . .	471
27.158.5.1	operator<< . . . . .	471
27.159	dcm::IPPSorter Class Reference . . . . .	471
27.159.1	Detailed Description . . . . .	473
27.159.2	Constructor & Destructor Documentation . . . . .	473
27.159.2.1	IPPSorter . . . . .	473
27.159.3	Member Function Documentation . . . . .	473
27.159.3.1	GetDirectionCosinesTolerance . . . . .	473
27.159.3.2	GetZSpacing . . . . .	473
27.159.3.3	GetZSpacingTolerance . . . . .	474
27.159.3.4	SetComputeZSpacing . . . . .	474
27.159.3.5	SetDirectionCosinesTolerance . . . . .	474
27.159.3.6	SetDropDuplicatePositions . . . . .	474
27.159.3.7	SetZSpacingTolerance . . . . .	474
27.159.3.8	Sort . . . . .	474
27.159.4	Member Data Documentation . . . . .	475
27.159.4.1	ComputeZSpacing . . . . .	475
27.159.4.2	DirCosTolerance . . . . .	475
27.159.4.3	DropDuplicatePositions . . . . .	475
27.159.4.4	ZSpacing . . . . .	475
27.159.4.5	ZTolerance . . . . .	475
27.160	dcm::Item Class Reference . . . . .	475
27.160.1	Detailed Description . . . . .	477
27.160.2	Constructor & Destructor Documentation . . . . .	477
27.160.2.1	Item . . . . .	477



27.160.2.2Item . . . . .	477
27.160.3Member Function Documentation . . . . .	477
27.160.3.1Clear . . . . .	477
27.160.3.2FindDataElement . . . . .	477
27.160.3.3GetDataElement . . . . .	477
27.160.3.4GetLength . . . . .	477
27.160.3.5GetNestedDataSet . . . . .	477
27.160.3.6GetNestedDataSet . . . . .	478
27.160.3.7InsertDataElement . . . . .	478
27.160.3.8Read . . . . .	478
27.160.3.9SetNestedDataSet . . . . .	478
27.160.3.10Write . . . . .	478
27.160.4Friends And Related Function Documentation . . . . .	478
27.160.4.1operator<< . . . . .	478
27.161gdcmm::IterationEvent Class Reference . . . . .	478
27.162gdcmm::JPEG12Codec Class Reference . . . . .	480
27.162.1Detailed Description . . . . .	481
27.162.2Constructor & Destructor Documentation . . . . .	481
27.162.2.1JPEG12Codec . . . . .	481
27.162.2.2~JPEG12Codec . . . . .	481
27.162.3Member Function Documentation . . . . .	481
27.162.3.1DecodeByStreams . . . . .	481
27.162.3.2EncodeBuffer . . . . .	481
27.162.3.3GetHeaderInfo . . . . .	481
27.162.3.4InternalCode . . . . .	481
27.162.3.5IsStateSuspension . . . . .	481
27.163gdcmm::JPEG16Codec Class Reference . . . . .	482
27.163.1Detailed Description . . . . .	483
27.163.2Constructor & Destructor Documentation . . . . .	483
27.163.2.1JPEG16Codec . . . . .	483
27.163.2.2~JPEG16Codec . . . . .	483
27.163.3Member Function Documentation . . . . .	483
27.163.3.1DecodeByStreams . . . . .	483
27.163.3.2EncodeBuffer . . . . .	483
27.163.3.3GetHeaderInfo . . . . .	483
27.163.3.4InternalCode . . . . .	483
27.163.3.5IsStateSuspension . . . . .	484

27.164	dcm::JPEG2000Codec Class Reference	484
27.164.1	Detailed Description	485
27.164.2	Constructor & Destructor Documentation	486
27.164.2.1	JPEG2000Codec	486
27.164.2.2	~JPEG2000Codec	486
27.164.3	Member Function Documentation	486
27.164.3.1	CanCode	486
27.164.3.2	CanDecode	486
27.164.3.3	Clone	486
27.164.3.4	Code	486
27.164.3.5	Decode	486
27.164.3.6	DecodeByStreams	486
27.164.3.7	DecodeExtent	486
27.164.3.8	GetHeaderInfo	486
27.164.3.9	GetQuality	487
27.164.3.10	GetRate	487
27.164.3.11	SetNumberOfResolutions	487
27.164.3.12	SetQuality	487
27.164.3.13	SetRate	487
27.164.3.14	SetReversible	487
27.164.3.15	SetTileSize	487
27.164.4	Friends And Related Function Documentation	487
27.164.4.1	Bitmap	487
27.164.4.2	ImageRegionReader	487
27.165	dcm::JPEG8Codec Class Reference	487
27.165.1	Detailed Description	489
27.165.2	Constructor & Destructor Documentation	489
27.165.2.1	JPEG8Codec	489
27.165.2.2	~JPEG8Codec	489
27.165.3	Member Function Documentation	489
27.165.3.1	DecodeByStreams	489
27.165.3.2	EncodeBuffer	489
27.165.3.3	GetHeaderInfo	489
27.165.3.4	InternalCode	489
27.165.3.5	IsStateSuspension	489
27.166	dcm::JPEGCodec Class Reference	490
27.166.1	Detailed Description	491

27.166.2	Constructor & Destructor Documentation	492
27.166.2.1	JPEGCodec	492
27.166.2.2	~JPGCodec	492
27.166.3	Member Function Documentation	492
27.166.3.1	CanCode	492
27.166.3.2	CanDecode	492
27.166.3.3	Clone	492
27.166.3.4	Code	492
27.166.3.5	ComputeOffsetTable	492
27.166.3.6	Decode	493
27.166.3.7	DecodeByStreams	493
27.166.3.8	DecodeExtent	493
27.166.3.9	EncodeBuffer	493
27.166.3.10	GetHeaderInfo	493
27.166.3.11	GetLossless	493
27.166.3.12	GetQuality	493
27.166.3.13	StateSuspension	493
27.166.3.14	Valid	493
27.166.3.15	SetBitSample	493
27.166.3.16	SetLossless	493
27.166.3.17	SetPixelFormat	493
27.166.3.18	SetQuality	494
27.166.4	Friends And Related Function Documentation	494
27.166.4.1	ImageRegionReader	494
27.166.5	Member Data Documentation	494
27.166.5.1	BitSample	494
27.166.5.2	Lossless	494
27.166.5.3	Quality	494
27.167	gdcm::JPEGLSCodec Class Reference	494
27.167.1	Detailed Description	496
27.167.2	Constructor & Destructor Documentation	496
27.167.2.1	JPEGLSCodec	496
27.167.2.2	~JPEGLSCodec	496
27.167.3	Member Function Documentation	496
27.167.3.1	CanCode	496
27.167.3.2	CanDecode	496
27.167.3.3	Clone	496

27.167.3.4	Code	. 496
27.167.3.5	Decode	. 496
27.167.3.6	Decode	. 496
27.167.3.7	DecodeExtent	. 497
27.167.3.8	GetBufferLength	. 497
27.167.3.9	GetHeaderInfo	. 497
27.167.3.10	GetLossless	. 497
27.167.3.11	SetBufferLength	. 497
27.167.3.12	SetLossless	. 497
27.167.3.13	SetLossyError	. 497
27.167.4	Friends And Related Function Documentation	. 497
27.167.4.1	ImageRegionReader	. 497
27.168	dcm::JSON Class Reference	. 497
27.168.1	Detailed Description	. 497
27.168.2	Constructor & Destructor Documentation	. 498
27.168.2.1	JSON	. 498
27.168.2.2	~JSON	. 498
27.168.3	Member Function Documentation	. 498
27.168.3.1	Code	. 498
27.168.3.2	Decode	. 498
27.168.3.3	GetPrettyPrint	. 498
27.168.3.4	PrettyPrintOff	. 498
27.168.3.5	PrettyPrintOn	. 498
27.168.3.6	SetPrettyPrint	. 498
27.169	dcm::KAKADUCodec Class Reference	. 498
27.169.1	Detailed Description	. 500
27.169.2	Constructor & Destructor Documentation	. 500
27.169.2.1	KAKADUCodec	. 500
27.169.2.2	~KAKADUCodec	. 500
27.169.3	Member Function Documentation	. 500
27.169.3.1	CanCode	. 500
27.169.3.2	CanDecode	. 500
27.169.3.3	Clone	. 500
27.169.3.4	Code	. 500
27.169.3.5	Decode	. 500
27.170	dcm::LO Class Reference	. 501
27.170.1	Detailed Description	. 502

27.170.2	Member Typedef Documentation	502
27.170.2.1	const_iterator	502
27.170.2.2	const_reference	502
27.170.2.3	const_reverse_iterator	502
27.170.2.4	difference_type	502
27.170.2.5	iterator	502
27.170.2.6	pointer	502
27.170.2.7	reference	502
27.170.2.8	reverse_iterator	502
27.170.2.9	size_type	502
27.170.2.10	superclass	502
27.170.2.11	value_type	502
27.170.3	Constructor & Destructor Documentation	502
27.170.3.1	LO	502
27.170.3.2	LO	502
27.170.3.3	LO	503
27.170.3.4	LO	503
27.170.4	Member Function Documentation	503
27.170.4.1	IsValid	503
27.171	gdcmm::LookupTable Class Reference	503
27.171.1	Detailed Description	505
27.171.2	Member Enumeration Documentation	505
27.171.2.1	LookupTableType	505
27.171.3	Constructor & Destructor Documentation	505
27.171.3.1	LookupTable	505
27.171.3.2	~LookupTable	505
27.171.3.3	LookupTable	506
27.171.4	Member Function Documentation	506
27.171.4.1	Allocate	506
27.171.4.2	Clear	506
27.171.4.3	Decode	506
27.171.4.4	Decode	506
27.171.4.5	GetBitSample	506
27.171.4.6	GetBufferAsRGBA	506
27.171.4.7	GetLUT	506
27.171.4.8	GetLUTDescriptor	506
27.171.4.9	GetLUTLength	506

27.171.4.10	GetPointer	. 506
27.171.4.11	InitializeBlueLUT	. 506
27.171.4.12	Initialized	. 506
27.171.4.13	InitializeGreenLUT	. 507
27.171.4.14	InitializeLUT	. 507
27.171.4.15	InitializeRedLUT	. 507
27.171.4.16	Print	. 507
27.171.4.17	SetBlueLUT	. 507
27.171.4.18	SetGreenLUT	. 507
27.171.4.19	SetLUT	. 507
27.171.4.20	SetRedLUT	. 507
27.171.4.21	WriteBufferAsRGBA	. 507
27.171.5	Member Data Documentation	. 507
27.171.5.1	BitSample	. 507
27.171.5.2	IncompleteLUT	. 507
27.171.5.3	Internal	. 507
27.172	dcm::Scanner::Itstr Struct Reference	. 507
27.172.1	Member Function Documentation	. 508
27.172.1.1	operator()	. 508
27.173	dcm::Macro Class Reference	. 508
27.173.1	Detailed Description	. 508
27.173.2	Member Typedef Documentation	. 509
27.173.2.1	ArrayIncludeMacrosType	. 509
27.173.2.2	MapModuleEntry	. 509
27.173.3	Constructor & Destructor Documentation	. 509
27.173.3.1	Macro	. 509
27.173.4	Member Function Documentation	. 509
27.173.4.1	AddMacroEntry	. 509
27.173.4.2	Clear	. 509
27.173.4.3	FindMacroEntry	. 509
27.173.4.4	GetMacroEntry	. 509
27.173.4.5	GetName	. 509
27.173.4.6	SetName	. 509
27.173.4.7	Verify	. 509
27.173.5	Friends And Related Function Documentation	. 509
27.173.5.1	operator<<	. 509
27.174	dcm::Macros Class Reference	. 509

27.174.1Detailed Description . . . . .	510
27.174.2Member Typedef Documentation . . . . .	510
27.174.2.1ModuleMapType . . . . .	510
27.174.3Constructor & Destructor Documentation . . . . .	510
27.174.3.1Macros . . . . .	510
27.174.4Member Function Documentation . . . . .	510
27.174.4.1AddMacro . . . . .	510
27.174.4.2Clear . . . . .	510
27.174.4.3GetMacro . . . . .	510
27.174.4.4IsEmpty . . . . .	510
27.174.5Friends And Related Function Documentation . . . . .	511
27.174.5.1operator<< . . . . .	511
27.175dcm::network::MaximumLengthSub Class Reference . . . . .	511
27.175.1Detailed Description . . . . .	511
27.175.2Constructor & Destructor Documentation . . . . .	511
27.175.2.1MaximumLengthSub . . . . .	511
27.175.3Member Function Documentation . . . . .	511
27.175.3.1GetMaximumLength . . . . .	511
27.175.3.2Print . . . . .	511
27.175.3.3Read . . . . .	511
27.175.3.4SetMaximumLength . . . . .	511
27.175.3.5Size . . . . .	511
27.175.3.6Write . . . . .	511
27.176dcm::MD5 Class Reference . . . . .	512
27.176.1Detailed Description . . . . .	512
27.176.2Constructor & Destructor Documentation . . . . .	512
27.176.2.1MD5 . . . . .	512
27.176.2.2~MD5 . . . . .	512
27.176.3Member Function Documentation . . . . .	512
27.176.3.1Compute . . . . .	512
27.176.3.2ComputeFile . . . . .	512
27.177dcm::MediaStorage Class Reference . . . . .	513
27.177.1Detailed Description . . . . .	516
27.177.2Member Enumeration Documentation . . . . .	516
27.177.2.1MSType . . . . .	516
27.177.2.2ObjectType . . . . .	518
27.177.3Constructor & Destructor Documentation . . . . .	519

27.177.3.1MediaStorage . . . . .	519
27.177.4Member Function Documentation . . . . .	519
27.177.4.1GetModality . . . . .	519
27.177.4.2GetModalityDimension . . . . .	519
27.177.4.3GetMSString . . . . .	519
27.177.4.4GetMSType . . . . .	519
27.177.4.5GetNumberOfModality . . . . .	519
27.177.4.6GetNumberOfMSString . . . . .	519
27.177.4.7GetNumberOfMSType . . . . .	519
27.177.4.8GetString . . . . .	519
27.177.4.9GuessFromModality . . . . .	519
27.177.4.10Image . . . . .	519
27.177.4.11\$Undefined . . . . .	520
27.177.4.12operator MSType . . . . .	520
27.177.4.13SetFromDataSet . . . . .	520
27.177.4.14SetFromFile . . . . .	520
27.177.4.15SetFromHeader . . . . .	520
27.177.4.16SetFromModality . . . . .	520
27.177.4.17SetFromSourceImageSequence . . . . .	520
27.177.5Friends And Related Function Documentation . . . . .	520
27.177.5.1operator<< . . . . .	520
27.178dcm::MemberCommand< T > Class Template Reference . . . . .	520
27.178.1Detailed Description . . . . .	522
27.178.2Member Typedef Documentation . . . . .	522
27.178.2.1Self . . . . .	522
27.178.2.2TConstMemberFunctionPointer . . . . .	523
27.178.2.3TMemberFunctionPointer . . . . .	523
27.178.3Constructor & Destructor Documentation . . . . .	523
27.178.3.1MemberCommand . . . . .	523
27.178.3.2~MemberCommand . . . . .	523
27.178.4Member Function Documentation . . . . .	523
27.178.4.1Execute . . . . .	523
27.178.4.2Execute . . . . .	523
27.178.4.3New . . . . .	523
27.178.4.4SetCallbackFunction . . . . .	523
27.178.4.5SetCallbackFunction . . . . .	524
27.178.5Member Data Documentation . . . . .	524



27.178.5.1m_ConstMemberFunction . . . . .	524
27.178.5.2m_MemberFunction . . . . .	524
27.178.5.3m_This . . . . .	524
27.179dcm::MeshPrimitive Class Reference . . . . .	524
27.179.1Detailed Description . . . . .	526
27.179.2Member Typedef Documentation . . . . .	526
27.179.2.1PrimitivesData . . . . .	526
27.179.3Member Enumeration Documentation . . . . .	526
27.179.3.1MPType . . . . .	526
27.179.4Constructor & Destructor Documentation . . . . .	527
27.179.4.1MeshPrimitive . . . . .	527
27.179.4.2~MeshPrimitive . . . . .	527
27.179.5Member Function Documentation . . . . .	527
27.179.5.1AddPrimitiveData . . . . .	527
27.179.5.2GetMPType . . . . .	527
27.179.5.3GetMPTypeString . . . . .	527
27.179.5.4GetNumberOfPrimitivesData . . . . .	527
27.179.5.5GetPrimitiveData . . . . .	527
27.179.5.6GetPrimitiveData . . . . .	527
27.179.5.7GetPrimitiveData . . . . .	527
27.179.5.8GetPrimitiveData . . . . .	527
27.179.5.9GetPrimitivesData . . . . .	527
27.179.5.10GetPrimitivesData . . . . .	527
27.179.5.11GetPrimitiveType . . . . .	527
27.179.5.12SetPrimitiveData . . . . .	527
27.179.5.13SetPrimitiveData . . . . .	527
27.179.5.14SetPrimitivesData . . . . .	527
27.179.5.15SetPrimitiveType . . . . .	527
27.179.6Member Data Documentation . . . . .	527
27.179.6.1PrimitiveData . . . . .	527
27.179.6.2PrimitiveType . . . . .	527
27.180dcm::ModifiedEvent Class Reference . . . . .	527
27.181dcm::Module Class Reference . . . . .	529
27.181.1Detailed Description . . . . .	529
27.181.2Member Typedef Documentation . . . . .	530
27.181.2.1ArrayIncludeMacrosType . . . . .	530
27.181.2.2MapModuleEntry . . . . .	530

27.181.3	Constructor & Destructor Documentation	530
27.181.3.1	Module	530
27.181.4	Member Function Documentation	530
27.181.4.1	AddMacro	530
27.181.4.2	AddModuleEntry	530
27.181.4.3	Clear	530
27.181.4.4	FindModuleEntryInMacros	530
27.181.4.5	GetModuleEntryInMacros	530
27.181.4.6	GetName	530
27.181.4.7	SetName	530
27.181.4.8	Verify	530
27.181.5	Friends And Related Function Documentation	530
27.181.5.1	operator<<	530
27.182	dcm::ModuleEntry Class Reference	531
27.182.1	Detailed Description	532
27.182.2	Member Typedef Documentation	532
27.182.2.1	Description	532
27.182.3	Constructor & Destructor Documentation	532
27.182.3.1	ModuleEntry	532
27.182.3.2	~ModuleEntry	533
27.182.4	Member Function Documentation	533
27.182.4.1	GetDescription	533
27.182.4.2	GetName	533
27.182.4.3	GetType	533
27.182.4.4	SetDescription	533
27.182.4.5	SetName	533
27.182.4.6	SetType	533
27.182.5	Friends And Related Function Documentation	533
27.182.5.1	operator<<	533
27.182.6	Member Data Documentation	533
27.182.6.1	DataElementType	533
27.182.6.2	DescriptionField	533
27.182.6.3	Name	533
27.183	dcm::Modules Class Reference	533
27.183.1	Detailed Description	534
27.183.2	Member Typedef Documentation	534
27.183.2.1	ModuleMapType	534

27.183.3	Constructor & Destructor Documentation	534
27.183.3.1	Modules	534
27.183.4	Member Function Documentation	534
27.183.4.1	AddModule	534
27.183.4.2	Clear	534
27.183.4.3	GetModule	534
27.183.4.4	IsEmpty	534
27.183.5	Friends And Related Function Documentation	535
27.183.5.1	operator<<	535
27.184	dcm::MovePatientRootQuery Class Reference	535
27.184.1	Detailed Description	536
27.184.2	Constructor & Destructor Documentation	536
27.184.2.1	MovePatientRootQuery	536
27.184.3	Member Function Documentation	536
27.184.3.1	GetAbstractSyntaxUID	536
27.184.3.2	GetTagListByLevel	536
27.184.3.3	InitializeDataSet	536
27.184.3.4	ValidateQuery	536
27.184.4	Friends And Related Function Documentation	537
27.184.4.1	QueryFactory	537
27.185	dcm::MoveStudyRootQuery Class Reference	537
27.185.1	Detailed Description	538
27.185.2	Constructor & Destructor Documentation	538
27.185.2.1	MoveStudyRootQuery	538
27.185.3	Member Function Documentation	538
27.185.3.1	GetAbstractSyntaxUID	538
27.185.3.2	GetTagListByLevel	539
27.185.3.3	InitializeDataSet	539
27.185.3.4	ValidateQuery	539
27.185.4	Friends And Related Function Documentation	539
27.185.4.1	QueryFactory	539
27.186	dcm::NestedModuleEntries Class Reference	539
27.186.1	Detailed Description	541
27.186.2	Member Typedef Documentation	541
27.186.2.1	SizeType	541
27.186.3	Constructor & Destructor Documentation	541
27.186.3.1	NestedModuleEntries	541

27.186.4	Member Function Documentation	541
27.186.4.1	AddModuleEntry	541
27.186.4.2	GetModuleEntry	541
27.186.4.3	GetModuleEntry	541
27.186.4.4	GetNumberOfModuleEntries	541
27.186.5	Friends And Related Function Documentation	541
27.186.5.1	operator<<	541
27.187	dcm::NoEvent Class Reference	542
27.187.1	Detailed Description	542
27.188	dcm::Object Class Reference	542
27.188.1	Detailed Description	544
27.188.2	Constructor & Destructor Documentation	544
27.188.2.1	Object	544
27.188.2.2	~Object	544
27.188.2.3	Object	544
27.188.3	Member Function Documentation	544
27.188.3.1	operator=	544
27.188.3.2	Print	544
27.188.3.3	Register	544
27.188.3.4	UnRegister	544
27.188.4	Friends And Related Function Documentation	544
27.188.4.1	operator<<	545
27.188.4.2	SmartPointer	545
27.189	dcm::OpenSSLCryptoFactory Class Reference	545
27.189.1	Constructor & Destructor Documentation	546
27.189.1.1	OpenSSLCryptoFactory	546
27.189.2	Member Function Documentation	546
27.189.2.1	CreateCMSProvider	546
27.189.2.2	InitOpenSSL	546
27.190	dcm::OpenSSLCryptographicMessageSyntax Class Reference	546
27.190.1	Constructor & Destructor Documentation	547
27.190.1.1	OpenSSLCryptographicMessageSyntax	547
27.190.1.2	~OpenSSLCryptographicMessageSyntax	547
27.190.2	Member Function Documentation	547
27.190.2.1	Decrypt	547
27.190.2.2	Encrypt	548
27.190.2.3	GetCipherType	548

27.190.2.4	ParseCertificateFile	548
27.190.2.5	ParseKeyFile	548
27.190.2.6	SetCipherType	548
27.190.2.7	SetPassword	548
27.190	gdcmm::OpenSSLP7CryptoFactory Class Reference	548
27.191	Constructor & Destructor Documentation	549
27.191.1.1	OpenSSLP7CryptoFactory	549
27.191.2	Member Function Documentation	549
27.191.2.1	CreateCMSProvider	550
27.190	gdcmm::OpenSSLP7CryptographicMessageSyntax Class Reference	550
27.192	Detailed Description	551
27.192.2	Constructor & Destructor Documentation	551
27.192.2.1	OpenSSLP7CryptographicMessageSyntax	551
27.192.2.2	~OpenSSLP7CryptographicMessageSyntax	551
27.192.3	Member Function Documentation	551
27.192.3.1	Decrypt	551
27.192.3.2	Encrypt	551
27.192.3.3	GetCipherType	551
27.192.3.4	ParseCertificateFile	552
27.192.3.5	ParseKeyFile	552
27.192.3.6	SetCipherType	552
27.192.3.7	SetPassword	552
27.190	gdcmm::Orientation Class Reference	552
27.193	Detailed Description	553
27.193.2	Member Enumeration Documentation	553
27.193.2.1	OrientationType	553
27.193.3	Constructor & Destructor Documentation	553
27.193.3.1	Orientation	553
27.193.3.2	~Orientation	553
27.193.4	Member Function Documentation	553
27.193.4.1	GetLabel	553
27.193.4.2	GetMajorAxisFromPatientRelativeDirectionCosine	554
27.193.4.3	GetObliquityThresholdCosineValue	554
27.193.4.4	GetType	554
27.193.4.5	Print	554
27.193.4.6	SetObliquityThresholdCosineValue	554
27.193.5	Friends And Related Function Documentation	554

27.193.5.operator<<	554
27.194.dcm::Overlay Class Reference	554
27.194.1.Detailed Description	557
27.194.2.Member Enumeration Documentation	557
27.194.2.1OverlayType	557
27.194.3.Constructor & Destructor Documentation	557
27.194.3.1Overlay	557
27.194.3.2~Overlay	557
27.194.3.3Overlay	557
27.194.4.Member Function Documentation	557
27.194.4.1Decompress	557
27.194.4.2GetBitPosition	557
27.194.4.3GetBitsAllocated	557
27.194.4.4GetColumns	557
27.194.4.5GetDescription	558
27.194.4.6GetGroup	558
27.194.4.7GetOrigin	558
27.194.4.8GetOverlayData	558
27.194.4.9GetOverlayTypeAsString	558
27.194.4.10GetOverlayTypeFromString	558
27.194.4.11GetRows	558
27.194.4.12GetType	558
27.194.4.13GetTypeAsEnum	558
27.194.4.14GetUnpackBuffer	558
27.194.4.15GetUnpackBufferLength	558
27.194.4.16GrabOverlayFromPixelData	558
27.194.4.17IsEmpty	558
27.194.4.18InPixelData	558
27.194.4.19InPixelData	559
27.194.4.20Zero	559
27.194.4.21Print	559
27.194.4.22SetBitPosition	559
27.194.4.23SetBitsAllocated	559
27.194.4.24SetColumns	559
27.194.4.25SetDescription	559
27.194.4.26SetFrameOrigin	559
27.194.4.27SetGroup	559

27.194.4.28	SetNumberOfFrames	559
27.194.4.29	SetOrigin	559
27.194.4.30	SetOverlay	560
27.194.4.31	SetRows	560
27.194.4.32	SetType	560
27.194.4.33	Update	560
27.195	dcm::ParseException Class Reference	560
27.195.1	Detailed Description	561
27.195.2	Constructor & Destructor Documentation	561
27.195.2.1	ParseException	561
27.195.2.2	~ParseException	561
27.195.3	Member Function Documentation	561
27.195.3.1	GetLastElement	561
27.195.3.2	operator=	561
27.195.3.3	SetLastElement	562
27.196	dcm::Parser Class Reference	562
27.196.1	Detailed Description	563
27.196.2	Member Typedef Documentation	563
27.196.2.1	EndElementHandler	563
27.196.2.2	StartElementHandler	563
27.196.3	Member Enumeration Documentation	563
27.196.3.1	ErrorType	563
27.196.4	Constructor & Destructor Documentation	563
27.196.4.1	Parser	563
27.196.4.2	~Parser	563
27.196.5	Member Function Documentation	563
27.196.5.1	GetBuffer	563
27.196.5.2	GetCurrentByteIndex	563
27.196.5.3	GetErrorCode	563
27.196.5.4	GetErrorString	563
27.196.5.5	GetUserData	563
27.196.5.6	Parse	564
27.196.5.7	ParseBuffer	564
27.196.5.8	Process	564
27.196.5.9	SetElementHandler	564
27.196.5.10	SetUserData	564
27.197	dcm::Patient Class Reference	564

27.197.1	Detailed Description	564
27.197.2	Constructor & Destructor Documentation	564
27.197.2.1	IPatient	564
27.198	gdcm::network::PDataTFPDU Class Reference	564
27.198.1	Detailed Description	566
27.198.2	Member Typedef Documentation	566
27.198.2.1	ISizeType	566
27.198.3	Constructor & Destructor Documentation	566
27.198.3.1	IPDataTFPDU	566
27.198.4	Member Function Documentation	566
27.198.4.1	AddPresentationDataValue	566
27.198.4.2	GetNumberOfPresentationDataValues	566
27.198.4.3	GetPresentationDataValue	566
27.198.4.4	IsLastFragment	566
27.198.4.5	Print	566
27.198.4.6	Read	566
27.198.4.7	ReadInto	566
27.198.4.8	Size	566
27.198.4.9	Write	566
27.199	gdcm::PDBElement Class Reference	567
27.199.1	Detailed Description	568
27.199.2	Constructor & Destructor Documentation	568
27.199.2.1	IPDBElement	568
27.199.3	Member Function Documentation	568
27.199.3.1	GetName	568
27.199.3.2	GetValue	568
27.199.3.3	operator==	568
27.199.3.4	SetName	568
27.199.3.5	SetValue	568
27.199.4	Friends And Related Function Documentation	568
27.199.4.1	operator<<	568
27.199.5	Member Data Documentation	568
27.199.5.1	NameField	568
27.199.5.2	ValueField	568
27.200	gdcm::PDBHeader Class Reference	569
27.200.1	Detailed Description	569
27.200.2	Constructor & Destructor Documentation	570



27.200.2.1PDBHeader . . . . .	570
27.200.2.2~PDBHeader . . . . .	570
27.200.3Member Function Documentation . . . . .	570
27.200.3.1FindPDBElementByName . . . . .	570
27.200.3.2GetPDBEEnd . . . . .	570
27.200.3.3GetPDBElementByName . . . . .	570
27.200.3.4GetPDBInfoTag . . . . .	570
27.200.3.5LoadFromDataElement . . . . .	570
27.200.3.6Print . . . . .	570
27.200.4Friends And Related Function Documentation . . . . .	570
27.200.4.1operator<< . . . . .	570
27.201dcm::PDFCodec Class Reference . . . . .	571
27.201.1Detailed Description . . . . .	572
27.201.2Constructor & Destructor Documentation . . . . .	572
27.201.2.1PDFCodec . . . . .	572
27.201.2.2~PDFCodec . . . . .	572
27.201.3Member Function Documentation . . . . .	572
27.201.3.1CanCode . . . . .	572
27.201.3.2CanDecode . . . . .	572
27.201.3.3Decode . . . . .	572
27.202dcm::network::PDUFactory Class Reference . . . . .	572
27.202.1Detailed Description . . . . .	573
27.202.2Member Function Documentation . . . . .	573
27.202.2.1ConstructAbortPDU . . . . .	573
27.202.2.2ConstructPDU . . . . .	573
27.202.2.3ConstructReleasePDU . . . . .	573
27.202.2.4CreateCEchoPDU . . . . .	573
27.202.2.5CreateCFindPDU . . . . .	573
27.202.2.6CreateCMovePDU . . . . .	573
27.202.2.7CreateCStoreRQPDU . . . . .	573
27.202.2.8CreateCStoreRSPPDU . . . . .	573
27.202.2.9DetermineEventByPDU . . . . .	573
27.202.2.10GetPDVs . . . . .	573
27.203dcm::PersonName Class Reference . . . . .	574
27.203.1Detailed Description . . . . .	574
27.203.2Member Function Documentation . . . . .	574
27.203.2.1GetMaxLength . . . . .	574

27.203.2.2	GetNumberOfComponents	. . . . .	574
27.203.2.3	Print	. . . . .	574
27.203.2.4	SetBlob	. . . . .	574
27.203.2.5	SetComponents	. . . . .	574
27.203.2.6	SetComponents	. . . . .	574
27.203.3	Member Data Documentation	. . . . .	574
27.203.3.1	Component	. . . . .	575
27.203.3.2	MaxLength	. . . . .	575
27.203.3.3	MaxNumberOfComponents	. . . . .	575
27.203.3.4	Padding	. . . . .	575
27.203.3.5	Separator	. . . . .	575
27.204	dcm::PGXCodec Class Reference	. . . . .	575
27.204.1	Detailed Description	. . . . .	576
27.204.2	Constructor & Destructor Documentation	. . . . .	576
27.204.2.1	PGXCodec	. . . . .	576
27.204.2.2	~PGXCodec	. . . . .	576
27.204.3	Member Function Documentation	. . . . .	576
27.204.3.1	CanCode	. . . . .	576
27.204.3.2	CanDecode	. . . . .	577
27.204.3.3	Clone	. . . . .	577
27.204.3.4	GetHeaderInfo	. . . . .	577
27.204.3.5	Read	. . . . .	577
27.204.3.6	Write	. . . . .	577
27.205	dcm::PhotometricInterpretation Class Reference	. . . . .	577
27.205.1	Detailed Description	. . . . .	578
27.205.2	Member Enumeration Documentation	. . . . .	578
27.205.2.1	PIType	. . . . .	578
27.205.3	Constructor & Destructor Documentation	. . . . .	579
27.205.3.1	PhotometricInterpretation	. . . . .	579
27.205.4	Member Function Documentation	. . . . .	579
27.205.4.1	GetPIString	. . . . .	579
27.205.4.2	GetPIType	. . . . .	579
27.205.4.3	GetSamplesPerPixel	. . . . .	579
27.205.4.4	GetString	. . . . .	579
27.205.4.5	GetType	. . . . .	579
27.205.4.6	IsLossless	. . . . .	579
27.205.4.7	IsLossy	. . . . .	579

27.205.4.8	IsRetired	579
27.205.4.9	IsSameColorSpace	579
27.205.4.10	operator PType	579
27.205.5	Friends And Related Function Documentation	579
27.205.5.1	operator<<	579
27.206	dcm::PixelFormat Class Reference	579
27.206.1	Detailed Description	581
27.206.2	Member Enumeration Documentation	581
27.206.2.1	ScalarType	581
27.206.3	Constructor & Destructor Documentation	582
27.206.3.1	PixelFormat	582
27.206.3.2	PixelFormat	582
27.206.4	Member Function Documentation	582
27.206.4.1	GetBitsAllocated	582
27.206.4.2	GetBitsStored	582
27.206.4.3	GetHighBit	582
27.206.4.4	GetMax	582
27.206.4.5	GetMin	582
27.206.4.6	GetPixelRepresentation	582
27.206.4.7	GetPixelSize	583
27.206.4.8	GetSamplesPerPixel	583
27.206.4.9	GetScalarType	583
27.206.4.10	GetScalarTypeAsString	583
27.206.4.11	IsValid	583
27.206.4.12	operator ScalarType	583
27.206.4.13	operator"!="	583
27.206.4.14	operator"!="	583
27.206.4.15	operator==	583
27.206.4.16	operator==	583
27.206.4.17	Print	583
27.206.4.18	SetBitsAllocated	584
27.206.4.19	SetBitsStored	584
27.206.4.20	SetHighBit	584
27.206.4.21	SetPixelRepresentation	584
27.206.4.22	SetSamplesPerPixel	584
27.206.4.23	SetScalarType	584
27.206.4.24	Validate	584

27.206.5 Friends And Related Function Documentation . . . . .	584
27.206.5.1 Bitmap . . . . .	584
27.206.5.2 operator<< . . . . .	584
27.207 dcm::Pixmap Class Reference . . . . .	584
27.207.1 Detailed Description . . . . .	586
27.207.2 Constructor & Destructor Documentation . . . . .	586
27.207.2.1 Pixmap . . . . .	586
27.207.2.2 ~Pixmap . . . . .	586
27.207.3 Member Function Documentation . . . . .	586
27.207.3.1 AreOverlaysInPixelData . . . . .	586
27.207.3.2 GetCurve . . . . .	587
27.207.3.3 GetCurve . . . . .	587
27.207.3.4 GetIconImage . . . . .	587
27.207.3.5 GetIconImage . . . . .	587
27.207.3.6 GetNumberOfCurves . . . . .	587
27.207.3.7 GetNumberOfOverlays . . . . .	587
27.207.3.8 GetOverlay . . . . .	587
27.207.3.9 GetOverlay . . . . .	587
27.207.3.10 Print . . . . .	587
27.207.3.11 RemoveOverlay . . . . .	587
27.207.3.12 SetIconImage . . . . .	587
27.207.3.13 SetNumberOfCurves . . . . .	587
27.207.3.14 SetNumberOfOverlays . . . . .	587
27.207.4 Member Data Documentation . . . . .	587
27.207.4.1 Curves . . . . .	587
27.207.4.2 Icon . . . . .	587
27.207.4.3 Overlays . . . . .	587
27.208 dcm::PixmapReader Class Reference . . . . .	587
27.208.1 Detailed Description . . . . .	590
27.208.2 Constructor & Destructor Documentation . . . . .	590
27.208.2.1 PixmapReader . . . . .	590
27.208.2.2 ~PixmapReader . . . . .	590
27.208.3 Member Function Documentation . . . . .	590
27.208.3.1 GetPixmap . . . . .	590
27.208.3.2 GetPixmap . . . . .	590
27.208.3.3 Read . . . . .	590
27.208.3.4 ReadACRNEMAImage . . . . .	590

27.208.3.5	ReadImage	590
27.208.3.6	ReadImageInternal	591
27.208.4	Member Data Documentation	591
27.208.4.1	PixelData	591
27.209	dcm::PixmapToPixmapFilter Class Reference	591
27.209.1	Detailed Description	592
27.209.2	Constructor & Destructor Documentation	592
27.209.2.1	PixmapToPixmapFilter	592
27.209.2.2	~PixmapToPixmapFilter	593
27.209.3	Member Function Documentation	593
27.209.3.1	GetInput	593
27.209.3.2	GetOutput	593
27.209.3.3	GetOutputAsPixmap	593
27.210	dcm::PixmapWriter Class Reference	593
27.210.1	Detailed Description	595
27.210.2	Constructor & Destructor Documentation	595
27.210.2.1	PixmapWriter	595
27.210.2.2	~PixmapWriter	595
27.210.3	Member Function Documentation	595
27.210.3.1	DoIconImage	595
27.210.3.2	GetImage	595
27.210.3.3	GetImage	595
27.210.3.4	GetPixmap	596
27.210.3.5	GetPixmap	596
27.210.3.6	PrepareWrite	596
27.210.3.7	SetImage	596
27.210.3.8	SetPixmap	596
27.210.3.9	Write	596
27.210.4	Member Data Documentation	596
27.210.4.1	PixelData	596
27.211	dcm::PNMCodec Class Reference	596
27.211.1	Detailed Description	598
27.211.2	Constructor & Destructor Documentation	598
27.211.2.1	PNMCodec	598
27.211.2.2	~PNMCodec	598
27.211.3	Member Function Documentation	598
27.211.3.1	CanCode	598

27.211.3.2CanDecode . . . . .	598
27.211.3.3Clone . . . . .	598
27.211.3.4GetBufferLength . . . . .	598
27.211.3.5GetHeaderInfo . . . . .	598
27.211.3.6Read . . . . .	599
27.211.3.7SetBufferLength . . . . .	599
27.211.3.8Write . . . . .	599
27.211@dcm::Preamble Class Reference . . . . .	599
27.212.1Detailed Description . . . . .	599
27.212.2Constructor & Destructor Documentation . . . . .	600
27.212.2.1Preamble . . . . .	600
27.212.2.2~Preamble . . . . .	600
27.212.2.3Preamble . . . . .	600
27.212.3Member Function Documentation . . . . .	600
27.212.3.1Clear . . . . .	600
27.212.3.2Create . . . . .	600
27.212.3.3GetInternal . . . . .	600
27.212.3.4GetLength . . . . .	600
27.212.3.5IsEmpty . . . . .	600
27.212.3.6IsValid . . . . .	600
27.212.3.7operator= . . . . .	600
27.212.3.8Print . . . . .	600
27.212.3.9Read . . . . .	600
27.212.3.10Remove . . . . .	600
27.212.3.11Valid . . . . .	600
27.212.3.12Write . . . . .	600
27.212.4Friends And Related Function Documentation . . . . .	600
27.212.4.1operator<< . . . . .	600
27.211@dcm::PresentationContext Class Reference . . . . .	600
27.213.1Detailed Description . . . . .	601
27.213.2Member Typedef Documentation . . . . .	601
27.213.2.1SizeType . . . . .	601
27.213.2.2TransferSyntaxArrayType . . . . .	601
27.213.3Constructor & Destructor Documentation . . . . .	601
27.213.3.1PresentationContext . . . . .	601
27.213.3.2PresentationContext . . . . .	601
27.213.4Member Function Documentation . . . . .	601

27.213.4.1AddTransferSyntax . . . . .	601
27.213.4.2GetAbstractSyntax . . . . .	602
27.213.4.3GetNumberOfTransferSyntaxes . . . . .	602
27.213.4.4GetPresentationContextID . . . . .	602
27.213.4.5GetTransferSyntax . . . . .	602
27.213.4.6operator== . . . . .	602
27.213.4.7Print . . . . .	602
27.213.4.8SetAbstractSyntax . . . . .	602
27.213.4.9SetPresentationContextID . . . . .	602
27.214dcm::network::PresentationContextAC Class Reference . . . . .	602
27.214.1Detailed Description . . . . .	602
27.214.2Constructor & Destructor Documentation . . . . .	603
27.214.2.1PresentationContextAC . . . . .	603
27.214.3Member Function Documentation . . . . .	603
27.214.3.1GetPresentationContextID . . . . .	603
27.214.3.2GetReason . . . . .	603
27.214.3.3GetTransferSyntax . . . . .	603
27.214.3.4Print . . . . .	603
27.214.3.5Read . . . . .	603
27.214.3.6SetPresentationContextID . . . . .	603
27.214.3.7SetReason . . . . .	603
27.214.3.8SetTransferSyntax . . . . .	603
27.214.3.9Size . . . . .	603
27.214.3.10Write . . . . .	603
27.215dcm::PresentationContextGenerator Class Reference . . . . .	603
27.215.1Detailed Description . . . . .	604
27.215.2Member Typedef Documentation . . . . .	604
27.215.2.1PresentationContextArrayType . . . . .	604
27.215.2.2SizeType . . . . .	604
27.215.3Constructor & Destructor Documentation . . . . .	604
27.215.3.1PresentationContextGenerator . . . . .	604
27.215.4Member Function Documentation . . . . .	604
27.215.4.1AddPresentationContext . . . . .	605
27.215.4.2GenerateFromFilenames . . . . .	605
27.215.4.3GenerateFromUID . . . . .	605
27.215.4.4GetDefaultTransferSyntax . . . . .	605
27.215.4.5GetPresentationContexts . . . . .	605

27.215.4.6SetDefaultTransferSyntax . . . . .	605
27.215.4.7SetMergeModeToAbstractSyntax . . . . .	605
27.215.4.8SetMergeModeToTransferSyntax . . . . .	605
27.216.0dcm::network::PresentationContextRQ Class Reference . . . . .	605
27.216.1Detailed Description . . . . .	606
27.216.2Member Typedef Documentation . . . . .	606
27.216.2.1SizeType . . . . .	606
27.216.3Constructor & Destructor Documentation . . . . .	606
27.216.3.1PresentationContextRQ . . . . .	606
27.216.3.2PresentationContextRQ . . . . .	606
27.216.3.3PresentationContextRQ . . . . .	606
27.216.4Member Function Documentation . . . . .	606
27.216.4.1AddTransferSyntax . . . . .	606
27.216.4.2GetAbstractSyntax . . . . .	607
27.216.4.3GetAbstractSyntax . . . . .	607
27.216.4.4GetNumberOfTransferSyntaxes . . . . .	607
27.216.4.5GetPresentationContextID . . . . .	607
27.216.4.6GetTransferSyntax . . . . .	607
27.216.4.7GetTransferSyntax . . . . .	607
27.216.4.8GetTransferSyntaxes . . . . .	607
27.216.4.9operator== . . . . .	607
27.216.4.10Print . . . . .	607
27.216.4.11Read . . . . .	607
27.216.4.12SetAbstractSyntax . . . . .	607
27.216.4.13SetPresentationContextID . . . . .	607
27.216.4.14Size . . . . .	607
27.216.4.15Write . . . . .	607
27.217.0dcm::network::PresentationDataValue Class Reference . . . . .	607
27.217.1Detailed Description . . . . .	608
27.217.2Constructor & Destructor Documentation . . . . .	608
27.217.2.1PresentationDataValue . . . . .	608
27.217.3Member Function Documentation . . . . .	608
27.217.3.1ConcatenatePDVBlobs . . . . .	608
27.217.3.2ConcatenatePDVBlobsAsExplicit . . . . .	608
27.217.3.3GetBlob . . . . .	608
27.217.3.4GetIsCommand . . . . .	608
27.217.3.5GetIsLastFragment . . . . .	608



27.217.3.6	GetMessageHeader	608
27.217.3.7	GetPresentationContextID	608
27.217.3.8	Print	608
27.217.3.9	Read	608
27.217.3.10	ReadInto	609
27.217.3.11	SetBlob	609
27.217.3.12	SetCommand	609
27.217.3.13	DataSet	609
27.217.3.14	SetLastFragment	609
27.217.3.15	SetMessageHeader	609
27.217.3.16	SetPresentationContextID	609
27.217.3.17	Size	609
27.217.3.18	Write	609
27.218	gdcmm::Printer Class Reference	609
27.218.1	Detailed Description	611
27.218.2	Member Enumeration Documentation	611
27.218.2.1	PrintStyles	611
27.218.3	Constructor & Destructor Documentation	611
27.218.3.1	Printer	611
27.218.3.2	~Printer	611
27.218.4	Member Function Documentation	611
27.218.4.1	GetPrintStyle	611
27.218.4.2	Print	611
27.218.4.3	PrintDataElement	611
27.218.4.4	PrintDataSet	611
27.218.4.5	PrintSQ	612
27.218.4.6	SetColor	612
27.218.4.7	SetFile	612
27.218.4.8	SetStyle	612
27.218.5	Member Data Documentation	612
27.218.5.1	F	612
27.218.5.2	MaxPrintLength	612
27.218.5.3	PrintStyle	612
27.219	gdcmm::PrivateDict Class Reference	612
27.219.1	Detailed Description	613
27.219.2	Constructor & Destructor Documentation	613
27.219.2.1	PrivateDict	613

27.219.2.2~PrivateDict . . . . .	613
27.219.3Member Function Documentation . . . . .	613
27.219.3.1AddDictEntry . . . . .	613
27.219.3.2FindDictEntry . . . . .	613
27.219.3.3GetDictEntry . . . . .	613
27.219.3.4IsEmpty . . . . .	613
27.219.3.5LoadDefault . . . . .	613
27.219.3.6PrintXML . . . . .	613
27.219.3.7RemoveDictEntry . . . . .	613
27.219.4Friends And Related Function Documentation . . . . .	613
27.219.4.1Dicts . . . . .	613
27.219.4.2operator<< . . . . .	613
27.220dcm::PrivateTag Class Reference . . . . .	614
27.220.1Detailed Description . . . . .	615
27.220.2Constructor & Destructor Documentation . . . . .	615
27.220.2.1PrivateTag . . . . .	615
27.220.2.2PrivateTag . . . . .	615
27.220.3Member Function Documentation . . . . .	615
27.220.3.1GetAsDataElement . . . . .	615
27.220.3.2GetOwner . . . . .	615
27.220.3.3operator< . . . . .	615
27.220.3.4ReadFromCommaSeparatedString . . . . .	615
27.220.3.5SetOwner . . . . .	615
27.220.4Friends And Related Function Documentation . . . . .	615
27.220.4.1operator<< . . . . .	615
27.221dcm::ProgressEvent Class Reference . . . . .	616
27.221.1Detailed Description . . . . .	617
27.221.2Member Typedef Documentation . . . . .	617
27.221.2.1Self . . . . .	617
27.221.2.2Superclass . . . . .	617
27.221.3Constructor & Destructor Documentation . . . . .	617
27.221.3.1ProgressEvent . . . . .	617
27.221.3.2~ProgressEvent . . . . .	617
27.221.3.3ProgressEvent . . . . .	617
27.221.4Member Function Documentation . . . . .	617
27.221.4.1CheckEvent . . . . .	617
27.221.4.2GetEventName . . . . .	617

27.221.4.3	GetProgress	617
27.221.4.4	MakeObject	617
27.221.4.5	SetProgress	618
27.222	dcm::PVRGCodec Class Reference	618
27.222.1	Detailed Description	619
27.222.2	Constructor & Destructor Documentation	619
27.222.2.1	PVRGCodec	619
27.222.2.2	~PVRGCodec	619
27.222.3	Member Function Documentation	619
27.222.3.1	CanCode	619
27.222.3.2	CanDecode	619
27.222.3.3	Clone	620
27.222.3.4	Code	620
27.222.3.5	Decode	620
27.222.3.6	SetLossyFlag	620
27.223	dcm::PythonFilter Class Reference	620
27.223.1	Detailed Description	620
27.223.2	Constructor & Destructor Documentation	621
27.223.2.1	PythonFilter	621
27.223.2.2	~PythonFilter	621
27.223.3	Member Function Documentation	621
27.223.3.1	GetFile	621
27.223.3.2	GetFile	621
27.223.3.3	SetDicts	621
27.223.3.4	SetFile	621
27.223.3.5	ToPyObject	621
27.223.3.6	UseDictAlways	621
27.224	dcm::QueryBase Class Reference	621
27.224.1	Detailed Description	622
27.224.2	Constructor & Destructor Documentation	622
27.224.2.1	~QueryBase	622
27.224.3	Member Function Documentation	622
27.224.3.1	GetAllRequiredTags	622
27.224.3.2	GetAllTags	622
27.224.3.3	GetHierarchicalSearchTags	622
27.224.3.4	GetName	623
27.224.3.5	GetOptionalTags	623

27.224.3.6GetQueryLevel . . . . .	623
27.224.3.7GetRequiredTags . . . . .	623
27.224.3.8GetUniqueTags . . . . .	623
27.225dcm::QueryFactory Class Reference . . . . .	623
27.225.1Detailed Description . . . . .	623
27.225.2Member Function Documentation . . . . .	624
27.225.2.1GetCharacterFromCurrentLocale . . . . .	624
27.225.2.2ListCharSets . . . . .	624
27.225.2.3ProduceCharacterSetDataElement . . . . .	624
27.225.2.4ProduceQuery . . . . .	624
27.226dcm::QueryImage Class Reference . . . . .	624
27.226.1Detailed Description . . . . .	625
27.226.2Member Function Documentation . . . . .	626
27.226.2.1GetHierachicalSearchTags . . . . .	626
27.226.2.2GetName . . . . .	626
27.226.2.3GetOptionalTags . . . . .	626
27.226.2.4GetQueryLevel . . . . .	626
27.226.2.5GetRequiredTags . . . . .	626
27.226.2.6GetUniqueTags . . . . .	626
27.227dcm::QueryPatient Class Reference . . . . .	626
27.227.1Detailed Description . . . . .	627
27.227.2Member Function Documentation . . . . .	627
27.227.2.1GetHierachicalSearchTags . . . . .	628
27.227.2.2GetName . . . . .	628
27.227.2.3GetOptionalTags . . . . .	628
27.227.2.4GetQueryLevel . . . . .	628
27.227.2.5GetRequiredTags . . . . .	628
27.227.2.6GetUniqueTags . . . . .	628
27.228dcm::QuerySeries Class Reference . . . . .	628
27.228.1Detailed Description . . . . .	629
27.228.2Member Function Documentation . . . . .	630
27.228.2.1GetHierachicalSearchTags . . . . .	630
27.228.2.2GetName . . . . .	630
27.228.2.3GetOptionalTags . . . . .	630
27.228.2.4GetQueryLevel . . . . .	630
27.228.2.5GetRequiredTags . . . . .	630
27.228.2.6GetUniqueTags . . . . .	630

27.229dcm::QueryStudy Class Reference . . . . .	630
27.229.1Detailed Description . . . . .	631
27.229.2Member Function Documentation . . . . .	632
27.229.2.1GetHierarchicalSearchTags . . . . .	632
27.229.2.2GetName . . . . .	632
27.229.2.3GetOptionalTags . . . . .	632
27.229.2.4GetQueryLevel . . . . .	632
27.229.2.5GetRequiredTags . . . . .	632
27.229.2.6GetUniqueTags . . . . .	632
27.230dcm::RAWCodec Class Reference . . . . .	632
27.230.1Detailed Description . . . . .	634
27.230.2Constructor & Destructor Documentation . . . . .	634
27.230.2.1RAWCodec . . . . .	634
27.230.2.2~RAWCodec . . . . .	634
27.230.3Member Function Documentation . . . . .	634
27.230.3.1CanCode . . . . .	634
27.230.3.2CanDecode . . . . .	634
27.230.3.3Clone . . . . .	634
27.230.3.4Code . . . . .	634
27.230.3.5Decode . . . . .	634
27.230.3.6DecodeByStreams . . . . .	635
27.230.3.7DecodeBytes . . . . .	635
27.230.3.8GetHeaderInfo . . . . .	635
27.231dcm::Reader Class Reference . . . . .	635
27.231.1Detailed Description . . . . .	637
27.231.2Constructor & Destructor Documentation . . . . .	638
27.231.2.1Reader . . . . .	638
27.231.2.2~Reader . . . . .	638
27.231.3Member Function Documentation . . . . .	638
27.231.3.1CanRead . . . . .	638
27.231.3.2GetFile . . . . .	638
27.231.3.3GetFile . . . . .	638
27.231.3.4GetStreamPtr . . . . .	638
27.231.3.5Read . . . . .	638
27.231.3.6ReadDataSet . . . . .	639
27.231.3.7ReadMetaInformation . . . . .	639
27.231.3.8ReadPreamble . . . . .	639

27.231.3.9ReadSelectedPrivateTags . . . . .	639
27.231.3.10ReadSelectedTags . . . . .	639
27.231.3.11ReadUpToTag . . . . .	639
27.231.3.12SetFile . . . . .	639
27.231.3.13SetFileName . . . . .	639
27.231.3.14SetStream . . . . .	640
27.231.4Friends And Related Function Documentation . . . . .	640
27.231.4.1StreamImageReader . . . . .	640
27.231.5Member Data Documentation . . . . .	640
27.231.5.1F . . . . .	640
27.232dcm::Region Class Reference . . . . .	640
27.232.1Detailed Description . . . . .	641
27.232.2Constructor & Destructor Documentation . . . . .	641
27.232.2.1Region . . . . .	641
27.232.2.2~Region . . . . .	641
27.232.3Member Function Documentation . . . . .	641
27.232.3.1Area . . . . .	641
27.232.3.2Clone . . . . .	641
27.232.3.3ComputeBoundingBox . . . . .	641
27.232.3.4Empty . . . . .	641
27.232.3.5IsValid . . . . .	641
27.232.3.6Print . . . . .	642
27.233dcm::Rescaler Class Reference . . . . .	642
27.233.1Detailed Description . . . . .	643
27.233.2Constructor & Destructor Documentation . . . . .	643
27.233.2.1Rescaler . . . . .	643
27.233.2.2~Rescaler . . . . .	643
27.233.3Member Function Documentation . . . . .	643
27.233.3.1ComputeInterceptSlopePixelType . . . . .	643
27.233.3.2ComputePixelTypeFromMinMax . . . . .	644
27.233.3.3GetIntercept . . . . .	644
27.233.3.4GetSlope . . . . .	644
27.233.3.5InverseRescale . . . . .	644
27.233.3.6InverseRescaleFunctionIntoBestFit . . . . .	644
27.233.3.7Rescale . . . . .	644
27.233.3.8RescaleFunctionIntoBestFit . . . . .	644
27.233.3.9SetIntercept . . . . .	644

27.233.3.1	<del>SetMinMaxForPixelType</del>	644
27.233.3.1	<del>SetPixelFormat</del>	644
27.233.3.1	<del>SetSlope</del>	644
27.233.3.1	<del>SetTargetPixelType</del>	644
27.233.3.1	<del>SetUseTargetPixelType</del>	644
27.234	dcm::RLECodec Class Reference	645
27.234.1	Detailed Description	646
27.234.2	Constructor & Destructor Documentation	646
27.234.2.1	RLECodec	646
27.234.2.2	~RLECodec	646
27.234.3	Member Function Documentation	646
27.234.3.1	CanCode	646
27.234.3.2	CanDecode	647
27.234.3.3	Clone	647
27.234.3.4	Code	647
27.234.3.5	Decode	647
27.234.3.6	DecodeByStreams	647
27.234.3.7	DecodeExtent	647
27.234.3.8	GetBufferLength	647
27.234.3.9	GetHeaderInfo	647
27.234.3.10	SetBufferLength	647
27.234.3.11	SetLength	647
27.234.4	Friends And Related Function Documentation	647
27.234.4.1	ImageRegionReader	647
27.235	dcm::network::RoleSelectionSub Class Reference	648
27.235.1	Detailed Description	648
27.235.2	Constructor & Destructor Documentation	648
27.235.2.1	RoleSelectionSub	648
27.235.3	Member Function Documentation	648
27.235.3.1	Print	648
27.235.3.2	Read	648
27.235.3.3	SetTuple	648
27.235.3.4	Size	648
27.235.3.5	Write	648
27.236	dcm::SerieHelper::Rule Struct Reference	648
27.236.1	Member Data Documentation	649
27.236.1.1	elem	649

27.236.1.2group . . . . .	649
27.236.1.3op . . . . .	649
27.236.1.4value . . . . .	649
27.237dcm::Scanner Class Reference . . . . .	649
27.237.1Detailed Description . . . . .	652
27.237.2Member Typedef Documentation . . . . .	652
27.237.2.1ConstIterator . . . . .	652
27.237.2.2MappingType . . . . .	652
27.237.2.3TagToValue . . . . .	652
27.237.2.4TagToValueValueType . . . . .	653
27.237.2.5ValuesType . . . . .	653
27.237.3Constructor & Destructor Documentation . . . . .	653
27.237.3.1Scanner . . . . .	653
27.237.3.2~Scanner . . . . .	653
27.237.4Member Function Documentation . . . . .	653
27.237.4.1AddPrivateTag . . . . .	653
27.237.4.2AddSkipTag . . . . .	653
27.237.4.3AddTag . . . . .	653
27.237.4.4Begin . . . . .	653
27.237.4.5ClearSkipTags . . . . .	653
27.237.4.6ClearTags . . . . .	653
27.237.4.7End . . . . .	653
27.237.4.8GetAllFileNamesFromTagToValue . . . . .	653
27.237.4.9GetFilenameFromTagToValue . . . . .	653
27.237.4.10GetFileNames . . . . .	653
27.237.4.11GetKeys . . . . .	653
27.237.4.12GetMapping . . . . .	654
27.237.4.13GetMappingFromTagToValue . . . . .	654
27.237.4.14GetMappings . . . . .	654
27.237.4.15GetOrderedValues . . . . .	654
27.237.4.16GetValue . . . . .	654
27.237.4.17GetValues . . . . .	654
27.237.4.18GetValues . . . . .	654
27.237.4.19Key . . . . .	655
27.237.4.20New . . . . .	655
27.237.4.21Print . . . . .	655
27.237.4.22ProcessPublicTag . . . . .	655



27.237.4.2Scan	655
27.237.5Friends And Related Function Documentation	655
27.237.5.1operator<<	655
27.238dcm::Segment Class Reference	655
27.238.1Detailed Description	657
27.238.2Member Typedef Documentation	658
27.238.2.1SurfaceVector	658
27.238.3Member Enumeration Documentation	658
27.238.3.1ALGOType	658
27.238.4Constructor & Destructor Documentation	658
27.238.4.1Segment	658
27.238.4.2~Segment	658
27.238.5Member Function Documentation	658
27.238.5.1AddSurface	658
27.238.5.2GetALGOType	658
27.238.5.3GetALGOTypeString	658
27.238.5.4GetAnatomicRegion	658
27.238.5.5GetAnatomicRegion	658
27.238.5.6GetPropertyCategory	658
27.238.5.7GetPropertyCategory	658
27.238.5.8GetPropertyType	658
27.238.5.9GetPropertyType	658
27.238.5.10GetSegmentAlgorithmName	658
27.238.5.10GetSegmentAlgorithmType	658
27.238.5.10GetSegmentDescription	658
27.238.5.10GetSegmentLabel	658
27.238.5.10GetSegmentNumber	658
27.238.5.10GetSurface	658
27.238.5.10GetSurfaceCount	659
27.238.5.10GetSurfaces	659
27.238.5.10GetSurfaces	659
27.238.5.10SetAnatomicRegion	659
27.238.5.20SetPropertyCategory	659
27.238.5.23SetPropertyType	659
27.238.5.28SetSegmentAlgorithmName	659
27.238.5.29SetSegmentAlgorithmType	659
27.238.5.29SetSegmentAlgorithmType	659

27.238.5.25	SetSegmentDescription	. . . . .	659
27.238.5.26	SetSegmentLabel	. . . . .	659
27.238.5.27	SetSegmentNumber	. . . . .	659
27.238.5.28	SetSurfaceCount	. . . . .	659
27.238.6	Member Data Documentation	. . . . .	659
27.238.6.1	AnatomicRegion	. . . . .	659
27.238.6.2	PropertyCategory	. . . . .	659
27.238.6.3	PropertyType	. . . . .	659
27.238.6.4	SegmentAlgorithmName	. . . . .	659
27.238.6.5	SegmentAlgorithmType	. . . . .	659
27.238.6.6	SegmentDescription	. . . . .	659
27.238.6.7	SegmentLabel	. . . . .	659
27.238.6.8	SegmentNumber	. . . . .	659
27.238.6.9	SurfaceCount	. . . . .	659
27.238.6.10	Surfaces	. . . . .	659
27.239	dcm::SegmentedPaletteColorLookupTable Class Reference	. . . . .	660
27.239.1	Detailed Description	. . . . .	661
27.239.2	Constructor & Destructor Documentation	. . . . .	661
27.239.2.1	SegmentedPaletteColorLookupTable	. . . . .	661
27.239.2.2	~SegmentedPaletteColorLookupTable	. . . . .	661
27.239.3	Member Function Documentation	. . . . .	661
27.239.3.1	Print	. . . . .	661
27.239.3.2	SetLUT	. . . . .	661
27.240	dcm::SegmentReader Class Reference	. . . . .	661
27.240.1	Detailed Description	. . . . .	663
27.240.2	Member Typedef Documentation	. . . . .	663
27.240.2.1	SegmentMap	. . . . .	663
27.240.2.2	SegmentVector	. . . . .	663
27.240.3	Constructor & Destructor Documentation	. . . . .	663
27.240.3.1	SegmentReader	. . . . .	663
27.240.3.2	~SegmentReader	. . . . .	663
27.240.4	Member Function Documentation	. . . . .	664
27.240.4.1	GetSegments	. . . . .	664
27.240.4.2	GetSegments	. . . . .	664
27.240.4.3	Read	. . . . .	664
27.240.4.4	ReadSegment	. . . . .	664
27.240.4.5	ReadSegments	. . . . .	664

27.240.5	Member Data Documentation . . . . .	664
27.240.5.1	Segments . . . . .	664
27.241.0	gdcmm::SegmentWriter Class Reference . . . . .	664
27.241.1	Detailed Description . . . . .	665
27.241.2	Member Typedef Documentation . . . . .	666
27.241.2.1	SegmentVector . . . . .	666
27.241.3	Constructor & Destructor Documentation . . . . .	666
27.241.3.1	SegmentWriter . . . . .	666
27.241.3.2	~SegmentWriter . . . . .	666
27.241.4	Member Function Documentation . . . . .	666
27.241.4.1	AddSegment . . . . .	666
27.241.4.2	GetNumberOfSegments . . . . .	666
27.241.4.3	GetSegment . . . . .	666
27.241.4.4	GetSegments . . . . .	666
27.241.4.5	GetSegments . . . . .	666
27.241.4.6	PrepareWrite . . . . .	666
27.241.4.7	SetNumberOfSegments . . . . .	666
27.241.4.8	SetSegments . . . . .	666
27.241.4.9	Write . . . . .	666
27.241.5	Member Data Documentation . . . . .	666
27.241.5.1	Segments . . . . .	666
27.242.0	gdcmm::SequenceOfFragments Class Reference . . . . .	666
27.242.1	Detailed Description . . . . .	668
27.242.2	Member Typedef Documentation . . . . .	669
27.242.2.1	ConstIterator . . . . .	669
27.242.2.2	FragmentVector . . . . .	669
27.242.2.3	Iterator . . . . .	669
27.242.2.4	SizeType . . . . .	669
27.242.3	Constructor & Destructor Documentation . . . . .	669
27.242.3.1	SequenceOfFragments . . . . .	669
27.242.4	Member Function Documentation . . . . .	669
27.242.4.1	AddFragment . . . . .	669
27.242.4.2	Begin . . . . .	669
27.242.4.3	Begin . . . . .	669
27.242.4.4	Clear . . . . .	669
27.242.4.5	ComputeByteLength . . . . .	669
27.242.4.6	ComputeLength . . . . .	669

27.242.4.7End	. 669
27.242.4.8End	. 669
27.242.4.9GetBuffer	. 669
27.242.4.10GetFragBuffer	. 669
27.242.4.10GetFragment	. 670
27.242.4.10GetLength	. 670
27.242.4.10GetNumberOfFragments	. 670
27.242.4.10GetTable	. 670
27.242.4.10GetTable	. 670
27.242.4.10New	. 670
27.242.4.10operator==	. 670
27.242.4.10Print	. 670
27.242.4.10Read	. 670
27.242.4.20ReadPreValue	. 670
27.242.4.20ReadValue	. 670
27.242.4.20SetLength	. 671
27.242.4.20Write	. 671
27.242.4.20WriteBuffer	. 671
27.243dcm::SequenceOfItems Class Reference	. 671
27.243.1Detailed Description	. 673
27.243.2Member Typedef Documentation	. 674
27.243.2.1ConstIterator	. 674
27.243.2.2ItemVector	. 674
27.243.2.3Iterator	. 674
27.243.2.4SizeType	. 674
27.243.3Constructor & Destructor Documentation	. 674
27.243.3.1SequenceOfItems	. 674
27.243.4Member Function Documentation	. 674
27.243.4.1AddItem	. 674
27.243.4.2Begin	. 674
27.243.4.3Begin	. 674
27.243.4.4Clear	. 674
27.243.4.5ComputeLength	. 674
27.243.4.6End	. 674
27.243.4.7End	. 675
27.243.4.8FindDataElement	. 675
27.243.4.9GetItem	. 675

27.243.4.10	GetItem	675
27.243.4.10	GetLength	675
27.243.4.10	GetNumberOfItems	675
27.243.4.13	UndefinedLength	675
27.243.4.14	New	675
27.243.4.15	operator=	675
27.243.4.16	operator==	675
27.243.4.17	Print	675
27.243.4.18	Read	676
27.243.4.19	SetLength	676
27.243.4.20	SetLengthToUndefined	676
27.243.4.23	SetNumberOfItems	676
27.243.4.24	Write	676
27.243.5	Member Data Documentation	676
27.243.5.1	Items	676
27.243.5.2	SequenceLengthField	676
27.244	dcm::SerieHelper Class Reference	677
27.244.1	Detailed Description	678
27.244.2	Member Typedef Documentation	678
27.244.2.1	SerieRestrictions	678
27.244.2.2	SingleSerieUIDFileSetmap	678
27.244.3	Constructor & Destructor Documentation	678
27.244.3.1	SerieHelper	678
27.244.3.2	~SerieHelper	678
27.244.4	Member Function Documentation	678
27.244.4.1	AddFile	678
27.244.4.2	AddFileName	679
27.244.4.3	AddRestriction	679
27.244.4.4	AddRestriction	679
27.244.4.5	AddRestriction	679
27.244.4.6	Clear	679
27.244.4.7	CreateDefaultUniqueSeriesIdentifier	679
27.244.4.8	CreateUniqueSeriesIdentifier	679
27.244.4.9	FileNameOrdering	679
27.244.4.10	GetFirstSingleSerieUIDFileSet	679
27.244.4.10	GetNextSingleSerieUIDFileSet	679
27.244.4.12	ImagePositionPatientOrdering	679

27.244.4.10	OrderFileList	679
27.244.4.11	SetDirectory	679
27.244.4.12	SetLoadMode	679
27.244.4.13	SetUseSeriesDetails	679
27.244.4.14	UserOrdering	679
27.244.5	Member Data Documentation	679
27.244.5.1	ItFileSetHt	679
27.244.5.2	SingleSerieUIDFileSetHT	679
27.245	dcm::Series Class Reference	679
27.245.1	Detailed Description	680
27.245.2	Constructor & Destructor Documentation	680
27.245.2.1	Series	680
27.246	dcm::network::ServiceClassApplicationInformation Class Reference	680
27.246.1	Detailed Description	680
27.246.2	Constructor & Destructor Documentation	680
27.246.2.1	ServiceClassApplicationInformation	680
27.246.3	Member Function Documentation	680
27.246.3.1	Print	680
27.246.3.2	Read	680
27.246.3.3	SetTuple	680
27.246.3.4	Size	680
27.246.3.5	Write	680
27.247	dcm::ServiceClassUser Class Reference	681
27.247.1	Detailed Description	683
27.247.2	Constructor & Destructor Documentation	683
27.247.2.1	ServiceClassUser	683
27.247.2.2	~ServiceClassUser	683
27.247.3	Member Function Documentation	683
27.247.3.1	GetAETitle	683
27.247.3.2	GetCalledAETitle	683
27.247.3.3	GetTimeout	683
27.247.3.4	InitializeConnection	683
27.247.3.5	IsPresentationContextAccepted	683
27.247.3.6	New	683
27.247.3.7	SendEcho	683
27.247.3.8	SendFind	684
27.247.3.9	SendMove	684

27.247.3.1SendMove	684
27.247.3.1SendMove	684
27.247.3.1SendStore	684
27.247.3.1SendStore	684
27.247.3.1SendStore	684
27.247.3.1SetAETitle	684
27.247.3.1SetCalledAETitle	684
27.247.3.1SetHostname	685
27.247.3.1SetPort	685
27.247.3.1SetPortSCP	685
27.247.3.2SetPresentationContexts	685
27.247.3.2SetTimeout	685
27.247.3.2StartAssociation	685
27.247.3.2StopAssociation	686
27.248dcm::SHA1 Class Reference	686
27.248.1Detailed Description	686
27.248.2Constructor & Destructor Documentation	686
27.248.2.1SHA1	686
27.248.2.2~SHA1	686
27.248.3Member Function Documentation	686
27.248.3.1Compute	687
27.248.3.2ComputeFile	687
27.249dcm::SimpleMemberCommand< T > Class Template Reference	687
27.249.1Detailed Description	689
27.249.2Member Typedef Documentation	689
27.249.2.1Self	689
27.249.2.2TMemberFunctionPointer	689
27.249.3Constructor & Destructor Documentation	689
27.249.3.1SimpleMemberCommand	689
27.249.3.2~SimpleMemberCommand	689
27.249.4Member Function Documentation	689
27.249.4.1Execute	689
27.249.4.2Execute	689
27.249.4.3New	690
27.249.4.4SetCallbackFunction	690
27.249.5Member Data Documentation	690
27.249.5.1m_MemberFunction	690

27.249.5.2m_This . . . . .	690
27.250dcm::SimpleSubjectWatcher Class Reference . . . . .	690
27.250.1Detailed Description . . . . .	691
27.250.2Constructor & Destructor Documentation . . . . .	691
27.250.2.1SimpleSubjectWatcher . . . . .	691
27.250.2.2~SimpleSubjectWatcher . . . . .	691
27.250.3Member Function Documentation . . . . .	691
27.250.3.1EndFilter . . . . .	691
27.250.3.2ShowAbort . . . . .	691
27.250.3.3ShowAnonymization . . . . .	691
27.250.3.4ShowData . . . . .	691
27.250.3.5ShowDataSet . . . . .	691
27.250.3.6ShowFileName . . . . .	691
27.250.3.7ShowIteration . . . . .	691
27.250.3.8ShowProgress . . . . .	691
27.250.3.9StartFilter . . . . .	692
27.250.3.10TestAbortOff . . . . .	692
27.250.3.11TestAbortOn . . . . .	692
27.251dcm::SmartPointer< ObjectType > Class Template Reference . . . . .	692
27.251.1Detailed Description . . . . .	694
27.251.2Constructor & Destructor Documentation . . . . .	694
27.251.2.1SmartPointer . . . . .	694
27.251.2.2SmartPointer . . . . .	694
27.251.2.3SmartPointer . . . . .	694
27.251.2.4SmartPointer . . . . .	694
27.251.2.5~SmartPointer . . . . .	694
27.251.3Member Function Documentation . . . . .	694
27.251.3.1GetPointer . . . . .	695
27.251.3.2operator ObjectType * . . . . .	695
27.251.3.3operator* . . . . .	695
27.251.3.4operator-> . . . . .	695
27.251.3.5operator= . . . . .	695
27.251.3.6operator= . . . . .	695
27.251.3.7operator= . . . . .	695
27.252dcm::network::SOPClassExtendedNegociationSub Class Reference . . . . .	695
27.252.1Detailed Description . . . . .	696
27.252.2Constructor & Destructor Documentation . . . . .	696



27.252.2.1SOPClassExtendedNegociationSub . . . . .	696
27.252.3Member Function Documentation . . . . .	696
27.252.3.1Print . . . . .	696
27.252.3.2Read . . . . .	696
27.252.3.3SetTuple . . . . .	696
27.252.3.4Size . . . . .	696
27.252.3.5Write . . . . .	696
27.253dcm::SOPClassUIDToIOD Class Reference . . . . .	696
27.253.1Detailed Description . . . . .	697
27.253.2Member Typedef Documentation . . . . .	697
27.253.2.1const . . . . .	697
27.253.3Member Function Documentation . . . . .	697
27.253.3.1GetIOD . . . . .	697
27.253.3.2GetIODFromSOPClassUID . . . . .	697
27.253.3.3GetNumberOfSOPClassToIOD . . . . .	697
27.253.3.4GetSOPClassUIDFromIOD . . . . .	697
27.253.3.5GetSOPClassUIDToIOD . . . . .	697
27.253.3.6GetSOPClassUIDToIODs . . . . .	697
27.254dcm::Sorter Class Reference . . . . .	697
27.254.1Detailed Description . . . . .	699
27.254.2Member Typedef Documentation . . . . .	699
27.254.2.1SelectionMap . . . . .	699
27.254.2.2SortFunction . . . . .	699
27.254.3Constructor & Destructor Documentation . . . . .	700
27.254.3.1Sorter . . . . .	700
27.254.3.2~Sorter . . . . .	700
27.254.4Member Function Documentation . . . . .	700
27.254.4.1AddSelect . . . . .	700
27.254.4.2GetFileNames . . . . .	700
27.254.4.3Print . . . . .	700
27.254.4.4SetSortFunction . . . . .	700
27.254.4.5Sort . . . . .	700
27.254.4.6StableSort . . . . .	700
27.254.5Friends And Related Function Documentation . . . . .	701
27.254.5.1operator<< . . . . .	701
27.254.6Member Data Documentation . . . . .	701
27.254.6.1FileNames . . . . .	701

27.254.6.2	Selection	. . . . .	701
27.254.6.3	SortFunc	. . . . .	701
27.255	dcm::Spacing Class Reference	. . . . .	701
27.255.1	Detailed Description	. . . . .	701
27.255.2	Member Enumeration Documentation	. . . . .	702
27.255.2.1	SpacingType	. . . . .	702
27.255.3	Constructor & Destructor Documentation	. . . . .	702
27.255.3.1	Spacing	. . . . .	702
27.255.3.2	~Spacing	. . . . .	702
27.255.4	Member Function Documentation	. . . . .	702
27.255.4.1	ComputePixelAspectRatioFromPixelSpacing	. . . . .	702
27.256	dcm::Spectroscopy Class Reference	. . . . .	703
27.256.1	Detailed Description	. . . . .	703
27.256.2	Constructor & Destructor Documentation	. . . . .	703
27.256.2.1	Spectroscopy	. . . . .	703
27.257	dcm::SplitMosaicFilter Class Reference	. . . . .	703
27.257.1	Detailed Description	. . . . .	704
27.257.2	Constructor & Destructor Documentation	. . . . .	704
27.257.2.1	SplitMosaicFilter	. . . . .	704
27.257.2.2	~SplitMosaicFilter	. . . . .	704
27.257.3	Member Function Documentation	. . . . .	704
27.257.3.1	ComputeMOSAICDimensions	. . . . .	704
27.257.3.2	GetFile	. . . . .	704
27.257.3.3	GetFile	. . . . .	704
27.257.3.4	GetImage	. . . . .	704
27.257.3.5	GetImage	. . . . .	704
27.257.3.6	SetFile	. . . . .	704
27.257.3.7	SetImage	. . . . .	704
27.257.3.8	Split	. . . . .	704
27.258	dcm::StartEvent Class Reference	. . . . .	704
27.259	dcm::static_assert_test< x > Struct Template Reference	. . . . .	706
27.260	dcm::STATIC_ASSERTION_FAILURE< x > Struct Template Reference	. . . . .	706
27.261	dcm::STATIC_ASSERTION_FAILURE< true > Struct Template Reference	. . . . .	706
27.261.1	Member Enumeration Documentation	. . . . .	706
27.261.1.1	anonymous enum	. . . . .	706
27.262	dcm::StreamImageReader Class Reference	. . . . .	706
27.262.1	Detailed Description	. . . . .	707

27.262.2	Constructor & Destructor Documentation	707
27.262.2.1	StreamImageReader	707
27.262.2.2	~StreamImageReader	707
27.262.3	Member Function Documentation	707
27.262.3.1	CanReadImage	707
27.262.3.2	DefinePixelExtent	707
27.262.3.3	DefineProperBufferLength	708
27.262.3.4	GetDimensionsValueForResolution	708
27.262.3.5	GetFile	708
27.262.3.6	Read	708
27.262.3.7	ReadImageInformation	708
27.262.3.8	SetFileName	709
27.262.3.9	SetStream	709
27.263	dcm::StreamImageWriter Class Reference	709
27.263.1	Detailed Description	711
27.263.2	Constructor & Destructor Documentation	712
27.263.2.1	StreamImageWriter	712
27.263.2.2	~StreamImageWriter	712
27.263.3	Member Function Documentation	712
27.263.3.1	CanWriteFile	712
27.263.3.2	DefinePixelExtent	712
27.263.3.3	DefineProperBufferLength	712
27.263.3.4	SetFile	712
27.263.3.5	SetFileName	712
27.263.3.6	SetStream	713
27.263.3.7	Write	713
27.263.3.8	WriteImageInformation	713
27.263.3.9	WriteImageSubregionRAW	713
27.263.3.10	WriteRawHeader	713
27.263.4	Member Data Documentation	714
27.263.4.1	mElementOffsets	714
27.263.4.2	mElementOffsets1	714
27.263.4.3	mSPFile	714
27.263.4.4	mWriter	714
27.263.4.5	mXMax	714
27.263.4.6	mXMin	714
27.263.4.7	mYMax	714

27.263.4.8mYMin . . . . .	714
27.263.4.9mZMax . . . . .	714
27.263.4.10ZMin . . . . .	714
27.264dcm::String< TDelimiter, TMaxLength, TPadChar > Class Template Reference . . . . .	714
27.264.1Detailed Description . . . . .	716
27.264.2Member Typedef Documentation . . . . .	716
27.264.2.1const_iterator . . . . .	716
27.264.2.2const_reference . . . . .	717
27.264.2.3const_reverse_iterator . . . . .	717
27.264.2.4difference_type . . . . .	717
27.264.2.5iterator . . . . .	717
27.264.2.6pointer . . . . .	717
27.264.2.7reference . . . . .	717
27.264.2.8reverse_iterator . . . . .	717
27.264.2.9size_type . . . . .	717
27.264.2.10value_type . . . . .	717
27.264.3Constructor & Destructor Documentation . . . . .	717
27.264.3.1String . . . . .	717
27.264.3.2String . . . . .	717
27.264.3.3String . . . . .	717
27.264.3.4String . . . . .	717
27.264.4Member Function Documentation . . . . .	717
27.264.4.1IsValid . . . . .	717
27.264.4.2operator const char * . . . . .	718
27.264.4.3Trim . . . . .	718
27.264.4.4Trim . . . . .	718
27.264.4.5Truncate . . . . .	718
27.265dcm::StringFilter Class Reference . . . . .	718
27.265.1Detailed Description . . . . .	719
27.265.2Constructor & Destructor Documentation . . . . .	719
27.265.2.1StringFilter . . . . .	719
27.265.2.2~StringFilter . . . . .	719
27.265.3Member Function Documentation . . . . .	719
27.265.3.1ExecuteQuery . . . . .	719
27.265.3.2ExecuteQuery . . . . .	719
27.265.3.3FromString . . . . .	719
27.265.3.4FromString . . . . .	719

27.265.3.5	GetFile	. . . . .	719
27.265.3.6	GetFile	. . . . .	719
27.265.3.7	SetDicts	. . . . .	719
27.265.3.8	SetFile	. . . . .	719
27.265.3.9	ToString	. . . . .	720
27.265.3.10	ToStringPair	. . . . .	720
27.265.3.11	ToStringPair	. . . . .	720
27.265.3.12	UseDictAlways	. . . . .	720
27.266	gdcm::Study Class Reference	. . . . .	720
27.266.1	Detailed Description	. . . . .	720
27.266.2	Constructor & Destructor Documentation	. . . . .	721
27.266.2.1	Study	. . . . .	721
27.267	gdcm::Subject Class Reference	. . . . .	721
27.267.1	Detailed Description	. . . . .	722
27.267.2	Constructor & Destructor Documentation	. . . . .	722
27.267.2.1	Subject	. . . . .	722
27.267.2.2	~Subject	. . . . .	722
27.267.3	Member Function Documentation	. . . . .	722
27.267.3.1	AddObserver	. . . . .	722
27.267.3.2	AddObserver	. . . . .	722
27.267.3.3	GetCommand	. . . . .	722
27.267.3.4	HasObserver	. . . . .	723
27.267.3.5	InvokeEvent	. . . . .	723
27.267.3.6	InvokeEvent	. . . . .	723
27.267.3.7	RemoveAllObservers	. . . . .	723
27.267.3.8	RemoveObserver	. . . . .	723
27.268	gdcm::Surface Class Reference	. . . . .	723
27.268.1	Detailed Description	. . . . .	726
27.268.2	Member Enumeration Documentation	. . . . .	726
27.268.2.1	STATES	. . . . .	726
27.268.2.2	VIEWType	. . . . .	726
27.268.3	Constructor & Destructor Documentation	. . . . .	726
27.268.3.1	Surface	. . . . .	726
27.268.3.2	~Surface	. . . . .	726
27.268.4	Member Function Documentation	. . . . .	727
27.268.4.1	GetAlgorithmFamily	. . . . .	727
27.268.4.2	GetAlgorithmFamily	. . . . .	727

27.268.4.3GetAlgorithmName	. 727
27.268.4.4GetAlgorithmVersion	. 727
27.268.4.5GetAxisOfRotation	. 727
27.268.4.6GetCenterOfRotation	. 727
27.268.4.7GetFiniteVolume	. 727
27.268.4.8GetManifold	. 727
27.268.4.9GetMaximumPointDistance	. 727
27.268.4.10GetMeanPointDistance	. 727
27.268.4.10GetMeshPrimitive	. 727
27.268.4.10GetMeshPrimitive	. 727
27.268.4.10GetNumberOfSurfacePoints	. 727
27.268.4.10GetNumberOfVectors	. 727
27.268.4.10GetPointCoordinatesData	. 727
27.268.4.10GetPointCoordinatesData	. 727
27.268.4.10GetPointPositionAccuracy	. 727
27.268.4.10GetPointsBoundingBoxCoordinates	. 727
27.268.4.10GetProcessingAlgorithm	. 728
27.268.4.20GetProcessingAlgorithm	. 728
27.268.4.20GetRecommendedDisplayCIELabValue	. 728
27.268.4.20GetRecommendedDisplayCIELabValue	. 728
27.268.4.23GetRecommendedDisplayGrayscaleValue	. 728
27.268.4.24GetRecommendedPresentationOpacity	. 728
27.268.4.25GetRecommendedPresentationType	. 728
27.268.4.26GetSTATES	. 728
27.268.4.27GetSTATESString	. 728
27.268.4.28GetSurfaceComments	. 728
27.268.4.29GetSurfaceNumber	. 728
27.268.4.30GetSurfaceProcessing	. 728
27.268.4.30GetSurfaceProcessingDescription	. 728
27.268.4.30GetSurfaceProcessingRatio	. 728
27.268.4.33GetVectorAccuracy	. 728
27.268.4.33GetVectorCoordinateData	. 728
27.268.4.35GetVectorCoordinateData	. 728
27.268.4.36GetVectorDimensionality	. 728
27.268.4.37GetVIEWType	. 728
27.268.4.38GetVIEWTypeString	. 728
27.268.4.39GetAlgorithmFamily	. 728

27.268.4.49	SetAlgorithmName	728
27.268.4.49	SetAlgorithmVersion	728
27.268.4.49	SetAxisOfRotation	728
27.268.4.49	SetCenterOfRotation	729
27.268.4.49	SetFiniteVolume	729
27.268.4.49	SetManifold	729
27.268.4.49	SetMaximumPointDistance	729
27.268.4.49	SetMeanPointDistance	729
27.268.4.49	SetMeshPrimitive	729
27.268.4.49	SetNumberOfSurfacePoints	729
27.268.4.50	SetNumberOfVectors	729
27.268.4.53	SetPointCoordinatesData	729
27.268.4.53	SetPointPositionAccuracy	729
27.268.4.53	SetPointsBoundingBoxCoordinates	729
27.268.4.53	SetProcessingAlgorithm	729
27.268.4.55	SetRecommendedDisplayCIELabValue	729
27.268.4.55	SetRecommendedDisplayCIELabValue	729
27.268.4.57	SetRecommendedDisplayCIELabValue	729
27.268.4.58	SetRecommendedDisplayGrayscaleValue	729
27.268.4.59	SetRecommendedPresentationOpacity	729
27.268.4.60	SetRecommendedPresentationType	729
27.268.4.63	SetSurfaceComments	729
27.268.4.63	SetSurfaceNumber	729
27.268.4.63	SetSurfaceProcessing	729
27.268.4.63	SetSurfaceProcessingDescription	729
27.268.4.65	SetSurfaceProcessingRatio	729
27.268.4.66	SetVectorAccuracy	729
27.268.4.66	SetVectorCoordinateData	729
27.268.4.68	SetVectorDimensionality	730
27.269	gdcm::SurfaceHelper Class Reference	730
27.269.1	Detailed Description	730
27.269.2	Member Typedef Documentation	730
27.269.2.1	ColorArray	730
27.269.3	Member Function Documentation	730
27.269.3.1	RecommendedDisplayCIELabToRGB	730
27.269.3.2	RecommendedDisplayCIELabToRGB	731
27.269.3.3	RGBToRecommendedDisplayCIELab	731

27.269.3.4RGBToRecommendedDisplayGrayscale . . . . .	732
27.270gdcmm::SurfaceReader Class Reference . . . . .	732
27.270.1Detailed Description . . . . .	733
27.270.2Constructor & Destructor Documentation . . . . .	734
27.270.2.1SurfaceReader . . . . .	734
27.270.2.2~SurfaceReader . . . . .	734
27.270.3Member Function Documentation . . . . .	734
27.270.3.1GetNumberOfSurfaces . . . . .	734
27.270.3.2Read . . . . .	734
27.270.3.3ReadPointMacro . . . . .	734
27.270.3.4ReadSurface . . . . .	734
27.270.3.5ReadSurfaces . . . . .	734
27.271gdcmm::SurfaceWriter Class Reference . . . . .	734
27.271.1Detailed Description . . . . .	736
27.271.2Constructor & Destructor Documentation . . . . .	736
27.271.2.1SurfaceWriter . . . . .	736
27.271.2.2~SurfaceWriter . . . . .	736
27.271.3Member Function Documentation . . . . .	736
27.271.3.1ComputeNumberOfSurfaces . . . . .	736
27.271.3.2GetNumberOfSurfaces . . . . .	736
27.271.3.3PrepareWrite . . . . .	736
27.271.3.4PrepareWritePointMacro . . . . .	736
27.271.3.5SetNumberOfSurfaces . . . . .	736
27.271.3.6Write . . . . .	736
27.271.4Member Data Documentation . . . . .	736
27.271.4.1NumberOfSurfaces . . . . .	736
27.272gdcmm::SwapCode Class Reference . . . . .	736
27.272.1Detailed Description . . . . .	737
27.272.2Member Enumeration Documentation . . . . .	737
27.272.2.1SwapCodeType . . . . .	737
27.272.3Constructor & Destructor Documentation . . . . .	738
27.272.3.1SwapCode . . . . .	738
27.272.4Member Function Documentation . . . . .	738
27.272.4.1GetIndex . . . . .	738
27.272.4.2GetSwapCodeString . . . . .	738
27.272.4.3operator SwapCode::SwapCodeType . . . . .	738
27.272.5Friends And Related Function Documentation . . . . .	738



27.272.5.1operator<<	738
27.273dcm::SwapperDoOp Class Reference	738
27.273.1Member Function Documentation	738
27.273.1.1Swap	738
27.273.1.2SwapArray	738
27.274dcm::SwapperNoOp Class Reference	739
27.274.1Detailed Description	739
27.274.2Member Function Documentation	739
27.274.2.1Swap	739
27.274.2.2SwapArray	739
27.275dcm::System Class Reference	739
27.275.1Detailed Description	740
27.275.2Member Function Documentation	740
27.275.2.1DeleteDirectory	740
27.275.2.2EncodeBytes	741
27.275.2.3FileExists	741
27.275.2.4FileIsDirectory	741
27.275.2.5FileIsSymlink	741
27.275.2.6FileSize	741
27.275.2.7FileTime	741
27.275.2.8FormatDateTime	741
27.275.2.9GetCurrentDateTime	742
27.275.2.10GetCurrentModuleFileName	742
27.275.2.11GetCurrentProcessFileName	742
27.275.2.12GetCurrentResourcesDirectory	742
27.275.2.13GetCWD	742
27.275.2.14GetHostName	742
27.275.2.15GetLastError	742
27.275.2.16GetLocaleCharset	742
27.275.2.17GetPermissions	742
27.275.2.18GetTimezoneOffsetFromUTC	742
27.275.2.19MakeDirectory	743
27.275.2.20ParseDateTime	743
27.275.2.21ParseDateTime	743
27.275.2.22RemoveFile	743
27.275.2.23SetPermissions	743
27.275.2.24StrCaseCmp	743

27.275.2.25	StrNCaseCmp	. . . . .	743
27.275.2.26	StrSep	. . . . .	743
27.275.2.27	StrTokR	. . . . .	743
27.276	dcm::Table Class Reference	. . . . .	744
27.276.1	Detailed Description	. . . . .	744
27.276.2	Member Typedef Documentation	. . . . .	744
27.276.2.1	MapTableEntry	. . . . .	744
27.276.3	Constructor & Destructor Documentation	. . . . .	744
27.276.3.1	Table	. . . . .	744
27.276.3.2	~Table	. . . . .	744
27.276.4	Member Function Documentation	. . . . .	744
27.276.4.1	GetTableEntry	. . . . .	744
27.276.4.2	InsertEntry	. . . . .	744
27.276.5	Friends And Related Function Documentation	. . . . .	744
27.276.5.1	operator<<	. . . . .	744
27.277	dcm::TableEntry Class Reference	. . . . .	745
27.277.1	Detailed Description	. . . . .	745
27.277.2	Constructor & Destructor Documentation	. . . . .	745
27.277.2.1	TableEntry	. . . . .	745
27.277.2.2	~TableEntry	. . . . .	745
27.278	dcm::TableReader Class Reference	. . . . .	745
27.278.1	Detailed Description	. . . . .	746
27.278.2	Constructor & Destructor Documentation	. . . . .	746
27.278.2.1	TableReader	. . . . .	746
27.278.2.2	~TableReader	. . . . .	746
27.278.3	Member Function Documentation	. . . . .	746
27.278.3.1	CharacterDataHandler	. . . . .	746
27.278.3.2	EndElement	. . . . .	746
27.278.3.3	GetDefs	. . . . .	747
27.278.3.4	GetFilename	. . . . .	747
27.278.3.5	HandleIOD	. . . . .	747
27.278.3.6	HandleIODEntry	. . . . .	747
27.278.3.7	HandleMacro	. . . . .	747
27.278.3.8	HandleMacroEntry	. . . . .	747
27.278.3.9	HandleMacroEntryDescription	. . . . .	747
27.278.3.10	HandleModule	. . . . .	747
27.278.3.11	HandleModuleEntry	. . . . .	747

27.278.3.1HandleModuleEntryDescription . . . . .	747
27.278.3.1HandleModuleInclude . . . . .	747
27.278.3.1Read . . . . .	747
27.278.3.1SetFilename . . . . .	747
27.278.3.1StartElement . . . . .	747
27.279dcm::network::TableRow Class Reference . . . . .	747
27.279.1Constructor & Destructor Documentation . . . . .	748
27.279.1.1TableRow . . . . .	748
27.279.1.2~TableRow . . . . .	748
27.279.2Member Data Documentation . . . . .	748
27.279.2.1transitions . . . . .	748
27.280dcm::Tag Class Reference . . . . .	749
27.280.1Detailed Description . . . . .	750
27.280.2Constructor & Destructor Documentation . . . . .	751
27.280.2.1Tag . . . . .	751
27.280.2.2Tag . . . . .	751
27.280.2.3Tag . . . . .	751
27.280.3Member Function Documentation . . . . .	751
27.280.3.1GetElement . . . . .	751
27.280.3.2GetElementTag . . . . .	751
27.280.3.3GetGroup . . . . .	752
27.280.3.4GetLength . . . . .	752
27.280.3.5GetPrivateCreator . . . . .	752
27.280.3.6IsGroupLength . . . . .	752
27.280.3.7IsGroupXX . . . . .	752
27.280.3.8IsIllegal . . . . .	752
27.280.3.9IsPrivate . . . . .	752
27.280.3.10IsPrivateCreator . . . . .	753
27.280.3.11IsPublic . . . . .	753
27.280.3.12operator"!=" . . . . .	753
27.280.3.13operator< . . . . .	753
27.280.3.14operator<= . . . . .	753
27.280.3.15operator= . . . . .	753
27.280.3.16operator== . . . . .	753
27.280.3.17operator[] . . . . .	753
27.280.3.18operator[] . . . . .	753
27.280.3.19PrintAsContinuousString . . . . .	753

27.280.3.20	PrintAsContinuousUpperCaseString . . . . .	754
27.280.3.21	PrintAsPipeSeparatedString . . . . .	754
27.280.3.22	Read . . . . .	754
27.280.3.23	ReadFromCommaSeparatedString . . . . .	754
27.280.3.24	ReadFromContinuousString . . . . .	754
27.280.3.25	ReadFromPipeSeparatedString . . . . .	754
27.280.3.26	SetElement . . . . .	754
27.280.3.27	SetElementTag . . . . .	754
27.280.3.28	SetElementTag . . . . .	755
27.280.3.29	SetGroup . . . . .	755
27.280.3.30	SetPrivateCreator . . . . .	755
27.280.3.31	Write . . . . .	755
27.280.4	Friends And Related Function Documentation . . . . .	755
27.280.4.1	operator<< . . . . .	755
27.280.4.2	operator>> . . . . .	755
27.280.5	Member Data Documentation . . . . .	755
27.280.5.1	bytes . . . . .	755
27.280.5.2	tag . . . . .	755
27.280.5.3	tags . . . . .	755
27.281	gdcmm::TagPath Class Reference . . . . .	756
27.281.1	Detailed Description . . . . .	756
27.281.2	Constructor & Destructor Documentation . . . . .	756
27.281.2.1	TagPath . . . . .	756
27.281.2.2	~TagPath . . . . .	756
27.281.3	Member Function Documentation . . . . .	756
27.281.3.1	ConstructFromString . . . . .	756
27.281.3.2	ConstructFromTagList . . . . .	756
27.281.3.3	IsValid . . . . .	756
27.281.3.4	Print . . . . .	757
27.281.3.5	Push . . . . .	757
27.281.3.6	Push . . . . .	757
27.282	gdcmm::Testing Class Reference . . . . .	757
27.282.1	Detailed Description . . . . .	758
27.282.2	Member Typedef Documentation . . . . .	758
27.282.2.1	MD5DataImagesType . . . . .	758
27.282.2.2	MediaStorageDataFilesType . . . . .	758
27.282.3	Constructor & Destructor Documentation . . . . .	758

27.282.3.1Testing . . . . .	758
27.282.3.2~Testing . . . . .	758
27.282.4Member Function Documentation . . . . .	758
27.282.4.1ComputeFileMD5 . . . . .	758
27.282.4.2ComputeMD5 . . . . .	759
27.282.4.3GetDataExtraRoot . . . . .	759
27.282.4.4GetDataRoot . . . . .	759
27.282.4.5GetFileName . . . . .	759
27.282.4.6GetFileNames . . . . .	759
27.282.4.7GetLossyFlagFromFile . . . . .	759
27.282.4.8GetMD5DataImage . . . . .	759
27.282.4.9GetMD5DataImages . . . . .	759
27.282.4.10GetMD5FromBrokenFile . . . . .	759
27.282.4.10GetMD5FromFile . . . . .	760
27.282.4.10GetMediaStorageDataFile . . . . .	760
27.282.4.10GetMediaStorageDataFiles . . . . .	760
27.282.4.10GetMediaStorageFromFile . . . . .	760
27.282.4.10GetNumberOfFileNames . . . . .	760
27.282.4.10GetNumberOfMD5DataImages . . . . .	760
27.282.4.10GetNumberOfMediaStorageDataFiles . . . . .	760
27.282.4.10GetPixelSpacingDataRoot . . . . .	760
27.282.4.10GetSelectedTagsOffsetFromFile . . . . .	760
27.282.4.20GetSourceDirectory . . . . .	760
27.282.4.20GetStreamOffsetFromFile . . . . .	760
27.282.4.20GetTempDirectory . . . . .	760
27.282.4.20GetTempDirectoryW . . . . .	760
27.282.4.20GetTempFilename . . . . .	760
27.282.4.20GetTempFilenameW . . . . .	760
27.282.4.20Print . . . . .	761
27.283dcm::Trace Class Reference . . . . .	761
27.283.1Detailed Description . . . . .	762
27.283.2Constructor & Destructor Documentation . . . . .	762
27.283.2.1Trace . . . . .	762
27.283.2.2~Trace . . . . .	762
27.283.3Member Function Documentation . . . . .	762
27.283.3.1DebugOff . . . . .	762
27.283.3.2DebugOn . . . . .	762

27.283.3.3ErrorOff . . . . .	762
27.283.3.4ErrorOn . . . . .	762
27.283.3.5GetDebugFlag . . . . .	762
27.283.3.6GetDebugStream . . . . .	762
27.283.3.7GetErrorFlag . . . . .	762
27.283.3.8GetErrorStream . . . . .	762
27.283.3.9GetStream . . . . .	762
27.283.3.10GetWarningFlag . . . . .	763
27.283.3.10GetWarningStream . . . . .	763
27.283.3.12SetDebug . . . . .	763
27.283.3.13SetDebugStream . . . . .	763
27.283.3.13SetError . . . . .	763
27.283.3.13SetErrorStream . . . . .	763
27.283.3.13SetStream . . . . .	763
27.283.3.13SetStreamToFile . . . . .	763
27.283.3.13SetWarning . . . . .	763
27.283.3.13SetWarningStream . . . . .	763
27.283.3.20WarningOff . . . . .	764
27.283.3.20WarningOn . . . . .	764
27.284dcm::TransferSyntax Class Reference . . . . .	764
27.284.1Detailed Description . . . . .	765
27.284.2Member Enumeration Documentation . . . . .	766
27.284.2.1NegociatedType . . . . .	766
27.284.2.2TSType . . . . .	766
27.284.3Constructor & Destructor Documentation . . . . .	766
27.284.3.1TransferSyntax . . . . .	766
27.284.4Member Function Documentation . . . . .	766
27.284.4.1CanStoreLossy . . . . .	766
27.284.4.2GetNegociatedType . . . . .	767
27.284.4.3GetString . . . . .	767
27.284.4.4GetSwapCode . . . . .	767
27.284.4.5GetTSString . . . . .	767
27.284.4.6GetTSType . . . . .	767
27.284.4.7IsEncapsulated . . . . .	767
27.284.4.8IsEncoded . . . . .	767
27.284.4.9IsExplicit . . . . .	767
27.284.4.10IsImplicit . . . . .	767

27.284.4.1	Lossless	. 767
27.284.4.1	Lossy	. 767
27.284.4.1	Valid	. 767
27.284.4.1	operator TType	. 767
27.284.5	Friends And Related Function Documentation	. 767
27.284.5.1	operator<<	. 768
27.285	dcm::network::TransferSyntaxSub Class Reference	. 768
27.285.1	Detailed Description	. 768
27.285.2	Constructor & Destructor Documentation	. 768
27.285.2.1	TransferSyntaxSub	. 768
27.285.3	Member Function Documentation	. 768
27.285.3.1	GetName	. 768
27.285.3.2	operator==	. 768
27.285.3.3	Print	. 768
27.285.3.4	Read	. 768
27.285.3.5	SetName	. 768
27.285.3.6	SetNameFromUID	. 768
27.285.3.7	Size	. 769
27.285.3.8	Write	. 769
27.286	dcm::network::Transition Struct Reference	. 769
27.286.1	Constructor & Destructor Documentation	. 769
27.286.1.1	Transition	. 770
27.286.1.2	~Transition	. 770
27.286.1.3	Transition	. 770
27.286.2	Member Function Documentation	. 770
27.286.2.1	MakeNew	. 770
27.286.3	Member Data Documentation	. 770
27.286.3.1	mAction	. 770
27.286.3.2	mEnd	. 770
27.287	dcm::Type Class Reference	. 770
27.287.1	Detailed Description	. 771
27.287.2	Member Enumeration Documentation	. 771
27.287.2.1	TypeType	. 771
27.287.3	Constructor & Destructor Documentation	. 772
27.287.3.1	Type	. 772
27.287.4	Member Function Documentation	. 772
27.287.4.1	GetTypeString	. 772

27.287.4.2GetTypeType . . . . .	772
27.287.4.3operator TypeType . . . . .	772
27.287.5Friends And Related Function Documentation . . . . .	772
27.287.5.1operator<< . . . . .	772
27.288gdcmm::UI Struct Reference . . . . .	772
27.288.1Friends And Related Function Documentation . . . . .	772
27.288.1.1operator<< . . . . .	772
27.288.2Member Data Documentation . . . . .	772
27.288.2.1Internal . . . . .	772
27.289gdcmm::UIDGenerator Class Reference . . . . .	773
27.289.1Detailed Description . . . . .	773
27.289.2Constructor & Destructor Documentation . . . . .	773
27.289.2.1UIDGenerator . . . . .	773
27.289.3Member Function Documentation . . . . .	774
27.289.3.1Generate . . . . .	774
27.289.3.2GenerateUUID . . . . .	774
27.289.3.3GetGDCMUID . . . . .	774
27.289.3.4GetRoot . . . . .	774
27.289.3.5IsValid . . . . .	774
27.289.3.6SetRoot . . . . .	774
27.290gdcmm::UIDs Class Reference . . . . .	774
27.290.1Detailed Description . . . . .	779
27.290.2Member Typedef Documentation . . . . .	779
27.290.2.1TransferSyntaxStringsType . . . . .	779
27.290.3Member Enumeration Documentation . . . . .	779
27.290.3.1TSName . . . . .	779
27.290.3.2TSType . . . . .	786
27.290.4Member Function Documentation . . . . .	792
27.290.4.1GetName . . . . .	792
27.290.4.2GetNumberOfTransferSyntaxStrings . . . . .	793
27.290.4.3GetString . . . . .	793
27.290.4.4GetTransferSyntaxString . . . . .	793
27.290.4.5GetTransferSyntaxStrings . . . . .	793
27.290.4.6GetUIDName . . . . .	793
27.290.4.7GetUIDString . . . . .	793
27.290.4.8operator TSType . . . . .	793
27.290.4.9SetFromUID . . . . .	793



27.294	dcm::network::ULAction Class Reference	793
27.291.1	Detailed Description	795
27.291.2	Constructor & Destructor Documentation	795
27.291.2.1	ULAction	795
27.291.2.2	~ULAction	795
27.291.3	Member Function Documentation	795
27.291.3.1	PerformAction	795
27.294	dcm::network::ULActionAA1 Class Reference	796
27.292.1	Member Function Documentation	796
27.292.1.1	PerformAction	796
27.294	dcm::network::ULActionAA2 Class Reference	797
27.293.1	Member Function Documentation	797
27.293.1.1	PerformAction	798
27.294	dcm::network::ULActionAA3 Class Reference	798
27.294.1	Member Function Documentation	799
27.294.1.1	PerformAction	799
27.294	dcm::network::ULActionAA4 Class Reference	799
27.295.1	Member Function Documentation	800
27.295.1.1	PerformAction	800
27.294	dcm::network::ULActionAA5 Class Reference	800
27.296.1	Member Function Documentation	801
27.296.1.1	PerformAction	801
27.294	dcm::network::ULActionAA6 Class Reference	801
27.297.1	Member Function Documentation	802
27.297.1.1	PerformAction	802
27.294	dcm::network::ULActionAA7 Class Reference	803
27.298.1	Member Function Documentation	803
27.298.1.1	PerformAction	803
27.294	dcm::network::ULActionAA8 Class Reference	804
27.299.1	Member Function Documentation	804
27.299.1.1	PerformAction	805
27.300	dcm::network::ULActionAE1 Class Reference	805
27.300.1	Member Function Documentation	806
27.300.1.1	PerformAction	806
27.304	dcm::network::ULActionAE2 Class Reference	806
27.301.1	Member Function Documentation	807
27.301.1.1	PerformAction	807

27.302	dcm::network::ULActionAE3 Class Reference	807
27.302.1	Member Function Documentation	808
27.302.1.1	PerformAction	808
27.303	dcm::network::ULActionAE4 Class Reference	808
27.303.1	Member Function Documentation	809
27.303.1.1	PerformAction	809
27.304	dcm::network::ULActionAE5 Class Reference	810
27.304.1	Member Function Documentation	810
27.304.1.1	PerformAction	810
27.305	dcm::network::ULActionAE6 Class Reference	811
27.305.1	Member Function Documentation	811
27.305.1.1	PerformAction	812
27.306	dcm::network::ULActionAE7 Class Reference	812
27.306.1	Member Function Documentation	813
27.306.1.1	PerformAction	813
27.307	dcm::network::ULActionAE8 Class Reference	813
27.307.1	Member Function Documentation	814
27.307.1.1	PerformAction	814
27.308	dcm::network::ULActionAR1 Class Reference	814
27.308.1	Member Function Documentation	815
27.308.1.1	PerformAction	815
27.309	dcm::network::ULActionAR10 Class Reference	815
27.309.1	Member Function Documentation	816
27.309.1.1	PerformAction	816
27.310	dcm::network::ULActionAR2 Class Reference	817
27.310.1	Member Function Documentation	817
27.310.1.1	PerformAction	817
27.311	dcm::network::ULActionAR3 Class Reference	818
27.311.1	Member Function Documentation	818
27.311.1.1	PerformAction	819
27.312	dcm::network::ULActionAR4 Class Reference	819
27.312.1	Member Function Documentation	820
27.312.1.1	PerformAction	820
27.313	dcm::network::ULActionAR5 Class Reference	820
27.313.1	Member Function Documentation	821
27.313.1.1	PerformAction	821
27.314	dcm::network::ULActionAR6 Class Reference	821

27.314.1Member Function Documentation . . . . .	822
27.314.1.1PerformAction . . . . .	822
27.315dcm::network::ULActionAR7 Class Reference . . . . .	822
27.315.1Member Function Documentation . . . . .	823
27.315.1.1PerformAction . . . . .	823
27.316dcm::network::ULActionAR8 Class Reference . . . . .	824
27.316.1Member Function Documentation . . . . .	824
27.316.1.1PerformAction . . . . .	824
27.317dcm::network::ULActionAR9 Class Reference . . . . .	825
27.317.1Member Function Documentation . . . . .	825
27.317.1.1PerformAction . . . . .	826
27.318dcm::network::ULActionDT1 Class Reference . . . . .	826
27.318.1Member Function Documentation . . . . .	827
27.318.1.1PerformAction . . . . .	827
27.319dcm::network::ULActionDT2 Class Reference . . . . .	827
27.319.1Member Function Documentation . . . . .	828
27.319.1.1PerformAction . . . . .	828
27.320dcm::network::ULBasicCallback Class Reference . . . . .	828
27.320.1Detailed Description . . . . .	829
27.320.2Constructor & Destructor Documentation . . . . .	829
27.320.2.1ULBasicCallback . . . . .	829
27.320.2.2~ULBasicCallback . . . . .	829
27.320.3Member Function Documentation . . . . .	829
27.320.3.1GetDataSets . . . . .	829
27.320.3.2GetResponses . . . . .	829
27.320.3.3HandleDataSet . . . . .	829
27.320.3.4HandleResponse . . . . .	830
27.321dcm::network::ULConnection Class Reference . . . . .	830
27.321.1Detailed Description . . . . .	831
27.321.2Constructor & Destructor Documentation . . . . .	831
27.321.2.1ULConnection . . . . .	831
27.321.2.2~ULConnection . . . . .	831
27.321.3Member Function Documentation . . . . .	831
27.321.3.1AddAcceptedPresentationContext . . . . .	831
27.321.3.2FindContext . . . . .	831
27.321.3.3GetAcceptedPresentationContexts . . . . .	831
27.321.3.4GetAcceptedPresentationContexts . . . . .	831

27.321.3.5	GetConnectionInfo	831
27.321.3.6	GetMaxPDUSize	831
27.321.3.7	GetPresentationContextACByID	831
27.321.3.8	GetPresentationContextIDFromPresentationContext	831
27.321.3.9	GetPresentationContextRQByID	831
27.321.3.10	GetPresentationContexts	832
27.321.3.10	GetProtocol	832
27.321.3.10	GetState	832
27.321.3.10	GetTimer	832
27.321.3.11	InitializeConnection	832
27.321.3.11	InitializeIncomingConnection	832
27.321.3.11	SetMaxPDUSize	832
27.321.3.11	SetPresentationContexts	832
27.321.3.11	SetPresentationContexts	832
27.321.3.11	SetState	832
27.321.3.11	StopProtocol	832
27.321.4	Friends And Related Function Documentation	832
27.321.4.1	ULActionAE6	832
27.321.4.2	ULConnectionManager	832
27.322	dcm::network::ULConnectionCallback Class Reference	832
27.322.1	Detailed Description	833
27.322.2	Constructor & Destructor Documentation	833
27.322.2.1	ULConnectionCallback	833
27.322.2.2	~ULConnectionCallback	834
27.322.3	Member Function Documentation	834
27.322.3.1	DataSetHandled	834
27.322.3.2	DataSetHandles	834
27.322.3.3	HandleDataSet	834
27.322.3.4	HandleResponse	834
27.322.3.5	ResetHandledDataSet	834
27.322.3.6	SetImplicitFlag	834
27.322.4	Member Data Documentation	834
27.322.4.1	Implicit	834
27.323	dcm::network::ULConnectionInfo Class Reference	834
27.323.1	Detailed Description	835
27.323.2	Constructor & Destructor Documentation	835
27.323.2.1	ULConnectionInfo	835

27.323.3	Member Function Documentation	835
27.323.3.1	GetCalledAETitle	835
27.323.3.2	GetCalledComputerName	835
27.323.3.3	GetCalledIPAddress	835
27.323.3.4	GetCalledIPPort	835
27.323.3.5	GetCallingAETitle	835
27.323.3.6	GetMaxPDULength	835
27.323.3.7	Initialize	835
27.323.3.8	SetMaxPDULength	835
27.324	dcm::network::ULConnectionManager Class Reference	835
27.324.1	Detailed Description	837
27.324.2	Constructor & Destructor Documentation	837
27.324.2.1	ULConnectionManager	837
27.324.2.2	~ULConnectionManager	837
27.324.3	Member Function Documentation	837
27.324.3.1	BreakConnection	837
27.324.3.2	BreakConnectionNow	837
27.324.3.3	EstablishConnection	837
27.324.3.4	EstablishConnectionMove	838
27.324.3.5	SendEcho	838
27.324.3.6	SendFind	838
27.324.3.7	SendFind	838
27.324.3.8	SendMove	838
27.324.3.9	SendMove	838
27.324.3.10	SendStore	838
27.324.3.11	SendStore	838
27.325	dcm::network::ULEvent Class Reference	838
27.325.1	Detailed Description	838
27.325.2	Constructor & Destructor Documentation	839
27.325.2.1	ULEvent	839
27.325.2.2	ULEvent	839
27.325.2.3	~ULEvent	839
27.325.3	Member Function Documentation	839
27.325.3.1	GetEvent	839
27.325.3.2	GetPDUs	839
27.325.3.3	SetEvent	839
27.325.3.4	SetPDU	839

27.326	dcm::network::ULTransitionTable Class Reference	839
27.326.1	Detailed Description	839
27.326.2	Constructor & Destructor Documentation	840
27.326.2.1	ULTransitionTable	840
27.326.3	Member Function Documentation	840
27.326.3.1	HandleEvent	840
27.326.3.2	PrintTable	840
27.327	dcm::network::ULWritingCallback Class Reference	840
27.327.1	Constructor & Destructor Documentation	841
27.327.1.1	ULWritingCallback	841
27.327.1.2	~ULWritingCallback	841
27.327.2	Member Function Documentation	841
27.327.2.1	HandleDataSet	841
27.327.2.2	HandleResponse	841
27.327.2.3	SetDirectory	841
27.328	dcm::UNExplicitDataElement Class Reference	842
27.328.1	Detailed Description	843
27.328.2	Member Function Documentation	843
27.328.2.1	GetLength	843
27.328.2.2	Read	843
27.328.2.3	ReadPreValue	843
27.328.2.4	ReadValue	843
27.328.2.5	ReadWithLength	843
27.329	dcm::UNExplicitImplicitDataElement Class Reference	843
27.329.1	Detailed Description	845
27.329.2	Member Function Documentation	845
27.329.2.1	GetLength	845
27.329.2.2	Read	845
27.329.2.3	ReadPreValue	845
27.329.2.4	ReadValue	845
27.330	dcm::Unpacker12Bits Class Reference	845
27.330.1	Detailed Description	845
27.330.2	Member Function Documentation	846
27.330.2.1	Pack	846
27.330.2.2	Unpack	846
27.331	dcm::Usage Class Reference	846
27.331.1	Detailed Description	847

27.331.2	Member Enumeration Documentation	847
27.331.2.1	UsageType	847
27.331.3	Constructor & Destructor Documentation	847
27.331.3.1	Usage	847
27.331.4	Member Function Documentation	847
27.331.4.1	GetUsageString	847
27.331.4.2	GetUsageType	848
27.331.4.3	operator UsageType	848
27.331.5	Friends And Related Function Documentation	848
27.331.5.1	operator<<	848
27.332	dcm::UserEvent Class Reference	848
27.333	dcm::network::UserInformation Class Reference	849
27.333.1	Detailed Description	850
27.333.2	Constructor & Destructor Documentation	850
27.333.2.1	UserInformation	850
27.333.2.2	~UserInformation	850
27.333.3	Member Function Documentation	850
27.333.3.1	AddRoleSelectionSub	850
27.333.3.2	AddSOPClassExtendedNegociationSub	850
27.333.3.3	GetMaximumLengthSub	850
27.333.3.4	GetMaximumLengthSub	850
27.333.3.5	operator=	850
27.333.3.6	Print	850
27.333.3.7	Read	850
27.333.3.8	Size	850
27.333.3.9	Write	850
27.334	dcm::UUIDGenerator Class Reference	850
27.334.1	Detailed Description	851
27.334.2	Member Function Documentation	851
27.334.2.1	Generate	851
27.334.2.2	IsValid	851
27.335	dcm::Validate Class Reference	851
27.335.1	Detailed Description	852
27.335.2	Constructor & Destructor Documentation	852
27.335.2.1	Validate	852
27.335.2.2	~Validate	852
27.335.3	Member Function Documentation	852

27.335.3.1GetValidatedFile . . . . .	853
27.335.3.2SetFile . . . . .	853
27.335.3.3Validation . . . . .	853
27.335.4Member Data Documentation . . . . .	853
27.335.4.1F . . . . .	853
27.335.4.2V . . . . .	853
27.336dcm::Value Class Reference . . . . .	853
27.336.1Detailed Description . . . . .	854
27.336.2Constructor & Destructor Documentation . . . . .	854
27.336.2.1Value . . . . .	854
27.336.2.2~Value . . . . .	854
27.336.3Member Function Documentation . . . . .	854
27.336.3.1Clear . . . . .	855
27.336.3.2GetLength . . . . .	855
27.336.3.3operator== . . . . .	855
27.336.3.4SetLength . . . . .	855
27.336.3.5SetLengthOnly . . . . .	855
27.336.4Friends And Related Function Documentation . . . . .	855
27.336.4.1DataElement . . . . .	855
27.337dcm::ValueIO< TDE, TSwap, TType > Class Template Reference . . . . .	855
27.337.1Detailed Description . . . . .	855
27.337.2Member Function Documentation . . . . .	856
27.337.2.1Read . . . . .	856
27.337.2.2Write . . . . .	856
27.338dcm::Version Class Reference . . . . .	856
27.338.1Detailed Description . . . . .	856
27.338.2Constructor & Destructor Documentation . . . . .	856
27.338.2.1Version . . . . .	856
27.338.2.2~Version . . . . .	856
27.338.3Member Function Documentation . . . . .	856
27.338.3.1GetBuildVersion . . . . .	857
27.338.3.2GetMajorVersion . . . . .	857
27.338.3.3GetMinorVersion . . . . .	857
27.338.3.4GetVersion . . . . .	857
27.338.3.5Print . . . . .	857
27.338.4Friends And Related Function Documentation . . . . .	857
27.338.4.1operator<< . . . . .	857



27.339	dcm::VL Class Reference . . . . .	857
27.339.1	Detailed Description . . . . .	858
27.339.2	Member Typedef Documentation . . . . .	858
27.339.2.1	Type . . . . .	858
27.339.3	Constructor & Destructor Documentation . . . . .	858
27.339.3.1	VL . . . . .	858
27.339.4	Member Function Documentation . . . . .	858
27.339.4.1	GetLength . . . . .	858
27.339.4.2	GetVL16Max . . . . .	858
27.339.4.3	GetVL32Max . . . . .	858
27.339.4.4	IsOdd . . . . .	858
27.339.4.5	IsUndefined . . . . .	859
27.339.4.6	operator uint32_t . . . . .	859
27.339.4.7	operator++ . . . . .	859
27.339.4.8	operator++ . . . . .	859
27.339.4.9	operator+= . . . . .	859
27.339.4.10	Read . . . . .	859
27.339.4.11	Read16 . . . . .	859
27.339.4.12	SetToUndefined . . . . .	859
27.339.4.13	Write . . . . .	859
27.339.4.14	Write16 . . . . .	859
27.339.5	Friends And Related Function Documentation . . . . .	859
27.339.5.1	operator<< . . . . .	859
27.340	dcm::VM Class Reference . . . . .	859
27.340.1	Detailed Description . . . . .	861
27.340.2	Member Enumeration Documentation . . . . .	861
27.340.2.1	VMType . . . . .	861
27.340.3	Constructor & Destructor Documentation . . . . .	862
27.340.3.1	VM . . . . .	862
27.340.4	Member Function Documentation . . . . .	862
27.340.4.1	Compatible . . . . .	862
27.340.4.2	GetIndex . . . . .	862
27.340.4.3	GetLength . . . . .	862
27.340.4.4	GetNumberOfElementsFromArray . . . . .	862
27.340.4.5	GetVMString . . . . .	862
27.340.4.6	GetVMType . . . . .	863
27.340.4.7	GetVMTypeFromLength . . . . .	863

27.340.4.8IsValid	863
27.340.4.9operator VMType	863
27.340.5Friends And Related Function Documentation	863
27.340.5.1operator<<	863
27.340dcm::VMToLength< T > Struct Template Reference	863
27.340dcm::VR Class Reference	863
27.342.1Detailed Description	865
27.342.2Member Enumeration Documentation	865
27.342.2.1VRType	865
27.342.3Constructor & Destructor Documentation	866
27.342.3.1VR	866
27.342.4Member Function Documentation	866
27.342.4.1CanDisplay	866
27.342.4.2Compatible	867
27.342.4.3GetLength	867
27.342.4.4GetLength	867
27.342.4.5GetSize	867
27.342.4.6GetSizeof	867
27.342.4.7GetVRString	867
27.342.4.8GetVRStringFromFile	867
27.342.4.9GetVRType	867
27.342.4.10GetVRTypeFromFile	867
27.342.4.11ASCII	867
27.342.4.12ASCII2	867
27.342.4.13Binary	867
27.342.4.14Binary2	867
27.342.4.15Dual	867
27.342.4.16Swap	867
27.342.4.17Valid	867
27.342.4.18Valid	867
27.342.4.19VRFile	867
27.342.4.20operator VRType	867
27.342.4.21Read	867
27.342.4.22Write	868
27.342.5Friends And Related Function Documentation	868
27.342.5.1operator<<	868
27.340dcm::VR16ExplicitDataElement Class Reference	868

27.343.1Detailed Description	869
27.343.2Member Function Documentation	869
27.343.2.1GetLength	869
27.343.2.2Read	870
27.343.2.3ReadPreValue	870
27.343.2.4ReadValue	870
27.343.2.5ReadWithLength	870
27.344dcm::VRToEncoding< T > Struct Template Reference	870
27.345dcm::VRToType< T > Struct Template Reference	870
27.345.1Detailed Description	870
27.346dcm::VRVLSize< T > Class Template Reference	871
27.347dcm::VRVLSize< 0 > Class Template Reference	871
27.347.1Member Function Documentation	871
27.347.1.1Read	871
27.347.1.2Write	871
27.348dcm::VRVLSize< 1 > Class Template Reference	871
27.348.1Member Function Documentation	871
27.348.1.1Read	871
27.348.1.2Write	871
27.349tkGDCMImageReader Class Reference	872
27.349.1Detailed Description	874
27.349.2Constructor & Destructor Documentation	874
27.349.2.1tkGDCMImageReader	874
27.349.2.2~tkGDCMImageReader	874
27.349.3Member Function Documentation	874
27.349.3.1CanReadFile	874
27.349.3.2ExecuteData	874
27.349.3.3ExecuteInformation	874
27.349.3.4FillMedicalImageInformation	875
27.349.3.5GetDescriptiveName	875
27.349.3.6GetFileExtensions	875
27.349.3.7GetIconImage	875
27.349.3.8GetOverlay	875
27.349.3.9LoadSingleFile	875
27.349.3.10New	875
27.349.3.11PrintSelf	875
27.349.3.12requestDataCompat	875

27.349.3.18	<a href="#">RequestInformationCompat</a>	. 875
27.349.3.19	<a href="#">SetCurve</a>	. 875
27.349.3.19	<a href="#">SetFileNames</a>	. 875
27.349.3.19	<a href="#">SetFilePattern</a>	. 875
27.349.3.19	<a href="#">SetFilePrefix</a>	. 875
27.349.3.19	<a href="#">SetMedicalImageProperties</a>	. 875
27.349.3.19	<a href="#">tkBooleanMacro</a>	. 875
27.349.3.20	<a href="#">tkBooleanMacro</a>	. 875
27.349.3.21	<a href="#">tkBooleanMacro</a>	. 875
27.349.3.22	<a href="#">tkBooleanMacro</a>	. 876
27.349.3.23	<a href="#">tkBooleanMacro</a>	. 876
27.349.3.24	<a href="#">tkGetMacro</a>	. 876
27.349.3.25	<a href="#">tkGetMacro</a>	. 876
27.349.3.26	<a href="#">tkGetMacro</a>	. 876
27.349.3.27	<a href="#">tkGetMacro</a>	. 876
27.349.3.28	<a href="#">tkGetMacro</a>	. 876
27.349.3.29	<a href="#">tkGetMacro</a>	. 876
27.349.3.30	<a href="#">tkGetMacro</a>	. 876
27.349.3.31	<a href="#">tkGetMacro</a>	. 876
27.349.3.32	<a href="#">tkGetMacro</a>	. 876
27.349.3.33	<a href="#">tkGetMacro</a>	. 876
27.349.3.34	<a href="#">tkGetMacro</a>	. 876
27.349.3.35	<a href="#">tkGetObjectMacro</a>	. 876
27.349.3.36	<a href="#">tkGetObjectMacro</a>	. 876
27.349.3.37	<a href="#">tkGetObjectMacro</a>	. 876
27.349.3.38	<a href="#">tkGetObjectMacro</a>	. 876
27.349.3.39	<a href="#">tkGetStringMacro</a>	. 876
27.349.3.40	<a href="#">tkGetStringMacro</a>	. 876
27.349.3.41	<a href="#">tkGetVector3Macro</a>	. 876
27.349.3.42	<a href="#">tkGetVector6Macro</a>	. 876
27.349.3.43	<a href="#">tkSetMacro</a>	. 876
27.349.3.44	<a href="#">tkSetMacro</a>	. 876
27.349.3.45	<a href="#">tkSetMacro</a>	. 876
27.349.3.46	<a href="#">tkSetMacro</a>	. 876
27.349.3.47	<a href="#">tkSetVector6Macro</a>	. 876
27.349.3.48	<a href="#">tkTypeRevisionMacro</a>	. 877
27.349.4	<a href="#">Member Data Documentation</a>	. 877

27.349.4.1	ApplyInverseVideo	877
27.349.4.2	ApplyLookupTable	877
27.349.4.3	ApplyPlanarConfiguration	877
27.349.4.4	ApplyShiftScale	877
27.349.4.5	ApplyYBRToRGB	877
27.349.4.6	Curve	877
27.349.4.7	DirectionCosines	877
27.349.4.8	FileNames	877
27.349.4.9	ForceRescale	877
27.349.4.10	GetDataScalarType	877
27.349.4.11	GetImageDataExtent	877
27.349.4.12	GetNumberOfScalarComponents	877
27.349.4.13	GetImageFormat	877
27.349.4.14	GetImageOrientationPatient	877
27.349.4.15	GetImagePositionPatient	877
27.349.4.16	LoadIconImage	877
27.349.4.17	LoadOverlays	877
27.349.4.18	LossyFlag	877
27.349.4.19	MedicalImageProperties	877
27.349.4.20	NumberOfIconImages	877
27.349.4.21	NumberOfOverlays	877
27.349.4.22	PlanarConfiguration	877
27.349.4.23	Scale	877
27.349.4.24	Shift	878
27.350	vtkGDCMImageReader2 Class Reference	878
27.350.1	Constructor & Destructor Documentation	880
27.350.1.1	vtkGDCMImageReader2	880
27.350.1.2	~vtkGDCMImageReader2	880
27.350.2	Member Function Documentation	880
27.350.2.1	CanReadFile	880
27.350.2.2	FillMedicalImageInformation	880
27.350.2.3	GetDescriptiveName	880
27.350.2.4	GetFileExtensions	880
27.350.2.5	GetIconImage	880
27.350.2.6	GetIconImagePort	881
27.350.2.7	GetOverlay	881
27.350.2.8	GetOverlayPort	881

27.350.2.9LoadSingleFile . . . . .	881
27.350.2.10New . . . . .	881
27.350.2.11PrintSelf . . . . .	881
27.350.2.12ProcessRequest . . . . .	881
27.350.2.13RequestData . . . . .	881
27.350.2.14RequestDataCompat . . . . .	881
27.350.2.15RequestInformation . . . . .	881
27.350.2.16RequestInformationCompat . . . . .	881
27.350.2.17SetCurve . . . . .	881
27.350.2.18SetFilePattern . . . . .	881
27.350.2.19SetFilePrefix . . . . .	881
27.350.2.20SetMedicalImageProperties . . . . .	881
27.350.2.21tkBooleanMacro . . . . .	881
27.350.2.22tkBooleanMacro . . . . .	881
27.350.2.23tkBooleanMacro . . . . .	881
27.350.2.24tkBooleanMacro . . . . .	881
27.350.2.25tkBooleanMacro . . . . .	881
27.350.2.26tkGetMacro . . . . .	881
27.350.2.27tkGetMacro . . . . .	881
27.350.2.28tkGetMacro . . . . .	881
27.350.2.29tkGetMacro . . . . .	881
27.350.2.30tkGetMacro . . . . .	882
27.350.2.31tkGetMacro . . . . .	882
27.350.2.32tkGetMacro . . . . .	882
27.350.2.33tkGetMacro . . . . .	882
27.350.2.34tkGetMacro . . . . .	882
27.350.2.35tkGetMacro . . . . .	882
27.350.2.36tkGetMacro . . . . .	882
27.350.2.37tkGetObjectMacro . . . . .	882
27.350.2.38tkGetObjectMacro . . . . .	882
27.350.2.39tkGetStringMacro . . . . .	882
27.350.2.40tkGetStringMacro . . . . .	882
27.350.2.41tkGetVector3Macro . . . . .	882
27.350.2.42tkGetVector6Macro . . . . .	882
27.350.2.43tkSetMacro . . . . .	882
27.350.2.44tkSetMacro . . . . .	882
27.350.2.45tkSetMacro . . . . .	882

27.350.2.46	SetMacro	882
27.350.2.47	SetVector6Macro	882
27.350.2.48	TypeRevisionMacro	882
27.350.3	Member Data Documentation	882
27.350.3.1	ApplyInverseVideo	882
27.350.3.2	ApplyLookupTable	882
27.350.3.3	ApplyPlanarConfiguration	882
27.350.3.4	ApplyShiftScale	882
27.350.3.5	ApplyYBRToRGB	882
27.350.3.6	Curve	882
27.350.3.7	DirectionCosines	883
27.350.3.8	ForceRescale	883
27.350.3.9	IconDataScalarType	883
27.350.3.10	IconImageDataExtent	883
27.350.3.11	IconNumberOfScalarComponents	883
27.350.3.12	ImageFormat	883
27.350.3.13	ImageOrientationPatient	883
27.350.3.14	ImagePositionPatient	883
27.350.3.15	LoadIconImage	883
27.350.3.16	LoadOverlays	883
27.350.3.17	LossyFlag	883
27.350.3.18	NumberOfIconImages	883
27.350.3.19	NumberOfOverlays	883
27.350.3.20	PlanarConfiguration	883
27.350.3.21	Scale	883
27.350.3.22	Shift	883
27.351	vtkGDCMImageWriter Class Reference	883
27.351.1	Detailed Description	885
27.351.2	Member Enumeration Documentation	886
27.351.2.1	CompressionTypes	886
27.351.3	Constructor & Destructor Documentation	886
27.351.3.1	vtkGDCMImageWriter	886
27.351.3.2	~vtkGDCMImageWriter	886
27.351.4	Member Function Documentation	886
27.351.4.1	GetDescriptiveName	886
27.351.4.2	GetFileExtensions	886
27.351.4.3	GetFileName	886

27.351.4.4New	886
27.351.4.5PrintSelf	886
27.351.4.6SetDirectionCosines	886
27.351.4.7SetDirectionCosinesFromImageOrientationPatient	886
27.351.4.8SetFileNames	886
27.351.4.9SetMedicalImageProperties	887
27.351.4.10tkBooleanMacro	887
27.351.4.11tkBooleanMacro	887
27.351.4.12tkGetMacro	887
27.351.4.13tkGetMacro	887
27.351.4.14tkGetMacro	887
27.351.4.15tkGetMacro	887
27.351.4.16tkGetMacro	887
27.351.4.17tkGetMacro	887
27.351.4.18tkGetMacro	887
27.351.4.19tkGetObjectMacro	887
27.351.4.20tkGetObjectMacro	887
27.351.4.21tkGetObjectMacro	887
27.351.4.22tkGetStringMacro	887
27.351.4.23tkGetStringMacro	887
27.351.4.24tkSetMacro	887
27.351.4.25tkSetMacro	887
27.351.4.26tkSetMacro	887
27.351.4.27tkSetMacro	887
27.351.4.28tkSetMacro	887
27.351.4.29tkSetMacro	887
27.351.4.30tkSetMacro	887
27.351.4.31tkSetStringMacro	887
27.351.4.32tkSetStringMacro	888
27.351.4.33tkTypeRevisionMacro	888
27.351.4.34Write	888
27.351.4.35WriteGDCMData	888
27.351.4.36WriteSlice	888
27.352tkGDCMMedicalImageProperties Class Reference	888
27.352.1Constructor & Destructor Documentation	889
27.352.1.1tkGDCMMedicalImageProperties	889
27.352.1.2~vtkGDCMMedicalImageProperties	889



27.352.2	Member Function Documentation	889
27.352.2.1	Clear	890
27.352.2.2	GetFile	890
27.352.2.3	New	890
27.352.2.4	PrintSelf	890
27.352.2.5	PushBackFile	890
27.352.2.6	vtkTypeRevisionMacro	890
27.352.3	Friends And Related Function Documentation	890
27.352.3.1	vtkGDCMImageReader	890
27.352.3.2	vtkGDCMImageReader2	890
27.352.3.3	vtkGDCMImageWriter	890
27.353	vtkGDCMPolyDataReader Class Reference	890
27.353.1	Detailed Description	892
27.353.2	Constructor & Destructor Documentation	892
27.353.2.1	vtkGDCMPolyDataReader	892
27.353.2.2	~vtkGDCMPolyDataReader	892
27.353.3	Member Function Documentation	892
27.353.3.1	FillMedicalImageInformation	892
27.353.3.2	New	892
27.353.3.3	PrintSelf	892
27.353.3.4	RequestData	892
27.353.3.5	RequestData_HemodynamicWaveformStorage	892
27.353.3.6	RequestData_RTStructureSetStorage	892
27.353.3.7	RequestInformation	892
27.353.3.8	RequestInformation_HemodynamicWaveformStorage	892
27.353.3.9	RequestInformation_RTStructureSetStorage	892
27.353.3.10	GetObjectMacro	892
27.353.3.11	vtkGetObjectMacro	892
27.353.3.12	GetStringMacro	893
27.353.3.13	SetStringMacro	893
27.353.3.14	vtkTypeRevisionMacro	893
27.353.4	Member Data Documentation	893
27.353.4.1	FileName	893
27.353.4.2	MedicalImageProperties	893
27.353.4.3	RTStructSetProperties	893
27.354	vtkGDCMPolyDataWriter Class Reference	893
27.354.1	Detailed Description	895

27.354.2	Constructor & Destructor Documentation	895
27.354.2.1	vtkGDCMPolyDataWriter	895
27.354.2.2	~vtkGDCMPolyDataWriter	895
27.354.3	Member Function Documentation	895
27.354.3.1	InitializeRTStructSet	895
27.354.3.2	New	895
27.354.3.3	PrintSelf	895
27.354.3.4	SetMedicalImageProperties	895
27.354.3.5	SetNumberOfInputPorts	895
27.354.3.6	SetRTStructSetProperties	895
27.354.3.7	vtkTypeRevisionMacro	896
27.354.3.8	WriteData	896
27.354.3.9	WriteRTSTRUCTData	896
27.354.3.10	WriteRTSTRUCTInfo	896
27.354.4	Member Data Documentation	896
27.354.4.1	MedicalImageProperties	896
27.354.4.2	RTStructSetProperties	896
27.355	vtkGDCMTesting Class Reference	896
27.355.1	Detailed Description	897
27.355.2	Member Typedef Documentation	898
27.355.2.1	MD5MetalImagesType	898
27.355.3	Constructor & Destructor Documentation	898
27.355.3.1	vtkGDCMTesting	898
27.355.3.2	~vtkGDCMTesting	898
27.355.4	Member Function Documentation	898
27.355.4.1	GetGDCMDataRoot	898
27.355.4.2	GetMD5MetalImage	898
27.355.4.3	GetMHDMD5FromFile	898
27.355.4.4	GetNumberOfMD5MetalImages	898
27.355.4.5	GetRAWMD5FromFile	898
27.355.4.6	GetVTKDataRoot	898
27.355.4.7	New	898
27.355.4.8	PrintSelf	898
27.355.4.9	vtkTypeRevisionMacro	898
27.356	vtkGDCMThreadedImageReader Class Reference	898
27.356.1	Constructor & Destructor Documentation	900
27.356.1.1	vtkGDCMThreadedImageReader	900

27.356.1.2~vtkGDCMThreadedImageReader	900
27.356.2Member Function Documentation	900
27.356.2.1ExecuteData	900
27.356.2.2ExecuteInformation	900
27.356.2.3New	900
27.356.2.4PrintSelf	900
27.356.2.5ReadFiles	900
27.356.2.6RequestDataCompat	900
27.356.2.7vtkBooleanMacro	900
27.356.2.8vtkGetMacro	900
27.356.2.9vtkSetMacro	900
27.356.2.10vtkSetMacro	900
27.356.2.11vtkSetMacro	900
27.356.2.12vtkTypeRevisionMacro	900
27.357.1vtkGDCMThreadedImageReader2 Class Reference	901
27.357.2Constructor & Destructor Documentation	902
27.357.2.1vtkGDCMThreadedImageReader2	902
27.357.2.2~vtkGDCMThreadedImageReader2	902
27.357.3Member Function Documentation	902
27.357.3.1GetFileName	902
27.357.3.2New	902
27.357.3.3PrintSelf	903
27.357.3.4RequestInformation	903
27.357.3.5SetFileName	903
27.357.3.6SetFileNames	903
27.357.3.7SplitExtent	903
27.357.3.8ThreadedRequestData	903
27.357.3.9vtkBooleanMacro	903
27.357.3.10vtkBooleanMacro	903
27.357.3.11vtkBooleanMacro	903
27.357.3.12vtkGetMacro	903
27.357.3.13vtkGetMacro	903
27.357.3.14vtkGetMacro	903
27.357.3.15vtkGetMacro	903
27.357.3.16vtkGetMacro	903
27.357.3.17vtkGetMacro	903
27.357.3.18vtkGetMacro	903

27.357.2.19	tkGetMacro	903
27.357.2.20	tkGetObjectMacro	903
27.357.2.21	tkGetVector3Macro	903
27.357.2.22	tkGetVector3Macro	903
27.357.2.23	tkGetVector6Macro	903
27.357.2.24	tkSetMacro	903
27.357.2.25	tkSetMacro	903
27.357.2.26	tkSetMacro	903
27.357.2.27	tkSetMacro	903
27.357.2.28	tkSetMacro	904
27.357.2.29	tkSetMacro	904
27.357.2.30	tkSetMacro	904
27.357.2.31	tkSetVector3Macro	904
27.357.2.32	tkSetVector3Macro	904
27.357.2.33	tkSetVector6Macro	904
27.357.2.34	tkTypeRevisionMacro	904
27.358	tkImageColorViewer Class Reference	904
27.358.1	Detailed Description	907
27.358.2	Member Enumeration Documentation	907
27.358.2.1	anonymous enum	907
27.358.3	Constructor & Destructor Documentation	907
27.358.3.1	tkImageColorViewer	907
27.358.3.2	~tkImageColorViewer	907
27.358.4	Member Function Documentation	907
27.358.4.1	AddInput	907
27.358.4.2	AddInputConnection	907
27.358.4.3	GetColorLevel	907
27.358.4.4	GetColorWindow	907
27.358.4.5	GetInput	907
27.358.4.6	GetOffScreenRendering	908
27.358.4.7	GetOverlayVisibility	908
27.358.4.8	GetPosition	908
27.358.4.9	GetSize	908
27.358.4.10	GetSliceMax	908
27.358.4.10	GetSliceMin	908
27.358.4.10	GetSliceRange	908
27.358.4.10	GetSliceRange	908

27.358.4.10	GetSliceRange	908
27.358.4.10	GetWindowName	908
27.358.4.10	InstallPipeline	908
27.358.4.11	New	908
27.358.4.11	PrintSelf	908
27.358.4.11	Render	908
27.358.4.20	SetColorLevel	908
27.358.4.20	SetColorWindow	908
27.358.4.20	SetDisplayId	908
27.358.4.20	SetInput	908
27.358.4.20	SetInputConnection	909
27.358.4.25	SetOffScreenRendering	909
27.358.4.26	SetOverlayVisibility	909
27.358.4.27	SetParentId	909
27.358.4.28	SetPosition	909
27.358.4.29	SetPosition	909
27.358.4.30	SetRenderer	909
27.358.4.30	SetRenderWindow	909
27.358.4.30	SetSize	909
27.358.4.30	SetSize	909
27.358.4.30	SetSlice	909
27.358.4.35	SetSliceOrientation	909
27.358.4.36	SetSliceOrientationToXY	909
27.358.4.37	SetSliceOrientationToXZ	909
27.358.4.38	SetSliceOrientationToYZ	909
27.358.4.39	SetupInteractor	910
27.358.4.40	SetWindowId	910
27.358.4.41	UnInstallPipeline	910
27.358.4.42	UpdateDisplayExtent	910
27.358.4.43	UpdateOrientation	910
27.358.4.44	WTK_LEGACY	910
27.358.4.45	WTK_LEGACY	910
27.358.4.46	WTK_LEGACY	910
27.358.4.47	WTK_LEGACY	910
27.358.4.48	BooleanMacro	910
27.358.4.49	GetMacro	910
27.358.4.50	GetMacro	910

27.358.4.51tkGetObjectMacro . . . . .	910
27.358.4.52tkGetObjectMacro . . . . .	910
27.358.4.53tkGetObjectMacro . . . . .	910
27.358.4.54tkGetObjectMacro . . . . .	910
27.358.4.55tkGetObjectMacro . . . . .	910
27.358.4.56tkTypeRevisionMacro . . . . .	910
27.358.5 Friends And Related Function Documentation . . . . .	910
27.358.5.1vtkImageColorViewerCallback . . . . .	910
27.358.6 Member Data Documentation . . . . .	910
27.358.6.1FirstRender . . . . .	910
27.358.6.2ImageActor . . . . .	910
27.358.6.3Interactor . . . . .	910
27.358.6.4InteractorStyle . . . . .	911
27.358.6.5OverlayImageActor . . . . .	911
27.358.6.6Renderer . . . . .	911
27.358.6.7RenderWindow . . . . .	911
27.358.6.8Slice . . . . .	911
27.358.6.9SliceOrientation . . . . .	911
27.358.6.10WindowLevel . . . . .	911
27.359 vtkImageMapToColors16 Class Reference . . . . .	911
27.359.1 Constructor & Destructor Documentation . . . . .	913
27.359.1.1vtkImageMapToColors16 . . . . .	913
27.359.1.2~vtkImageMapToColors16 . . . . .	913
27.359.2 Member Function Documentation . . . . .	913
27.359.2.1GetMTime . . . . .	913
27.359.2.2New . . . . .	913
27.359.2.3PrintSelf . . . . .	913
27.359.2.4RequestData . . . . .	913
27.359.2.5RequestInformation . . . . .	913
27.359.2.6SetLookupTable . . . . .	913
27.359.2.7SetOutputFormatToLuminance . . . . .	913
27.359.2.8SetOutputFormatToLuminanceAlpha . . . . .	913
27.359.2.9SetOutputFormatToRGB . . . . .	913
27.359.2.10SetOutputFormatToRGBA . . . . .	913
27.359.2.11ThreadedRequestData . . . . .	913
27.359.2.12tkBooleanMacro . . . . .	913
27.359.2.13tkGetMacro . . . . .	913

27.359.2.1	tkGetMacro	. . . . .	913
27.359.2.1	tkGetMacro	. . . . .	913
27.359.2.1	tkGetObjectMacro	. . . . .	913
27.359.2.1	tkSetMacro	. . . . .	914
27.359.2.1	tkSetMacro	. . . . .	914
27.359.2.1	tkSetMacro	. . . . .	914
27.359.2.2	tkTypeRevisionMacro	. . . . .	914
27.359.3	Member Data Documentation	. . . . .	914
27.359.3.1	ActiveComponent	. . . . .	914
27.359.3.2	DataWasPassed	. . . . .	914
27.359.3.3	LookupTable	. . . . .	914
27.359.3.4	OutputFormat	. . . . .	914
27.359.3.5	PassAlphaToOutput	. . . . .	914
27.360	vtkImageMapToWindowLevelColors2 Class Reference	. . . . .	914
27.360.1	Constructor & Destructor Documentation	. . . . .	916
27.360.1.1	vtkImageMapToWindowLevelColors2	. . . . .	916
27.360.1.2	~vtkImageMapToWindowLevelColors2	. . . . .	916
27.360.2	Member Function Documentation	. . . . .	916
27.360.2.1	New	. . . . .	916
27.360.2.2	PrintSelf	. . . . .	916
27.360.2.3	RequestData	. . . . .	916
27.360.2.4	RequestInformation	. . . . .	916
27.360.2.5	ThreadedRequestData	. . . . .	916
27.360.2.6	tkGetMacro	. . . . .	916
27.360.2.7	tkGetMacro	. . . . .	916
27.360.2.8	tkSetMacro	. . . . .	916
27.360.2.9	tkSetMacro	. . . . .	916
27.360.2.10	tkTypeRevisionMacro	. . . . .	916
27.360.3	Member Data Documentation	. . . . .	916
27.360.3.1	Level	. . . . .	916
27.360.3.2	Window	. . . . .	916
27.361	vtkImagePlanarComponentsToComponents Class Reference	. . . . .	916
27.361.1	Constructor & Destructor Documentation	. . . . .	918
27.361.1.1	vtkImagePlanarComponentsToComponents	. . . . .	918
27.361.1.2	~vtkImagePlanarComponentsToComponents	. . . . .	918
27.361.2	Member Function Documentation	. . . . .	918
27.361.2.1	New	. . . . .	918

27.361.2.2	PrintSelf	918
27.361.2.3	RequestData	918
27.361.2.4	vtkTypeRevisionMacro	918
27.362	vtkImageRGBToYBR Class Reference	918
27.362.1	Constructor & Destructor Documentation	919
27.362.1.1	vtkImageRGBToYBR	919
27.362.1.2	~vtkImageRGBToYBR	919
27.362.2	Member Function Documentation	919
27.362.2.1	New	919
27.362.2.2	PrintSelf	919
27.362.2.3	ThreadedExecute	919
27.362.2.4	vtkTypeRevisionMacro	919
27.363	vtkImageYBRToRGB Class Reference	920
27.363.1	Constructor & Destructor Documentation	921
27.363.1.1	vtkImageYBRToRGB	921
27.363.1.2	~vtkImageYBRToRGB	921
27.363.2	Member Function Documentation	921
27.363.2.1	New	921
27.363.2.2	PrintSelf	921
27.363.2.3	ThreadedExecute	921
27.363.2.4	vtkTypeRevisionMacro	921
27.364	vtkLookupTable16 Class Reference	921
27.364.1	Constructor & Destructor Documentation	922
27.364.1.1	vtkLookupTable16	922
27.364.1.2	~vtkLookupTable16	922
27.364.2	Member Function Documentation	922
27.364.2.1	Build	923
27.364.2.2	GetPointer	923
27.364.2.3	MapScalarsThroughTable2	923
27.364.2.4	New	923
27.364.2.5	PrintSelf	923
27.364.2.6	SetNumberOfTableValues	923
27.364.2.7	vtkTypeRevisionMacro	923
27.364.2.8	WritePointer	923
27.364.3	Member Data Documentation	923
27.364.3.1	Table16	923
27.365	vtkRTStructSetProperties Class Reference	923



27.365.1 Detailed Description . . . . .	925
27.365.2 Constructor & Destructor Documentation . . . . .	925
27.365.2.1 vtkRTStructSetProperties . . . . .	925
27.365.2.2 ~vtkRTStructSetProperties . . . . .	925
27.365.3 Member Function Documentation . . . . .	925
27.365.3.1 AddContourReferencedFrameOfReference . . . . .	926
27.365.3.2 AddReferencedFrameOfReference . . . . .	926
27.365.3.3 AddStructureSetROI . . . . .	926
27.365.3.4 AddStructureSetROIObservation . . . . .	926
27.365.3.5 Clear . . . . .	926
27.365.3.6 DeepCopy . . . . .	926
27.365.3.7 GetContourReferencedFrameOfReferenceClassUID . . . . .	926
27.365.3.8 GetContourReferencedFrameOfReferenceInstanceUID . . . . .	926
27.365.3.9 GetNumberOfContourReferencedFrameOfReferences . . . . .	926
27.365.3.10 GetNumberOfContourReferencedFrameOfReferences . . . . .	926
27.365.3.11 GetNumberOfReferencedFrameOfReferences . . . . .	926
27.365.3.12 GetNumberOfStructureSetROIs . . . . .	926
27.365.3.13 GetReferencedFrameOfReferenceClassUID . . . . .	926
27.365.3.14 GetReferencedFrameOfReferenceInstanceUID . . . . .	926
27.365.3.15 SetStructureSetObservationNumber . . . . .	926
27.365.3.16 GetStructureSetROIDescription . . . . .	926
27.365.3.17 GetStructureSetROIGenerationAlgorithm . . . . .	926
27.365.3.18 GetStructureSetROIName . . . . .	926
27.365.3.19 GetStructureSetROINumber . . . . .	926
27.365.3.20 GetStructureSetROIObservationLabel . . . . .	926
27.365.3.21 GetStructureSetROIRefFrameRefUID . . . . .	926
27.365.3.22 GetStructureSetRTROIInterpretedType . . . . .	926
27.365.3.23 New . . . . .	927
27.365.3.24 PrintSelf . . . . .	927
27.365.3.25 kGetStringMacro . . . . .	927
27.365.3.26 kGetStringMacro . . . . .	927
27.365.3.27 kGetStringMacro . . . . .	927
27.365.3.28 kGetStringMacro . . . . .	927
27.365.3.29 kGetStringMacro . . . . .	927
27.365.3.30 kGetStringMacro . . . . .	927
27.365.3.31 kGetStringMacro . . . . .	927
27.365.3.32 kGetStringMacro . . . . .	927

27.365.3.31kGetStringMacro . . . . .	927
27.365.3.34kSetStringMacro . . . . .	927
27.365.3.35kSetStringMacro . . . . .	927
27.365.3.36kSetStringMacro . . . . .	927
27.365.3.37kSetStringMacro . . . . .	927
27.365.3.38kSetStringMacro . . . . .	927
27.365.3.39kSetStringMacro . . . . .	927
27.365.3.40kSetStringMacro . . . . .	927
27.365.3.41kSetStringMacro . . . . .	927
27.365.3.42kSetStringMacro . . . . .	927
27.365.3.43kTypeRevisionMacro . . . . .	927
27.365.4 Member Data Documentation . . . . .	927
27.365.4.1Internals . . . . .	927
27.365.4.2ReferenceFrameOfReferenceUID . . . . .	927
27.365.4.3ReferenceSeriesInstanceUID . . . . .	928
27.365.4.4SeriesInstanceUID . . . . .	928
27.365.4.5SOPInstanceUID . . . . .	928
27.365.4.6StructureSetDate . . . . .	928
27.365.4.7StructureSetLabel . . . . .	928
27.365.4.8StructureSetName . . . . .	928
27.365.4.9StructureSetTime . . . . .	928
27.365.4.10StudyInstanceUID . . . . .	928
27.366 dcm::Waveform Class Reference . . . . .	928
27.366.1 Detailed Description . . . . .	928
27.366.2 Constructor & Destructor Documentation . . . . .	928
27.366.2.1 Waveform . . . . .	928
27.367 dcm::Writer Class Reference . . . . .	928
27.367.1 Detailed Description . . . . .	931
27.367.2 Constructor & Destructor Documentation . . . . .	932
27.367.2.1 Writer . . . . .	932
27.367.2.2 ~Writer . . . . .	932
27.367.3 Member Function Documentation . . . . .	932
27.367.3.1 CheckFileMetaInformationOff . . . . .	932
27.367.3.2 CheckFileMetaInformationOn . . . . .	932
27.367.3.3 GetFile . . . . .	932
27.367.3.4 GetStreamPtr . . . . .	932
27.367.3.5 SetCheckFileMetaInformation . . . . .	932

27.367.3.6SetFile . . . . .	932
27.367.3.7SetFileName . . . . .	933
27.367.3.8SetStream . . . . .	933
27.367.3.9SetWriteDataSetOnly . . . . .	933
27.367.3.10Write . . . . .	933
27.367.4Friends And Related Function Documentation . . . . .	933
27.367.4.1StreamImageWriter . . . . .	933
27.367.5Member Data Documentation . . . . .	933
27.367.5.1Ofstream . . . . .	933
27.367.5.2Stream . . . . .	933
27.368dcm::XMLDictReader Class Reference . . . . .	934
27.368.1Detailed Description . . . . .	935
27.368.2Constructor & Destructor Documentation . . . . .	935
27.368.2.1XMLDictReader . . . . .	935
27.368.2.2~XMLDictReader . . . . .	935
27.368.3Member Function Documentation . . . . .	935
27.368.3.1CharacterDataHandler . . . . .	935
27.368.3.2EndElement . . . . .	935
27.368.3.3GetDict . . . . .	935
27.368.3.4HandleDescription . . . . .	935
27.368.3.5HandleEntry . . . . .	935
27.368.3.6StartElement . . . . .	935
27.369dcm::XMLPrinter Class Reference . . . . .	935
27.369.1Member Enumeration Documentation . . . . .	937
27.369.1.1PrintStyles . . . . .	937
27.369.2Constructor & Destructor Documentation . . . . .	937
27.369.2.1XMLPrinter . . . . .	937
27.369.2.2~XMLPrinter . . . . .	937
27.369.3Member Function Documentation . . . . .	937
27.369.3.1GetPrintStyle . . . . .	937
27.369.3.2HandleBulkData . . . . .	937
27.369.3.3Print . . . . .	937
27.369.3.4PrintDataElement . . . . .	937
27.369.3.5PrintDataSet . . . . .	937
27.369.3.6PrintSQ . . . . .	937
27.369.3.7SetFile . . . . .	937
27.369.3.8SetStyle . . . . .	937

27.369.4Member Data Documentation . . . . .	937
27.369.4.1F . . . . .	937
27.369.4.2PrintStyle . . . . .	937
27.370gdcm::XMLPrivateDictReader Class Reference . . . . .	938
27.370.1Detailed Description . . . . .	939
27.370.2Constructor & Destructor Documentation . . . . .	939
27.370.2.1XMLPrivateDictReader . . . . .	939
27.370.2.2~XMLPrivateDictReader . . . . .	939
27.370.3Member Function Documentation . . . . .	939
27.370.3.1CharacterDataHandler . . . . .	939
27.370.3.2EndElement . . . . .	939
27.370.3.3GetPrivateDict . . . . .	939
27.370.3.4HandleDescription . . . . .	939
27.370.3.5HandleEntry . . . . .	939
27.370.3.6StartElement . . . . .	939
<b>28 File Documentation . . . . .</b>	<b>941</b>
28.1 gdcm2pnm.man File Reference . . . . .	941
28.2 gdcm2vtk.man File Reference . . . . .	941
28.3 gdcmAAbortPDU.h File Reference . . . . .	941
28.4 gdcmAAAssociateACPDU.h File Reference . . . . .	942
28.5 gdcmAAAssociateRJPDU.h File Reference . . . . .	942
28.6 gdcmAAAssociateRQPDU.h File Reference . . . . .	943
28.7 gdcmAbstractSyntax.h File Reference . . . . .	944
28.8 gdcmanon.man File Reference . . . . .	945
28.9 gdcmAnonymizeEvent.h File Reference . . . . .	945
28.10gdcmAnonymizer.h File Reference . . . . .	947
28.11gdcmApplicationContext.h File Reference . . . . .	947
28.12gdcmApplicationEntity.h File Reference . . . . .	948
28.13gdcmAReleaseRPPDU.h File Reference . . . . .	949
28.14gdcmAReleaseRQPDU.h File Reference . . . . .	950
28.15gdcmARTIMTimer.h File Reference . . . . .	951
28.16gdcmASN1.h File Reference . . . . .	952
28.17gdcmAsynchronousOperationsWindowSub.h File Reference . . . . .	953
28.18gdcmAttribute.h File Reference . . . . .	953
28.19gdcmAudioCodec.h File Reference . . . . .	955
28.20gdcmBase64.h File Reference . . . . .	955

28.21gdcmbaseCompositeMessage.h File Reference . . . . .	956
28.22gdcmbasePDU.h File Reference . . . . .	957
28.23gdcmbaseRootQuery.h File Reference . . . . .	958
28.24gdcmbasicOffsetTable.h File Reference . . . . .	960
28.25gdcmbitmap.h File Reference . . . . .	961
28.26gdcmbitmapToBitmapFilter.h File Reference . . . . .	962
28.27gdcmboundingBox.h File Reference . . . . .	963
28.28gdcmbuffer.h File Reference . . . . .	963
28.29gdcmbyteswap.h File Reference . . . . .	964
28.30gdcmbyteswapFilter.h File Reference . . . . .	965
28.31gdcmbytevalue.h File Reference . . . . .	966
28.32gdcmcapiCryptoFactory.h File Reference . . . . .	967
28.33gdcmcapiCryptographicMessageSyntax.h File Reference . . . . .	967
28.34gdcmcEchoMessages.h File Reference . . . . .	968
28.35gdcmcFindMessages.h File Reference . . . . .	969
28.36gdcmcMoveMessages.h File Reference . . . . .	970
28.37gdcmcCodec.h File Reference . . . . .	971
28.38gdcmcCoder.h File Reference . . . . .	972
28.39gdcmcCodeString.h File Reference . . . . .	973
28.40gdcmcCommand.h File Reference . . . . .	974
28.41gdcmcCommandDataSet.h File Reference . . . . .	976
28.42gdcmcCompositeMessageFactory.h File Reference . . . . .	976
28.43gdcmcCompositeNetworkFunctions.h File Reference . . . . .	977
28.44gdcmcConstCharWrapper.h File Reference . . . . .	978
28.45gdcmcconv.man File Reference . . . . .	978
28.46gdcmcCP246ExplicitDataElement.h File Reference . . . . .	979
28.47gdcmcCryptoFactory.h File Reference . . . . .	979
28.48gdcmcCryptographicMessageSyntax.h File Reference . . . . .	980
28.49gdcmcCSAElement.h File Reference . . . . .	981
28.50gdcmcCSAHeader.h File Reference . . . . .	982
28.51gdcmcCSAHeaderDict.h File Reference . . . . .	983
28.52gdcmcCSAHeaderDictEntry.h File Reference . . . . .	985
28.53gdcmcCStoreMessages.h File Reference . . . . .	986
28.54gdcmcCurve.h File Reference . . . . .	987
28.55gdcmcDataElement.h File Reference . . . . .	988
28.56gdcmcDataEvent.h File Reference . . . . .	990
28.57gdcmcDataSet.h File Reference . . . . .	991

28.58gdcmDataSetEvent.h File Reference . . . . .	992
28.59gdcmDataSetHelper.h File Reference . . . . .	992
28.60gdcmDecoder.h File Reference . . . . .	993
28.61gdcmDefinedTerms.h File Reference . . . . .	994
28.62gdcmDeflateStream.h File Reference . . . . .	995
28.63gdcmDefs.h File Reference . . . . .	995
28.64gdcmDeltaEncodingCodec.h File Reference . . . . .	997
28.65gdcmDICOMDIR.h File Reference . . . . .	997
28.66gdcmDICOMDIRGenerator.h File Reference . . . . .	998
28.67gdcmDict.h File Reference . . . . .	999
28.68gdcmDictConverter.h File Reference . . . . .	1001
28.69gdcmDictEntry.h File Reference . . . . .	1001
28.70gdcmDictPrinter.h File Reference . . . . .	1003
28.71gdcmDicts.h File Reference . . . . .	1003
28.72gdcmdiff.man File Reference . . . . .	1004
28.73gdcmDIMSE.h File Reference . . . . .	1005
28.74gdcmDirectionCosines.h File Reference . . . . .	1005
28.75gdcmDirectory.h File Reference . . . . .	1006
28.76gdcmDirectoryHelper.h File Reference . . . . .	1007
28.77gdcmDummyValueGenerator.h File Reference . . . . .	1008
28.78gdcmdump.man File Reference . . . . .	1008
28.79gdcmDumper.h File Reference . . . . .	1009
28.80gdcmElement.h File Reference . . . . .	1009
28.81gdcmEncapsulatedDocument.h File Reference . . . . .	1011
28.82gdcmEnumeratedValues.h File Reference . . . . .	1011
28.83gdcmEvent.h File Reference . . . . .	1012
28.83.1 Macro Definition Documentation . . . . .	1013
28.83.1.1 gdcmEventMacro . . . . .	1013
28.84gdcmException.h File Reference . . . . .	1014
28.85gdcmExplicitDataElement.h File Reference . . . . .	1015
28.86gdcmExplicitImplicitDataElement.h File Reference . . . . .	1015
28.87gdcmFiducials.h File Reference . . . . .	1016
28.88gdcmFile.h File Reference . . . . .	1017
28.89gdcmFileAnonymizer.h File Reference . . . . .	1018
28.90gdcmFileChangeTransferSyntax.h File Reference . . . . .	1018
28.91gdcmFileDerivation.h File Reference . . . . .	1019
28.92gdcmFileExplicitFilter.h File Reference . . . . .	1020

28.93gdcmlFileMetaInformation.h File Reference . . . . .	1021
28.94gdcmlFilename.h File Reference . . . . .	1022
28.95gdcmlFileNameEvent.h File Reference . . . . .	1022
28.96gdcmlFilenameGenerator.h File Reference . . . . .	1023
28.97gdcmlFileSet.h File Reference . . . . .	1024
28.98gdcmlFileStreamer.h File Reference . . . . .	1026
28.99gdcmlFindPatientRootQuery.h File Reference . . . . .	1026
28.100gdcmlFindStudyRootQuery.h File Reference . . . . .	1027
28.101gdcmlFragment.h File Reference . . . . .	1028
28.102gdcmlgendir.man File Reference . . . . .	1030
28.103gdcmlGlobal.h File Reference . . . . .	1030
28.104gdcmlGroupDict.h File Reference . . . . .	1031
28.105gdcmlIconImage.h File Reference . . . . .	1031
28.106gdcmlIconImageFilter.h File Reference . . . . .	1032
28.107gdcmlIconImageGenerator.h File Reference . . . . .	1033
28.108gdcmlImage.h File Reference . . . . .	1034
28.109gdcmlImageApplyLookupTable.h File Reference . . . . .	1035
28.110gdcmlImageChangePhotometricInterpretation.h File Reference . . . . .	1036
28.111gdcmlImageChangePlanarConfiguration.h File Reference . . . . .	1037
28.112gdcmlImageChangeTransferSyntax.h File Reference . . . . .	1038
28.113gdcmlImageCodec.h File Reference . . . . .	1039
28.114gdcmlImageConverter.h File Reference . . . . .	1040
28.115gdcmlImageFragmentSplitter.h File Reference . . . . .	1041
28.116gdcmlImageHelper.h File Reference . . . . .	1042
28.117gdcmlImageReader.h File Reference . . . . .	1043
28.118gdcmlImageRegionReader.h File Reference . . . . .	1044
28.119gdcmlImageToImageFilter.h File Reference . . . . .	1045
28.120gdcmlImageWriter.h File Reference . . . . .	1046
28.121gdcmlimg.man File Reference . . . . .	1047
28.122gdcmlImplementationClassUIDSub.h File Reference . . . . .	1047
28.123gdcmlImplementationUIDSub.h File Reference . . . . .	1048
28.124gdcmlImplementationVersionNameSub.h File Reference . . . . .	1049
28.125gdcmlImplicitDataElement.h File Reference . . . . .	1051
28.126gdcmlinfo.man File Reference . . . . .	1051
28.127gdcmlIOD.h File Reference . . . . .	1051
28.128gdcmlIODEntry.h File Reference . . . . .	1053
28.129gdcmlIODs.h File Reference . . . . .	1056

28.130dcmIPPSorter.h File Reference . . . . .	1057
28.131dcmItem.h File Reference . . . . .	1058
28.132dcmJPEG12Codec.h File Reference . . . . .	1060
28.133dcmJPEG16Codec.h File Reference . . . . .	1060
28.134dcmJPEG2000Codec.h File Reference . . . . .	1061
28.135dcmJPEG8Codec.h File Reference . . . . .	1062
28.136dcmJPEGCodec.h File Reference . . . . .	1063
28.137dcmJPEGLSCodec.h File Reference . . . . .	1064
28.138dcmJSON.h File Reference . . . . .	1065
28.139dcmKAKADUCodec.h File Reference . . . . .	1066
28.140dcmLegacyMacro.h File Reference . . . . .	1067
28.140.1Macro Definition Documentation . . . . .	1068
28.140.1.1GDCM_LEGACY . . . . .	1068
28.140.1.2GDCM_LEGACY_BODY . . . . .	1068
28.140.1.3GDCM_LEGACY_REPLACED_BODY . . . . .	1068
28.141dcmLO.h File Reference . . . . .	1068
28.142dcmLookupTable.h File Reference . . . . .	1069
28.143dcmMacro.h File Reference . . . . .	1070
28.144dcmMacroEntry.h File Reference . . . . .	1072
28.144.1Macro Definition Documentation . . . . .	1074
28.144.1.1GDCMMACROENTRY_H . . . . .	1074
28.145dcmMacros.h File Reference . . . . .	1074
28.146dcmMaximumLengthSub.h File Reference . . . . .	1076
28.147dcmMD5.h File Reference . . . . .	1077
28.148dcmMediaStorage.h File Reference . . . . .	1078
28.149dcmMeshPrimitive.h File Reference . . . . .	1079
28.150dcmModule.h File Reference . . . . .	1081
28.151dcmModuleEntry.h File Reference . . . . .	1082
28.152dcmModules.h File Reference . . . . .	1084
28.153dcmMovePatientRootQuery.h File Reference . . . . .	1085
28.154dcmMoveStudyRootQuery.h File Reference . . . . .	1086
28.155dcmNestedModuleEntries.h File Reference . . . . .	1087
28.156dcmNetworkEvents.h File Reference . . . . .	1089
28.157dcmNetworkStateID.h File Reference . . . . .	1090
28.158dcmObject.h File Reference . . . . .	1091
28.159dcmOpenSSLCryptoFactory.h File Reference . . . . .	1092
28.160dcmOpenSSLCryptographicMessageSyntax.h File Reference . . . . .	1092



28.164dcmOpenSSLP7CryptoFactory.h File Reference . . . . .	1093
28.162dcmOpenSSLP7CryptographicMessageSyntax.h File Reference . . . . .	1094
28.163dcmOrientation.h File Reference . . . . .	1096
28.164dcmOverlay.h File Reference . . . . .	1096
28.165dcmpap3.man File Reference . . . . .	1097
28.166dcmParseException.h File Reference . . . . .	1097
28.167dcmParser.h File Reference . . . . .	1099
28.168dcmPatient.h File Reference . . . . .	1099
28.169dcmPDataTFPDU.h File Reference . . . . .	1100
28.170dcmPDBElement.h File Reference . . . . .	1101
28.171dcmPDBHeader.h File Reference . . . . .	1103
28.172dcmpdf.man File Reference . . . . .	1103
28.173dcmPDFCodec.h File Reference . . . . .	1103
28.174dcmPDUFactory.h File Reference . . . . .	1104
28.175dcmPersonName.h File Reference . . . . .	1105
28.176dcmPGXCodec.h File Reference . . . . .	1106
28.177dcmPhotometricInterpretation.h File Reference . . . . .	1106
28.178dcmPixelFormat.h File Reference . . . . .	1107
28.179dcmPixmap.h File Reference . . . . .	1108
28.180dcmPixmapReader.h File Reference . . . . .	1109
28.181dcmPixmapToPixmapFilter.h File Reference . . . . .	1111
28.182dcmPixmapWriter.h File Reference . . . . .	1111
28.183dcmPNMCodec.h File Reference . . . . .	1112
28.184dcmPreamble.h File Reference . . . . .	1113
28.185dcmPresentationContext.h File Reference . . . . .	1114
28.186dcmPresentationContextAC.h File Reference . . . . .	1115
28.187dcmPresentationContextGenerator.h File Reference . . . . .	1117
28.188dcmPresentationContextRQ.h File Reference . . . . .	1117
28.189dcmPresentationDataValue.h File Reference . . . . .	1118
28.190dcmPrinter.h File Reference . . . . .	1119
28.191dcmPrivateTag.h File Reference . . . . .	1120
28.192dcmProgressEvent.h File Reference . . . . .	1122
28.193dcmPVRGCodec.h File Reference . . . . .	1122
28.194dcmPythonFilter.h File Reference . . . . .	1123
28.195dcmQueryBase.h File Reference . . . . .	1124
28.196dcmQueryFactory.h File Reference . . . . .	1126
28.197dcmQueryImage.h File Reference . . . . .	1127

28.190dcmQueryPatient.h File Reference . . . . .	1127
28.190dcmQuerySeries.h File Reference . . . . .	1128
28.200dcmQueryStudy.h File Reference . . . . .	1129
28.200dcmraw.man File Reference . . . . .	1130
28.200dcmRAWCodec.h File Reference . . . . .	1130
28.200dcmReader.h File Reference . . . . .	1131
28.200dcmRegion.h File Reference . . . . .	1133
28.200dcmRescaler.h File Reference . . . . .	1134
28.200dcmRLECodec.h File Reference . . . . .	1134
28.200dcmRoleSelectionSub.h File Reference . . . . .	1135
28.200dcmScanner.h File Reference . . . . .	1136
28.200dcmscanner.man File Reference . . . . .	1137
28.210dcmscu.man File Reference . . . . .	1137
28.210dcmSegment.h File Reference . . . . .	1137
28.210dcmSegmentedPaletteColorLookupTable.h File Reference . . . . .	1139
28.210dcmSegmentHelper.h File Reference . . . . .	1139
28.210dcmSegmentReader.h File Reference . . . . .	1141
28.210dcmSegmentWriter.h File Reference . . . . .	1142
28.210dcmSequenceOfFragments.h File Reference . . . . .	1143
28.210dcmSequenceOfItems.h File Reference . . . . .	1144
28.210dcmSerieHelper.h File Reference . . . . .	1144
28.210dcmSeries.h File Reference . . . . .	1146
28.220dcmServiceClassApplicationInformation.h File Reference . . . . .	1147
28.220dcmServiceClassUser.h File Reference . . . . .	1148
28.220dcmSHA1.h File Reference . . . . .	1148
28.220dcmSimpleSubjectWatcher.h File Reference . . . . .	1149
28.220dcmSmartPointer.h File Reference . . . . .	1150
28.220dcmSOPClassExtendedNegociationSub.h File Reference . . . . .	1151
28.220dcmSOPClassUIDToIOD.h File Reference . . . . .	1152
28.220dcmSorter.h File Reference . . . . .	1153
28.220dcmSpacing.h File Reference . . . . .	1155
28.220dcmSpectroscopy.h File Reference . . . . .	1155
28.230dcmSplitMosaicFilter.h File Reference . . . . .	1156
28.230dcmStaticAssert.h File Reference . . . . .	1157
28.231.1Macro Definition Documentation . . . . .	1157
28.231.1.1GDCM_DO_JOIN . . . . .	1157
28.231.1.2GDCM_DO_JOIN2 . . . . .	1157

28.231.1.3GDCM_JOIN . . . . .	1157
28.231.1.4GDCM_STATIC_ASSERT . . . . .	1158
28.232dcmStreamImageReader.h File Reference . . . . .	1158
28.233dcmStreamImageWriter.h File Reference . . . . .	1158
28.234dcmString.h File Reference . . . . .	1159
28.235dcmStringFilter.h File Reference . . . . .	1160
28.236dcmStudy.h File Reference . . . . .	1161
28.237dcmSubject.h File Reference . . . . .	1162
28.238dcmSurface.h File Reference . . . . .	1163
28.239dcmSurfaceHelper.h File Reference . . . . .	1164
28.240dcmSurfaceReader.h File Reference . . . . .	1165
28.241dcmSurfaceWriter.h File Reference . . . . .	1166
28.242dcmSwapCode.h File Reference . . . . .	1167
28.243dcmSwapper.h File Reference . . . . .	1168
28.244dcmSystem.h File Reference . . . . .	1169
28.245dcmTable.h File Reference . . . . .	1170
28.246dcmTableEntry.h File Reference . . . . .	1171
28.247dcmTableReader.h File Reference . . . . .	1172
28.248dcmTag.h File Reference . . . . .	1174
28.249dcmTagPath.h File Reference . . . . .	1175
28.250dcmTagToVR.h File Reference . . . . .	1175
28.251dcmTar.man File Reference . . . . .	1175
28.252dcmTerminal.h File Reference . . . . .	1176
28.253dcmTestDriver.h File Reference . . . . .	1177
28.254dcmTesting.h File Reference . . . . .	1177
28.255dcmTrace.h File Reference . . . . .	1178
28.255.1Macro Definition Documentation . . . . .	1179
28.255.1.1GDCM_FUNCTION . . . . .	1179
28.255.1.2dcmAssertAlwaysMacro . . . . .	1179
28.255.1.3dcmAssertMacro . . . . .	1179
28.255.1.4dcmDebugMacro . . . . .	1180
28.255.1.5dcmErrorMacro . . . . .	1180
28.255.1.6dcmWarningMacro . . . . .	1180
28.256dcmTransferSyntax.h File Reference . . . . .	1181
28.257dcmTransferSyntaxSub.h File Reference . . . . .	1182
28.258dcmType.h File Reference . . . . .	1183
28.259dcmTypes.h File Reference . . . . .	1185

28.260dcmUIDGenerator.h File Reference . . . . .	1185
28.261dcmUIDs.h File Reference . . . . .	1186
28.262dcmULAction.h File Reference . . . . .	1187
28.263dcmULActionAA.h File Reference . . . . .	1188
28.264dcmULActionAE.h File Reference . . . . .	1189
28.265dcmULActionAR.h File Reference . . . . .	1190
28.266dcmULActionDT.h File Reference . . . . .	1190
28.267dcmULBasicCallback.h File Reference . . . . .	1191
28.268dcmULConnection.h File Reference . . . . .	1192
28.269dcmULConnectionCallback.h File Reference . . . . .	1193
28.270dcmULConnectionInfo.h File Reference . . . . .	1194
28.271dcmULConnectionManager.h File Reference . . . . .	1195
28.272dcmULEvent.h File Reference . . . . .	1196
28.273dcmULTransitionTable.h File Reference . . . . .	1197
28.274dcmULWritingCallback.h File Reference . . . . .	1198
28.275dcmUNExplicitDataElement.h File Reference . . . . .	1198
28.276dcmUNExplicitImplicitDataElement.h File Reference . . . . .	1199
28.277dcmUnpacker12Bits.h File Reference . . . . .	1200
28.278dcmUsage.h File Reference . . . . .	1200
28.279dcmUserInformation.h File Reference . . . . .	1203
28.280dcmUUIDGenerator.h File Reference . . . . .	1204
28.281dcmValidate.h File Reference . . . . .	1204
28.282dcmValue.h File Reference . . . . .	1205
28.283dcmValueIO.h File Reference . . . . .	1206
28.284dcmVersion.h File Reference . . . . .	1207
28.285dcmviewer.man File Reference . . . . .	1208
28.286dcmVL.h File Reference . . . . .	1208
28.287dcmVM.h File Reference . . . . .	1209
28.287.1Macro Definition Documentation . . . . .	1210
28.287.1.1TYPETOLENGTH . . . . .	1210
28.288dcmVR.h File Reference . . . . .	1210
28.288.1Macro Definition Documentation . . . . .	1212
28.288.1.1TYPETOENCODING . . . . .	1212
28.288.1.2VRTemplateCase . . . . .	1212
28.289dcmVR16ExplicitDataElement.h File Reference . . . . .	1213
28.290dcmWaveform.h File Reference . . . . .	1213
28.291dcmWin32.h File Reference . . . . .	1214

28.291.1Macro Definition Documentation . . . . .	1214
28.291.1.1GDCM_EXPORT . . . . .	1214
28.292gdcmlWriter.h File Reference . . . . .	1214
28.293gdcml.xml.man File Reference . . . . .	1215
28.294gdcmlXMLDictReader.h File Reference . . . . .	1215
28.295gdcmlXMLPrinter.h File Reference . . . . .	1216
28.296gdcmlXMLPrivateDictReader.h File Reference . . . . .	1216
28.297README.txt File Reference . . . . .	1217
28.298testsList.txt File Reference . . . . .	1217
28.299tkGDCMImageReader.h File Reference . . . . .	1217
28.299.1Macro Definition Documentation . . . . .	1218
28.299.1.1VTK_CMYK . . . . .	1218
28.299.1.2VTK_INVERSE_LUMINANCE . . . . .	1218
28.299.1.3VTK_LOOKUP_TABLE . . . . .	1218
28.299.1.4VTK_YBR . . . . .	1218
28.300tkGDCMImageReader2.h File Reference . . . . .	1218
28.300.1Macro Definition Documentation . . . . .	1219
28.300.1.1VTK_CMYK . . . . .	1219
28.300.1.2VTK_INVERSE_LUMINANCE . . . . .	1219
28.300.1.3VTK_LOOKUP_TABLE . . . . .	1219
28.300.1.4VTK_YBR . . . . .	1219
28.301tkGDCMImageWriter.h File Reference . . . . .	1219
28.302tkGDCMMedicalImageProperties.h File Reference . . . . .	1220
28.303tkGDCMPolyDataReader.h File Reference . . . . .	1221
28.304tkGDCMPolyDataWriter.h File Reference . . . . .	1221
28.305tkGDCMTesting.h File Reference . . . . .	1222
28.306tkGDCMThreadedImageReader.h File Reference . . . . .	1222
28.307tkGDCMThreadedImageReader2.h File Reference . . . . .	1223
28.308tkImageColorViewer.h File Reference . . . . .	1224
28.309tkImageMapToColors16.h File Reference . . . . .	1224
28.310tkImageMapToWindowLevelColors2.h File Reference . . . . .	1225
28.311tkImagePlanarComponentsToComponents.h File Reference . . . . .	1225
28.312tkImageRGBToYBR.h File Reference . . . . .	1226
28.313tkImageYBRToRGB.h File Reference . . . . .	1226
28.314tkLookupTable16.h File Reference . . . . .	1227
28.315tkRTStructSetProperties.h File Reference . . . . .	1227

<b>29 Example Documentation</b>	<b>1229</b>
29.1 AWTMedical3.java	1229
29.2 BasicAnonymizer.cs	1233
29.3 BasicImageAnonymizer.cs	1234
29.4 CastConvertPhilips.py	1236
29.5 ChangeSequenceUltrasound.cxx	1238
29.6 CheckBigEndianBug.cxx	1239
29.7 ClinicalTrialAnnotate.cxx	1241
29.8 ClinicalTrialIdentificationWorkflow.cs	1242
29.9 CompressImage.cxx	1245
29.10CompressLossyJPEG.cs	1246
29.11Convert16BitsTo8Bits.cxx	1247
29.12ConvertMPL.py	1248
29.13ConvertMultiFrameToSingleFrame.cxx	1249
29.14ConvertNumpy.py	1250
29.15ConvertPIL.py	1251
29.16ConvertRGBToLuminance.cxx	1252
29.17ConvertSingleBitTo8Bits.cxx	1253
29.18ConvertToQImage.cxx	1254
29.19CreateARGBImage.cxx	1256
29.20CreateCMYKImage.cxx	1257
29.21CreateJPIPDataSet.cxx	1258
29.22CreateRAWStorage.py	1259
29.23csa2img.cxx	1261
29.24CStoreQtProgress.cxx	1263
29.25DecompressImage.cs	1266
29.26DecompressImage.java	1267
29.27DecompressImage.py	1268
29.28DecompressImageMultiframe.cs	1269
29.29DecompressJPEGFile.cs	1270
29.30DecompressPixmap.java	1272
29.31DiffFile.cxx	1272
29.32DiscriminateVolume.cxx	1273
29.33DumbAnonymizer.py	1277
29.34DumpADAC.cxx	1279
29.35DumpExamCard.cxx	1283
29.36DumpGEMSMovieGroup.cxx	1290

29.37DumpImageHeaderInfo.cxx . . . . .	1296
29.38DumpPhilipsECHO.cxx . . . . .	1299
29.39DumpToSQLITE3.cxx . . . . .	1304
29.40DuplicatePCDE.cxx . . . . .	1306
29.41ELSCINT1WaveToText.cxx . . . . .	1308
29.42EncapsulateFileInRawData.cxx . . . . .	1310
29.43ExtractEncapsulatedFile.cs . . . . .	1311
29.44ExtractEncryptedContent.cxx . . . . .	1312
29.45ExtractIconFromFile.cxx . . . . .	1313
29.46ExtractImageRegion.cs . . . . .	1315
29.47ExtractImageRegion.java . . . . .	1316
29.48ExtractImageRegionWithLUT.cs . . . . .	1317
29.49Extracting_All_Resolution.cxx . . . . .	1318
29.50ExtractOneFrame.cs . . . . .	1324
29.51Fake_Image_Using_Stream_Image_Writer.cxx . . . . .	1325
29.52FileAnonymize.cs . . . . .	1328
29.53FileAnonymize.java . . . . .	1329
29.54FileChangeTS.cs . . . . .	1330
29.55FileStreaming.cs . . . . .	1332
29.56FindAllPatientName.py . . . . .	1333
29.57FixBrokenJ2K.cxx . . . . .	1334
29.58FixCommaBug.py . . . . .	1336
29.59FixJAIBugJPEGLS.cxx . . . . .	1337
29.60gdcmmorthoplanes.cxx . . . . .	1339
29.61gdcmmreslice.cxx . . . . .	1346
29.62gdcmmrtionplan.cxx . . . . .	1348
29.63gdcmmrtplan.cxx . . . . .	1352
29.64gdcmmscene.cxx . . . . .	1356
29.65gdcmmtexture.cxx . . . . .	1358
29.66gdcmmvolume.cxx . . . . .	1359
29.67GenAllVR.cxx . . . . .	1361
29.68GenerateDICOMDIR.cs . . . . .	1363
29.69GenerateRTSTRUCT.cxx . . . . .	1364
29.70GenerateStandardSOPClasses.cxx . . . . .	1367
29.71GenFakeIdentifyFile.cxx . . . . .	1368
29.72GenFakeImage.cxx . . . . .	1370
29.73GenLongSeqs.cxx . . . . .	1372

29.74GenSeqs.cxx . . . . .	1373
29.75GetArray.cs . . . . .	1374
29.76GetJPEGSamplePrecision.cxx . . . . .	1376
29.77GetPortionCSAHeader.py . . . . .	1377
29.78GetSequenceUltrasound.cxx . . . . .	1378
29.79GetSubSequenceData.cxx . . . . .	1380
29.80headsq2dcm.py . . . . .	1383
29.81HelloActiviz.cs . . . . .	1383
29.82HelloActiviz2.cs . . . . .	1385
29.83HelloActiviz3.cs . . . . .	1386
29.84HelloActiviz4.cs . . . . .	1387
29.85HelloActiviz5.cs . . . . .	1387
29.86HelloSimple.java . . . . .	1389
29.87HelloVizWorld.cxx . . . . .	1389
29.88HelloVTKWorld.cs . . . . .	1390
29.89HelloVTKWorld.java . . . . .	1391
29.90HelloVTKWorld2.cs . . . . .	1392
29.91HelloWorld.cxx . . . . .	1393
29.92HelloWorld.py . . . . .	1394
29.93iU22tomultisc.cxx . . . . .	1395
29.94LargeVRDSExplicit.cxx . . . . .	1396
29.95MagnifyFile.cxx . . . . .	1398
29.96ManipulateFile.cs . . . . .	1400
29.97ManipulateFile.py . . . . .	1400
29.98ManipulateSequence.py . . . . .	1402
29.99MergeFile.py . . . . .	1403
29.100MergeTwoFiles.cxx . . . . .	1404
29.101MetalImageMD5Activiz.cs . . . . .	1405
29.102MIPViewer.java . . . . .	1406
29.103MpegVideoInfo.cs . . . . .	1409
29.104MPRViewer.java . . . . .	1413
29.105MPRViewer2.java . . . . .	1415
29.106MrProtocol.cxx . . . . .	1419
29.107NewSequence.cs . . . . .	1426
29.108NewSequence.py . . . . .	1427
29.109offscreenimage.cxx . . . . .	1428
29.110PatchFile.cxx . . . . .	1430



29.11 PhilipsPrivateRescaleInterceptSlope.py . . . . .	1431
29.11 PlaySound.py . . . . .	1432
29.11 pmsct_rgb1.cxx . . . . .	1433
29.11 PrivateDict.py . . . . .	1437
29.11 PublicDict.cxx . . . . .	1437
29.11 QIDO-RS.cxx . . . . .	1438
29.11 ReadAndDumpDICOMDIR.cxx . . . . .	1439
29.11 ReadAndDumpDICOMDIR.py . . . . .	1442
29.11 ReadAndPrintAttributes.cxx . . . . .	1444
29.12 ReadExplicitLengthSQIVR.cxx . . . . .	1446
29.12 ReadFiles.java . . . . .	1447
29.12 ReadGEMSSDO.cxx . . . . .	1448
29.12 ReadMultiTimesException.cxx . . . . .	1450
29.12 ReadSeriesIntoVTK.java . . . . .	1451
29.12 ReadUTF8QtDir.cxx . . . . .	1452
29.12 RefCounting.cs . . . . .	1454
29.12 ReformatFile.cs . . . . .	1454
29.12 RemovePrivateTags.py . . . . .	1456
29.12 RescaleImage.cs . . . . .	1456
29.13 Reslicesphere.cxx . . . . .	1457
29.13 RewriteSCAsMR.py . . . . .	1465
29.13 Re2img.cxx . . . . .	1466
29.13 structapp.cxx . . . . .	1469
29.13 ScanDirectory.cs . . . . .	1471
29.13 ScanDirectory.java . . . . .	1471
29.13 ScanDirectory.py . . . . .	1475
29.13 SendFileSCU.cs . . . . .	1476
29.13 SimplePrint.cs . . . . .	1476
29.13 SimplePrintPatientName.cs . . . . .	1478
29.14 SimpleScanner.cxx . . . . .	1478
29.14 SortImage.cxx . . . . .	1480
29.14 SortImage.py . . . . .	1482
29.14 SortImage2.cs . . . . .	1482
29.14 StandardizeFiles.cs . . . . .	1483
29.14 StreamImageReaderTest.cxx . . . . .	1484
29.14 TestByteSwap.cxx . . . . .	1488
29.14 TestReader.cxx . . . . .	1490

29.148	TestReader.py . . . . .	1491
29.149	hreadgdcmm.cxx . . . . .	1492
29.150	TraverseModules.cxx . . . . .	1495
29.151	uid_unique.cxx . . . . .	1496
29.152	VolumeSorter.cxx . . . . .	1497
29.153	WriteBuffer.py . . . . .	1500
<b>Index</b>		<b>1502</b>

## Chapter 1

# GDCM Documentation

This is the developpers documentation.

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

<http://gdcm.sourceforge.net/2.4/gdcm-2.4.2.pdf>

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

<http://gdcm.sourceforge.net/2.4/gdcm-2.4.2-doc.tar.gz>

### Author

Mathieu Malaterre



## Chapter 2

# off-screen rendering of DICOM images

### 2.1 SYNOPSIS

```
gdcm2pnm [options] file-in bitmap-out
```

### 2.2 DESCRIPTION

The **gdcm2pnm** command line program takes as input a DICOM file and produces a rendered bitmap file.

### 2.3 PARAMETERS

file-in    DICOM input filename

bitmap-out    Bitmap output filename

### 2.4 OPTIONS

#### 2.4.1 OPTIONS

#### 2.4.2 general options

```
-h    --help  
      print this help text and exit  
  
-v    --version  
      print version information and exit  
  
-V    --verbose  
      verbose mode (warning+error).  
  
-W    --warning  
      warning mode, print warning information
```

```
-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

## 2.5 Simple usage

**gdcm2pnm** will take as input DICOM and render it into a bitmap file using the window/level attributes value.

```
$ gdcm2pnm input.dcm output.png
```

It is much different from the **gdcmraw** or **gdcmimg** command line tool as it will render a DICOM image. This means that the output will be rendered in 8bits ready for display.

## 2.6 SEE ALSO

**gdcm2vtk(1)**, **gdcmimg(1)**

## 2.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 3

# Convert a file supported by VTK into DICOM.

### 3.1 SYNOPSIS

```
gdc2vtk [options] file-in file-out
```

### 3.2 DESCRIPTION

The **gdc2vtk** takes as input any file supported by VTK (including DICOM file) and will generate as output a DICOM file.

### 3.3 PARAMETERS

```
file-in    input filename (DICOM or VTK supported)

file-out    output filename (DICOM or VTK supported)
```

### 3.4 OPTIONS

#### 3.4.1 OPTIONS

--force-rescale	force rescale.
--force-spacing	force spacing.
--palette-color	when supported generate a PALETTE COLOR file.
--argb	when supported generate a ARGB file.
--compress	when supported generate a compressed file.
--use-vtkdicom	Use vtkDICOMImageReader (instead of GDCM).
--modality	set Modality.
--lower-left	set lower left.
--shift	set shift.
--scale	set scale.
--compress	set compression (MetaIO).
-T --study-uid	Study UID.
-S --series-uid	Series UID.
--root-uid	Root UID.

### 3.4.2 compression options

```
-J --jpeg          Compress image in jpeg.
-K --j2k          Compress image in j2k.
-L --jpegls       Compress image in jpeg-ls.
-R --rle          Compress image in rle (lossless only).
```

### 3.4.3 general options

```
-h  --help
    print this help text and exit

-v  --version
    print version information and exit

-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

### 3.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

## 3.5 DESCRIPTION

Convert a file supported by VTK into DICOM.

Typical usage is:

```
$ gdcmm2vtk inputfile output.dcm
```

It uses the internal factory mechanism of VTK to recognize a file (CanRead function). See VTK supported file here:

What image file formats can VTK read and write? [http://www.vtk.org/Wiki/VTK\\_FAQ#What\\_image\\_file\\_formats\\_can\\_VTK](http://www.vtk.org/Wiki/VTK_FAQ#What_image_file_formats_can_VTK)

If your input file has 4 components, the 4th comp (alpha) will be removed from the output file as DICOM does not support alpha component anymore (see `--argb` option).

Special care was taken for the following file format:

1. DICOM: Direction Cosines and `vtkMedicalImageInformation` are passed to the output
2. BMP: The file can be saved with a Lookup Table (see `--palette-color`)
3. GE Signa: `vtkMedicalImageProperties` is passed to the output
4. MINC: Direction Cosines is passed to the output
5. TIFF: `vtkTIFFReader` is currently in bad shape in VTK (different behavior in VTK 5.2 and `git/master`). Only u

### 3.5.1 CONVERT MetaImage (mhd, mha)

```
$ gdcmm2vtk inputfile output.mha
```



This command will convert the input DICOM file: inputfile into a MetaImage .mha file. Same goes for .mhd file.

### 3.5.2 CONVERT MHA/MHD

```
$ gdc2vtk inputfile output.mha
```

or

```
$ gdc2vtk inputfile output.mhd
```

This command will convert the input DICOM file: inputfile into a MetaImageData .mha/.mhd file.

### 3.5.3 CONVERT VTI

```
$ gdc2vtk inputfile output.vti
```

This command will convert the input DICOM file: inputfile into a XML VTK ImageData .vti file.

### 3.5.4 CONVERT VTK

```
$ gdc2vtk inputfile output.vtk
```

This command will convert the input DICOM file: inputfile into an old VTK Structured PointSets .vtk file.

## 3.6 CONVERT DICOM

```
$ gdc2vtk input.dcm output.dcm
```

[vtkGDCMImageReader](#) will be used to read in a DICOM file, not the default `vtkDICOMImageReader`. See option `--use-vtkdicom` to use `vtkDICOMImageReader`.

## 3.7 RoundTrip DICOM to MHD to DICOM

```
$ gdc2vtk input_ybr.dcm output.mhd
$ gdc2vtk --modality US --imageformat 7 output.mhd output.dcm
```

The above section shows how to convert a DICOM using the Photometric Interpretation of YBR\_FULL (or even YBR\_FULL\_422 is lossy) into another file format: MetaImage (mhd). Since this file format does not handle color space, we have to explicitly set it using the `--imageformat` command line option. The `--modality` command line option is required in this case since the default Secondary Capture Image Storage Class family does not allow for YBR Photometric Interpretation.

## 3.8 gdc2vtk notes

IMPORTANT NOTE: The internal VTK structured will be filled from the input DICOM, and then pass to the output DICOM writer. Some information might be lost during the conversion DICOM to VTK to DICOM. This option is mostly used to test the `vtkGDCMImageReader/vtkGDCMImageWriter` combination.

IMPORTANT NOTE: When converting from a lossy format such as JPEG, the information of lossiness is important. The output DICOM will contains the required Lossy Image Compression attribute that indicates that image was lossy-compressed somewhere along the pipeline. See also `gdcmimg` (better handling of JPEG in general).

IMPORTANT NOTE: When using `-use-vtkdicom` the output DICOM file will always be written as MR Image Storage as this information is not available from the reader itself. This allow setting the Image Orientation (Patient) properly.

### 3.9 SEE ALSO

`gdcmdump(1)`, `gdcmviewer(1)`, `gdcmimg(1)`

### 3.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 4

# Tool to anonymize a DICOM file.

### 4.1 SYNOPSIS

```
gdcmanon [options] file-in file-out
gdcmanon [options] dir-in  dir-out
```

### 4.2 DESCRIPTION

The **gdcmanon** tool is an implementation of PS 3.15 / E.1 / Basic Application Level Confidentiality Profile (Implementation of E.1.1 De-identify & E.1.2 Re-identify)

This tool is split into two very different operating mode:

- An implementation of PS 3.15, see -e and -d flags
- A dumb mode, see -dumb

Dumb mode and PS 3.15 do not work well together, you should really only use one type of anonymization. In case of doubt, avoid using -dumb.

In order to use the PS 3.15 implementation (-d & -e flag), you'll need a certificate to do de-identification operations, and the associated private key to do the re-identification operation. If you are only doing a one-shot anonymization and do not need to properly re-identify the DICOM file, you can safely discard the private key and only keep the certificate. See OpenSSL section below for an example on how to generate the private key/certificate pair.

gdcmanon will exit early if OpenSSL was not configured/build properly into the library (see GDCM\_USE\_SYSTEM\_OPENSSL in cmake).

### 4.3 PARAMETERS

```
file-in  DICOM input filename
```

```
file-out DICOM output filename
```

or

```
file-in  DICOM input directory
```

```
file-out DICOM output directory
```

## 4.4 OPTIONS

You need to specify at least one operating mode, from the following list (and only one):

### 4.4.1 Required parameters

-e --de-identify	De-identify DICOM (default)
-d --re-identify	Re-identify DICOM
--dumb	Dumb mode anonymizer

Warning when operating in dumb mode, you need to also specify an operation to do, such as 'remove' or 'empty' a tag, see below the dumb mode options.

### 4.4.2 OPTIONS

-i --input	DICOM filename / directory
-o --output	DICOM filename / directory
-r --recursive	recursively process (sub-)directories.
--continue	Do not stop when file found is not DICOM.
--root-uid	Root UID.
--resources-path	Resources path.
-k --key	Path to RSA Private Key.
-c --certificate	Path to Certificate.

### 4.4.3 encryption options

--des	DES.
--des3	Triple DES.
--aes128	AES 128.
--aes192	AES 192.
--aes256	AES 256.

### 4.4.4 dumb mode options

--empty %d,%d	DICOM tag(s) to empty
--remove %d,%d	DICOM tag(s) to remove
--replace %d,%d,%s	DICOM tag(s) to replace

### 4.4.5 general options

-h --help	print this help text and exit
-v --version	print version information and exit
-V --verbose	verbose mode (warning+error).
-W --warning	warning mode, print warning information
-E --error	error mode, print error information
-D --debug	debug mode, print debug information

#### 4.4.6 environment variable

```
GDCM_ROOT_UID Root UID
GDCM_RESOURCES_PATH path pointing to resources files (Part3.xml, ...)
```

### 4.5 Typical usage

#### 4.5.1 De-identification (anonymization, encrypt)

The only thing required for this operation is a certificate file (in PEM format).

```
$ gdcmanon --certificate certificate.pem -e original.dcm original_anonymized.dcm
```

You can use `--asn1` option from `gdcmdump` to dump the generated DataSet as ASN1 structure (see `gdcmdump(1)` for example).

#### 4.5.2 Re-identification (de-anonymization, decrypt)

The only thing required for this operation is a private key (in PEM format). It is required that the private key used for the re-identification process, was the actual private key used to generate the certificate file (certificate.pem) used during the de-identification step.

```
$ gdcmanon --key privatekey.pem -d original_anonymized.dcm original_copy.dcm
```

You can then check that `original.dcm` and `original_copy.dcm` are identical.

#### 4.5.3 Multiple files caveat

It is very important to understand the following section, when anonymizing more than one single file. When anonymizing multiple DICOM files, you are required to use the directory input. You cannot call multiple time the `gdcmanon` command line tool. Indeed the tool stores in memory during the process only a hash table of conversion so that each time a particular value is found it get always replaced by the same de-identified value (think: consistent Series Instance UID).

#### 4.5.4 Dumb mode

This functionality is not described in the DICOM standard. Users are advised that improper use of that mode is not recommended, meaning that important tag can be emptied/removed/replaced resulting in illegal/invalid DICOM file. Only use when you know what you are doing. If you delete a Type 1 attribute, chance is that your DICOM file will be not accepted in most DICOM third party viewer. Unfortunately this is often this mode that is implemented in popular DICOM Viewer, always prefer what the DICOM standard describes, and avoid the dumb mode.

The following example shows how to use dumb mode and achieve 5 operations at the same time:

- Empty the tag (0010,0010) Patient's Name,
- Empty the tag (0010,0020) Patient ID,
- Remove the tag (0010,0040) Patient's Sex
- Remove the tag (0010,1010) Patient's Age

- Replace the tag (0010,1030) Patient's Weight with the value '10'

You are required to check which DICOM attribute is Type 1 and Type 1C, before trying to '**Empty**' or '**Remove**' a particular DICOM attribute. For the same reason, you are required to check what are valid value in a replace operation.

```
$ gdcmanon --dumb --empty 10,10 --empty 10,20 --remove 10,40 --remove 10,1010 --replace 10,1030,10 012345.002.050
```

Multiple operation of `--dumb` mode can take place, just reuse the output of the previous operation. Always use `gdcmdump` on the input and output file to check what was actually achieved. You can use a diff program to check only what changed (see `gdcmdiff(1)` for example).

#### 4.5.4.1 Irreversible Anonymization

In some very rare cases, one would want to anonymize using the PS 3.15 mode so as to take benefit of the automatic conversion of all content that could contain Patient related information.

In the end all Patient related information has been removed and has been secretly stored in the 0400,0500 DICOM attribute. However to make sure that no-one ever try to break that security using brute-force algorithm, one want want to remove completely this DICOM attribute. This will make the DICOM:

- Completely free of any Patient related information (as per PS 3.15 specification)
- Remove any mean of people to brute force attack the file to find out the identity of the Patient

In this case one could simply do, as a first step execute the reversible anonymizer:

```
$ gdcmanon -c certificate.pem input.dcm anonymized_reversible.dcm
```

and now completely remove the DICOM attribute containing the secretly encrypted Patient related information:

```
$ gdcmanon --dumb --remove 400,500 --remove 12,62 --remove 12,63 anonymized_reversible.dcm anonymized_irreversible.dcm
```

#### Remarks

As mentionned in DICOM Sup 142, this anonymization is preferred over de-identification since: It is not required that the Encrypted Attributes Data Set be created; indeed, there may be circumstances where the Dataset is expected to be archived long enough that any contemporary encryption technology may be inadequate to provide long term protection against unauthorized recovery of identification

## 4.6 OpenSSL

On most system you can have access to OpenSSL to generate the Private Key/Certificate pair.

### 4.6.1 Generating a Private Key

Command line to generate a rsa key (512bit)

```
$ openssl genrsa -out CA_key.pem
```

Command line to generate a rsa key (2048bit)

```
$ openssl genrsa -out CA_key.pem 2048
```

Command line to generate a rsa key (2048bit) + passphrase

```
$ openssl genrsa -des3 -out CA_key.pem 2048
```

#### 4.6.2 Generating a Certificate

From your previously generated Private Key, you can now generate a certificate in PEM (DER format is currently not supported).

```
$ openssl req -new -key CA_key.pem -x509 -days 365 -out CA_cert.cer
```

## 4.7 DICOM Standard:

Page to the DICOM Standard:

<http://dicom.nema.org/>

The DICOM Standard at the time of releasing gdcmanon is:

<ftp://medical.nema.org/medical/dicom/2008/>

Direct link to PS 3.15-2008:

[ftp://medical.nema.org/medical/dicom/2008/08\\_15pu.pdf](ftp://medical.nema.org/medical/dicom/2008/08_15pu.pdf)

## 4.8 Warnings

Certain attributes may still contains Protected Health Information (PHI) after an anonymization step. This is typically the case for Patient's Address (0010,1040). The reason is that this particular attribute is not supposed to be in the composite IODs in the first place. DICOM Supp 142 includes it (however gdcmanon does not implement it).

## 4.9 SEE ALSO

**gdcconv(1), gdcmdump(1), gdcmdiff(1), openssl(1), dumpasn1(1)**

## 4.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre





## Chapter 5

# Tool to convert DICOM to DICOM.

### 5.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

### 5.2 DESCRIPTION

The **gdcmconv** command line program takes as input a DICOM file (file-in) and process it to generate an output DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

### 5.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out    DICOM output filename
```

### 5.4 OPTIONS

#### 5.4.1 PARAMETERS

```
-i --input      DICOM filename
-o --output      DICOM filename
```

#### 5.4.2 OPTIONS

```
-X --explicit      Change Transfer Syntax to explicit.
-M --implicit      Change Transfer Syntax to implicit.
-U --use-dict       Use dict for VR (only public by default).
  --with-private-dict Use private dict for VR (advanced user only).
-C --check-meta     Check File Meta Information (advanced user only).
  --root-uid        Root UID.
  --remove-gl       Remove group length (deprecated in DICOM 2008).
  --remove-private-tags Remove private tags.
  --remove-retired   Remove retired tags.
```

### 5.4.3 image options

```
-l --apply-lut           Apply LUT (non-standard, advanced user only).
-P --photometric-interpretation %s Change Photometric Interpretation (when possible).
-w --raw                Decompress image.
-d --deflated            Compress using deflated (gzip).
-J --jpeg               Compress image in jpeg.
-K --j2k                Compress image in j2k.
-L --jpegls             Compress image in jpeg-ls.
-R --rle                Compress image in rle (lossless only).
-F --force              Force decompression/merging before recompression/splitting.
  --generate-icon        Generate icon.
  --icon-minmax %d,%d    Min/Max value for icon.
  --icon-auto-minmax     Automatically compute best Min/Max values for icon.
  --compress-icon        Decide whether icon follows main TransferSyntax or remains uncompressed.
  --planar-configuration [01] Change planar configuration.
-Y --lossy              Use the lossy (if possible) compressor.
-S --split %d           Write 2D image with multiple fragments (using max size)
```

### 5.4.4 JPEG options

```
-q --quality %*f        set quality.
```

### 5.4.5 JPEG-LS options

```
-e --lossy-error %*i    set error.
```

### 5.4.6 J2K options

```
-r --rate %*f           set rate.
-q --quality %*f        set quality.
-t --tile %d,%d         set tile size.
-n --number-resolution %d set number of resolution.
  --irreversible         set irreversible.
```

### 5.4.7 general options

```
-h --help               print this help text and exit
-v --version            print version information and exit
-V --verbose            verbose mode (warning+error).
-W --warning            warning mode, print warning information
-E --error              error mode, print error information
-D --debug              debug mode, print debug information
```

### 5.4.8 special options

```
-I --ignore-errors      convert even if file is corrupted (advanced users only, see disclaimers).
```

### 5.4.9 environment variable

```
GDCM_ROOT_UID Root UID
```

## 5.5 Simple usage

**gdcmmconv** is a great tool to convert broken DICOM implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmmconv input.dcm output.dcm
```

or if you prefer being explicit:

```
$ gdcmmconv -i input.dcm -o output.dcm
```

Even though **gdcmmconv** can overwrite directly on the same file (`input.dcm = output.dcm`), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

Typical cases where you would want to use **gdcmmconv** in its simple form:

- convert non-cp246 conforming file into conforming cp246,
- convert implicit little endian transfer syntax file meta header into proper explicit little endian transfer syntax,
- convert the GE-13 bytes bug,
- convert dual syntax file: implicit/explicit,
- convert Philips dual Little Endian/Big Endian file,
- convert GDCM 1.2.0 broken UN-2-bytes fields,
- &...
- All other broken files listed in the supported section.

When no option other is used, only the dataset is inspected. So encapsulated Pixel Data, for instance, is not inspected for well known bugs.

When doing this kind of work, this is usually a good idea to perform some kind of quality control, see **gdcmmconv** Quality Control section (down below).

## 5.6 Typical usage

### 5.6.1 File Meta Header

Running

```
$ gdcmmconv input.dcm output.dcm
```

Is not enough to recompute file meta header, when input file is buggy. You may want to use: `--check-meta`

```
$ gdcmmconv --check-meta input.dcm output.dcm
```

See typical cases such as: `GE_DLX-8-MONO2-PrivateSyntax.dcm` or `PICKER-16-MONO2-No_DicomV3_Preamble.dcm` from `gdcmmData`.

### 5.6.2 Conversion to Explicit Transfer Syntax

To convert a file that was written using Implicit Transfer Syntax into Explicit Transfer Syntax simply use:

```
$ gdcmmconv --explicit uncompressed.dcm compressed.dcm
```

### 5.6.3 Compressing to lossless JPEG

To compress an uncompressed DICOM file to a JPEG Lossless encapsulated format:

```
$ gdcmmconv --jpeg uncompressed.dcm compressed.dcm
```

### 5.6.4 Compressing to lossy JPEG

To compress an uncompressed DICOM file to a JPEG Lossy encapsulated format:

```
$ gdcmmconv --lossy --jpeg -q 90 uncompressed.dcm compressed.dcm
```

Note:

`-q` is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify

### 5.6.5 Compressing to lossless JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossless encapsulated format:

```
$ gdcmmconv --jpegls uncompressed.dcm compressed.dcm
```

### 5.6.6 Compressing to lossy JPEG-LS

To compress an uncompressed DICOM file to a JPEG-LS Lossy encapsulated format:

```
$ gdcmmconv --lossy --jpegls -e 2 uncompressed.dcm lossy_compressed.dcm
```

Note:

`-e` (or `--lossy-error`) means that the maximum tolerate error is 2 for each pixel value

### 5.6.7 Compressing to lossless J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossless encapsulated format:

```
$ gdcmmconv --j2k uncompressed.dcm compressed.dcm
```

### 5.6.8 Compressing to lossy J2K

To compress an uncompressed DICOM file to a JPEG-2000 Lossy encapsulated format:

```
$ gdcmmconv --lossy -q 55,50,45 --j2k uncompressed.dcm lossy_compressed.dcm
```

Note:

`-q` is just one of the many way to specify lossy quality, you need to inspect the other cmd line flag to specify

### 5.6.9 Compressing to lossless RLE

To compress an uncompressed DICOM file to a RLE Lossless encapsulated format:

```
$ gdcconv --rle uncompressed.dcm compressed.dcm
```

There is no such thing as lossy RLE compression.

### 5.6.10 Split encapsulated DICOM:

To split an encapsulated stream into smaller chunk (1024 bytes each):

```
$ gdcconv --split 1024 rle.dcm rle_1024.dcm
```

If an odd number of bytes is passed it will be rounded down to the next even number (eg. 1025 -> 1024) since DICOM only allow even number for Value Length.

### 5.6.11 Forcing (re)compression

Sometime it is necessary to use the `-force` option. By default when user specify `-j2k` and input file is already in JPEG 2000 encapsulated DICOM format then no operation takes places. By using `-force` you make sure that (re)compression operation takes places.

Real life example of why you would use `-force`:

- When Pixel Data is missing data / is padded with junk
- When you would like to make sure GDCM can handle decompression & recompression cycle

### 5.6.12 Decompressing a Compressed DICOM

```
$ gdcconv --raw compressed.dcm uncompressed.dcm
```

### 5.6.13 Compressing an uncompressed Icon

By default when compressing a DICOM Image file, `gdcconv` will not compress the icon. A user option needs to be turned on to explicitly force the compression of the Icon Image Sequence Pixel Data

For example, by default we will not compress the Icon Image Sequence Pixel Data attribute:

```
$ gdcconv --jpeg gdcData/simpleImageWithIcon.dcm uncompressed_icon.dcm
```

In the following example we will explicitly compress the Icon Image Sequence Pixel Data attribute. In that case the same Transfer Syntax is being used for both the main Pixel Data and the Pixel Data from the Icon Image Sequence:

```
$ gdcconv --jpeg --compress-icon gdcData/simpleImageWithIcon.dcm compressed_icon.dcm
```

### 5.6.14 Generating an Icon

For some application it might be necessary to produce a small preview of the main image to be able to quickly load that short preview instead of the main image. In that case:

```
$ gdcconv --raw --generate-icon gdcData/test.acr test_icon.dcm
```

In some cases the main Pixel Data element is expressed as pixel defined on 16bits. Since Icon can only store at most pixel of size 8bits, a rescale operation needs to take place. In order to properly select a better interval for doing the rescale operation user can specify the min max used for the rescale operation:

```
$ gdcconv --raw --generate-icon --icon-minmax 0,192 gdcData/012345.002.050.dcm icon_minmax.dcm
```

### 5.6.15 Changing the planar Configuration

Often RLE files are compressed using a different Planar Configuration (RRR ... GGG... BBB...) instead of the usual triplet (RGB ... RGB ... RGB ). So upon decompression the Planar Configuration is 1. This is perfectly legal in DICOM, however this is unconventional, and thus it may be a good idea to also change the planar configuration and set it to the default :

```
$ gdcconv --raw --planar-configuration 0 compressed.dcm uncompressed1.dcm
```

To reinvert the planar configuration of file 'uncompressed1.dcm', simply do:

```
$ gdcconv --raw --planar-configuration 1 uncompressed1.dcm uncompressed2.dcm
```

## 5.7 Lossless Conversion

When talking about lossless conversion, there is an ambiguity that need to be understood. To achieve higher compression ratio, the RGB color space is usually not used, in favor of a YBR one. Changing from one color space to the other is (bit level) not lossless.

For more detail, see what are the true lossless transformations as described:

[http://gdc.sourceforge.net/wiki/index.php/Color\\_Space\\_Transformations](http://gdc.sourceforge.net/wiki/index.php/Color_Space_Transformations)

## 5.8 Quality Control

One important part when using gdcconv it to have a way to quality control the output.

You can use 3rd party tool to check the output of gdcconv is correct.

### 5.8.1 DCMTK / dicom3tools

Using another DICOM implementation such as the one from DCMTK or dicom3tools can be a good process to check the output of gdcconv.

- For DCMTK use: dcmdump
- For dicom3tools use: dcdump

For reference, gdcconv --raw will act as dcmdjpeg +cn +px, since it never tries to convert color space.

### 5.8.2 VIM: vimdiff

You can setup your favorite editor to compare the output, for instance in vim:

```
autocmd BufReadPre *.dcm set ro
autocmd BufReadPost *.dcm silent %!dcmdump -M +uc "%"
```

then simply do:

```
$ vimdiff input.dcm output.dcm
```

### 5.8.3 vbindiff

On UNIX you can visually compare binary file using the vbindiff command:

```
$ vbindiff input.dcm output.dcm
```

## 5.9 SEE ALSO

**gdcmdump(1)**, **gdcmraw(1)**, **gdcminfo(1)**, **gdcmdiff(1)**

## 5.10 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre





## Chapter 6

# dumps differences of two DICOM files

### 6.1 SYNOPSIS

```
gdcmdiff [options] file1 file2
```

### 6.2 DESCRIPTION

The **gdcmdiff** command line program takes as input two DICOM files: file1 and file2.

### 6.3 PARAMETERS

file1    DICOM input filename

file2    DICOM output filename

### 6.4 OPTIONS

#### 6.4.1 OPTIONS

```
-m        --meta            Compare metainformation. Default is off.  
-t <n>    --truncate <n>   String values trimmed to n characters.
```

#### 6.4.2 general options

```
-h        --help  
          print this help text and exit  
  
-v        --version  
          print version information and exit  
  
-V        --verbose  
          verbose mode (warning+error).  
  
-W        --warning  
          warning mode, print warning information
```

```
-E  --error  
    error mode, print error information  
  
-D  --debug  
    debug mode, print debug information
```

## 6.5 Simple usage

**gdcmdiff** is a great tool to produce a diff in between two DICOM files. Usage is simply:

```
$ gdcmdiff input1.dcm input2.dcm
```

## 6.6 SEE ALSO

**gdcmdump(1)**, **gdcminfo(1)**

## 6.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 7

**dumps a DICOM file, it will display the structure and values contained in the specified DICOM file.**

### 7.1 SYNOPSIS

```
gdcmdump [options] dcm_file
gdcmdump [options] dcm_directory
```

### 7.2 DESCRIPTION

The **gdcmdump** command line program dumps a DICOM file to the console. For those familiar with dcmdump (DCMTK) output, gdcmdump has some minor differences. Namely:

- For Implicit Transfer Syntax gdcmdump will print ?? instead of the dictionary VR

gdcmdump has a limited private dictionary that is used to lookup private element whenever possible.

### 7.3 PARAMETERS

```
dcm_file          DICOM input filename
dcm_directory     DICOM input directory
```

### 7.4 OPTIONS

#### 7.4.1 OPTIONS

-x --xml-dict	generate the XML dict (only private elements for now).
-r --recursive	recursive (input is a directory)
-d --dump	dump value (limited use).
-p --print	print value instead of simply dumping (default).
-c --color	print in color.
-C --csa	print SIEMENS CSA Header (0029,[12]0,SIEMENS CSA HEADER).
-P --pdb	print GEMS Protocol Data Block (0025,1b,GEMS_SERS_01).
--elscint	print ELSCINT Protocol Information (01f7,26,ELSCINT1).
--vepro	print VEPRO Protocol Information (0055,20,VEPRO VIF 3.0 DATA).

```

                or VEPRO Protocol Information (0055,20,VEPRO VIM 5.0 DATA).
--sds           print Philips MR Series Data Storage (1.3.46.670589.11.0.0.12.2) Information (2005,32,Philips)
-A --asn1       print encapsulated ASN1 structure >(0400,0520).
--map-uid-names map UID to names.

```

## 7.4.2 general options

```

-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information

```

## 7.4.3 special options

```

-I --ignore-errors  dumps even if file is corrupted (advanced users only, see disclaimers).

```

## 7.5 Typical usage

### 7.5.1 Printing Implicit Transfer Syntax

The VR are not found in the file, thus are presented with a "(??)", and right next to it (if found) the correct VR.

Eg.:

```
$ gdcmdump GE_DLX-8-MONO2-PrivateSyntax.dcm
```

```

# Dicom-File-Format
\&...
(0008,0000) ?? (UL) 434                                # 4,1 Generic Group Length
(0008,0005) ?? (CS) [ISO_IR 100]                        # 10,1-n Specific Character Set
(0008,0008) ?? (CS) [ORIGINAL\\PRIMARY\\SINGLE PLANE ]   # 30,2-n Image Type
(0008,0016) ?? (UI) [1.2.840.10008.5.1.4.1.1.12.1]      # 28,1 SOP Class UID
(0008,0018) ?? (UI) [1.2.840.113619.2.16.1.0.906539207.1.24207] # 42,1 SOP Instance UID
(0008,0020) ?? (DA) [19980923]                          # 8,1 Study Date
(0008,0021) ?? (DA) [19980923]                          # 8,1 Series Date
(0008,0022) ?? (DA) [19980923]                          # 8,1 Acquisition Date
(0008,0023) ?? (DA) [19980923]                          # 8,1 Content Date
(0008,0030) ?? (TM) [101229.000]                        # 10,1 Study Time
(0008,0031) ?? (TM) [101229.000]                        # 10,1 Series Time
(0008,0032) ?? (TM) [102653.000]                        # 10,1 Acquisition Time
(0008,0033) ?? (TM) [102653.000]                        # 10,1 Content Time
\&...

```

### 7.5.2 Print Private Attributes

GDCM has a limited private dictionary. Whenever possible, it will try to lookup the private data element.

```
$ gdcmdump 012345.002.050.dcm
```

```
\&...
(0009,0010) LO [GEMS_IDEN_01] # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ] # 14,1 Full fidelity
(0009,1002) SH [MRCV] # 4,1 Suite id
(0009,1004) SH [SIGNA ] # 6,1 Product id
(0009,1027) SL 985968524 # 4,1 Image actual date
(0009,1030) SH [19356UMR2 ] # 10,1 Service id
(0009,1031) SH [999 ] # 4,1 Mobile location number
(0009,10e3) UI [1.2.840.113619.1.1.4.1762386977] # 32,1 Equipment UID
(0009,10e6) SH [08] # 2,1 Genesis Version - now
(0009,10e7) UL 2757786872 # 4,1 Exam Record checksum
(0009,10e9) SL 985968523 # 4,1 Actual series data time stamp
\&...
(0019,0000) UL 1208 # 4,1 Generic Group Length
(0019,0010) LO [GEMS_ACQU_01] # 12,1 Private Creator
(0019,100f) DS [424.399994] # 10,1 Horiz. Frame of ref.
(0019,1011) SS 0 # 2,1 Series contrast
\&...
(0019,10e0) DS [0.000000] # 8,1 User data 24 {# DTI Diffusion Dir., relea
(0019,10e2) DS [0.000000] # 8,1 Velocity Encode Scale
(0019,10f2) SS 0 # 2,1 Fast phases
(0019,10f9) DS [98] # 2,1 Transmit gain
\&...
(0021,0000) UL 372 # 4,1 Generic Group Length
(0021,0010) LO [GEMS_RELA_01] # 12,1 Private Creator
(0021,1003) SS 0 # 2,1 Series from which Prescribed
\&...
```

### 7.5.3 SIEMENS CSA Header

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical SIEMENS MR DICOM file.

Eg.:

```
$ gdcmdump --csa MR_SIEMENS_forceLoad29-1010_29-1020.dcm
```

```
(0029,0010)siemens csa header
Image shadow data (0029,xx10)

0 - 'EchoLinePosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
1 - 'EchoColumnPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '64      '
2 - 'EchoPartitionPosition' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '32      '
3 - 'UsedChannelMask' VM 1, VR UL, SyngoDT 9, NoOfItems 6, Data '255      '
4 - 'Actual3DImaPartNumber' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
5 - 'ICE_Dims' VM 1, VR LO, SyngoDT 19, NoOfItems 6, Data 'X_1_1_1_1_1_31_1_1_1_1_19'
6 - 'B_value' VM 1, VR IS, SyngoDT 6, NoOfItems 6, Data '0      '
7 - 'Filter1' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
8 - 'Filter2' VM 1, VR IS, SyngoDT 6, NoOfItems 0, Data
\&...
```

### 7.5.4 GEMS Protocol Data Block

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical GEMS MR DICOM file.

Protocol Data Block : 0025,xx1b,GEMS\_SERS\_01

```
$ gdcmdump --pdb GE_MR_0025xx1bProtocolDataBlock.dcm
```

```
ENTRY "Head First"
POSITION "Supine"
ANREF "NA"
COIL "HEAD"
PLANE "OBLIQUE"
SEDESCFLAG "1"
SEDESC "AX FSE T2"
IMODE "2D"
PSEQ "FSE-XL"
IOPT "FC, EDR, TRF, Fast"
PLUG "22"
FILTCHOICE "None"
BWRT "-1"
TRICKSIMG "1"
TAG_SPACE "7"
TAG_TYPE "None"
\&...
```

### 7.5.5 ELSCINT Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical ELSCINT CT DICOM file.

ELSCINT Protocol Information: (01f7,26,ELSCINT1)

```
$ gdcmdump --elscint ELSCINT1_ProtocolInformation.dcm
```

```
ELSCINT1 Dumping info from tag (01f7,26,elscint1)
```

```
ELSCINT1/Item name: []
  ApprovedStep [yes]
  RefSurview [1\0]
  STD-first-img-pos [11.5]
  current-step [yes]
  ntimed-steps [0]
  orig-n-slices [390]
  protocol-file [Head_Multi_1032_usr.proc]
  protocol-name [FACE-TRAUMA/Head/Hx]
  protocol-path [/usr/diamond.root/spr/]
  protocol-step [1]
  protocol-version [2.51]
```

```
ELSCINT1/Item name: [doseright]
```

```
  ACS [n/a]
  ACS-bed-position [0]
  ACS-calc-mas [0]
  ACS-ig-parameter [0]
  ACS-learn-allowed [no]
  ACS-water-radius [-1.000000]
  ACS-water-radius-scan [-1]
\&...
```

### 7.5.6 VEPRO Protocol Information

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical VEPRO CT DICOM file.

ELSCINT Protocol Information: (0055,20,VEPRO VIM 5.0 DATA)

```
$ gdcmdump --vepro VEPRO_ProtocolInformation.dcm

VIMDATA2: (0055,20,VEPRO VIM 5.0 DATA)
  ID: VIM
  Version: 5.0
  UserName:
  UserAdress1: Name of Institution
  UserAdress2: Street of Institution
  UserAdress3: City of Institution
  UserAdress4:
  UserAdress5:
  RecDate: 20101001
  RecTime: 211321
  RecPlace:
  RecSource: DICOM Distributor
  DF1: P-09/10-41808
  DF2: Sultana Razia
  DF3: 19411001
  DF4: F
  DF5:
  DF6:
  DF7:
  DF8: CT Scan Brain without Contrast
  DF9: 10/10-0034873
  DF10: 10/10-00348
  DF11:
  DF12:
  DF13:
  DF14: Head 0.5
  DF15: 4
  DF16:
  DF17:
  DF18:
  DF19:
  DF20:
  StudyUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285934880.206831
  SeriesUID: 1.2.392.200036.9116.2.6.1.48.1214228007.1285935201.938653
  Modality: CT
```

### 7.5.7 Philips Private MR Series Data Storage (1.3.46.670589.11.0.0.12.2)

Using this option it is possible to dump as a readable text what is contained in the private attribute as found in typical Philips Private MR Series Data Storage file.

PMS Series Data Storage (2005,32,Philips MR Imaging DD 002)

```
$ gdcmdump --sds PMS_SeriesDataStorage.dcm

\&...
PMS/Item name: [PDF_CONTROL_GEN_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_RECON_PARS/IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_CONTROL_SCAN_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_EXAM_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_HARDWARE_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_PREP_PARS /IEEE_PDF/Y ]
\&...
PMS/Item name: [PDF_SPT_PARS/IEEE_PDF/Y ]
```

```

SP_scan_resol [256\256] # 2
SP_pda_profiles [0\0] # 2
SP_filter [324074] # 1
SP_analyse_with_iqt [0] # 1
SP_main_system_type [3] # 1
SP_gradient_system [6] # 1
SP_coil_type [2\2\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_id [2\34\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_part [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_act_q [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_act_coil_freq [0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_coil_m_pos [255\255\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
SP_coil_t_pos [255\128\255\0\0\0\0\0\0\0\0\0\0\0\0\255] # 16
SP_surface_coil_con [0\1\0\0\0\0\0\0\0\0\0\0\0\0\0\0] # 16
SP_proton_freq [127801349] # 1
SP_tm_result [2\2\2\2\2\2\2\2\2\2\2\2\2\2\2\2] # 16
SP_f0_result [0] # 1
SP_as_result [0] # 1
SP_po_result [0] # 1
SP_rg_result [0] # 1
SP_dc_result [0] # 1
SP_ph_result [0] # 1
\&...

```

## 7.5.8 Encapsulated ASN1 Structure

This option is mainly used for dumping the ASN1 structure of the encrypted Attribute (0040,0520)

```
$ gdcmdump encrypted.dcm
```

```

\&...
(0400,0500) SQ # u/1,1 Encrypted Attributes Sequence
  (fffe,e000) na (Item with undefined length)
    (0400,0510) UI [1.2.840.10008.1.2] # 18,1 Encrypted Content Transfer Syntax UID
    (0400,0520) OB 30\82\03\ba\06\09\2a\86\48\55\04\08\13 # 958,1 Encrypted Content
  (fffe,e00d)
(fffe,e0dd)
\&...

```

```
$ gdcmdump --asn1 encrypted.dcm
```

```

0:d=0 hl=4 l= 954 cons: SEQUENCE
4:d=1 hl=2 l= 9 prim: OBJECT :pkcs7-envelopedData
15:d=1 hl=4 l= 939 cons: cont [ 0 ]
19:d=2 hl=4 l= 935 cons: SEQUENCE
23:d=3 hl=2 l= 1 prim: INTEGER :00
26:d=3 hl=4 l= 366 cons: SET
30:d=4 hl=4 l= 362 cons: SEQUENCE
34:d=5 hl=2 l= 1 prim: INTEGER :00
37:d=5 hl=2 l= 82 cons: SEQUENCE
39:d=6 hl=2 l= 69 cons: SEQUENCE
41:d=7 hl=2 l= 11 cons: SET
43:d=8 hl=2 l= 9 cons: SEQUENCE
45:d=9 hl=2 l= 3 prim: OBJECT :countryName
50:d=9 hl=2 l= 2 prim: PRINTABLESTRING :AU
54:d=7 hl=2 l= 19 cons: SET
56:d=8 hl=2 l= 17 cons: SEQUENCE
58:d=9 hl=2 l= 3 prim: OBJECT :stateOrProvinceName
63:d=9 hl=2 l= 10 prim: PRINTABLESTRING :Some-State
75:d=7 hl=2 l= 33 cons: SET
77:d=8 hl=2 l= 31 cons: SEQUENCE
79:d=9 hl=2 l= 3 prim: OBJECT :organizationName
84:d=9 hl=2 l= 24 prim: PRINTABLESTRING :Internet Widgits Pty Ltd
110:d=6 hl=2 l= 9 prim: INTEGER :AC966D88787A51B4

```



```

121:d=5 hl=2 l= 13 cons: SEQUENCE
123:d=6 hl=2 l= 9 prim: OBJECT :rsaEncryption
134:d=6 hl=2 l= 0 prim: NULL
136:d=5 hl=4 l= 256 prim: OCTET STRING [HEX DUMP]:822368070285AD756C962ECB973514B291F946...
396:d=3 hl=4 l= 558 cons: SEQUENCE
400:d=4 hl=2 l= 9 prim: OBJECT :pkcs7-data
411:d=4 hl=2 l= 29 cons: SEQUENCE
413:d=5 hl=2 l= 9 prim: OBJECT :aes-256-cbc
424:d=5 hl=2 l= 16 prim: OCTET STRING [HEX DUMP]:3B49AFE71749F2BFF1519EBAEA95A393
442:d=4 hl=4 l= 512 prim: cont [ 0 ]

```

## 7.6 SEE ALSO

**gdcmdump(1), gdcmrw(1), gdcmanon(1)**

## 7.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 8

# Tool to generate a DICOMDIR file from a File-Set.

### 8.1 SYNOPSIS

```
gdcmgendir [options] file-in file-out
```

### 8.2 DESCRIPTION

### 8.3 PARAMETERS

file-in    DICOM input filename

file-out   DICOM output filename

### 8.4 OPTIONS

#### 8.4.1 Parameters

#### 8.4.2 OPTIONS

-i --input	DICOM filename or directory
-o --output	DICOM filename or directory
-r --recursive	recursive.
--descriptor	descriptor.
--root-uid	Root UID.

#### 8.4.3 general options

-h    --help	print this help text and exit
-v    --version	print version information and exit

```
-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

#### 8.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

### 8.5 Typical usage

### 8.6 NOTE

One may have to run some preliminary steps in order to get gdcmgendir to generate the DICOMDIR file. Namely two steps:

- Batch renaming of the DICOM filename into something compatible with ISO 9660 filename convention
- Convert all DICOM file into the Explicit VR Little Endian Uncompressed (1.2.840.10008.1.2.1)

Step 1. can be solved in a numerous way. Eg. on UNIX environment this could either be solved using the `mkisofs` command line tool. Filenames should not contains any extension since the VR CS does not allow for the '.' character. Only upper case, digit 0-9, the space ' ' and the underscore '\_' character are valid in VR CS, with a maximum of 8 bytes. Another simple tool that can be handy is 'rename' in conjunction with 'basename'.

Step 2. can simply be achieved using the `gdcconv` command line tool:

```
$ for i in `ls IMG*`; do gdcconv --raw --force $i /tmp/out/$i; done
```

### 8.7 SEE ALSO

**gdcconv(1)**, **gdcmanon(1)**, **rename(1)**, **mkisofs(1)**

### 8.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 9

# Manipulate DICOM image file.

`gdcmimg` is a low level tool to allow de-/encapsulation from/to DICOM image. This tool does not understand Transfer Syntax conversion. It will encapsulate the raw data as-is. This has some impact in some cases, see special warnings below.

It is important to note that `gdcmimg` can only encapsulate proper input file, for instance JPG and or JP2 are accepted since an associated DICOM Transfer Syntax can be found. However input such as TIFF and/or PNG are not, since DICOM does not support those. See instead a tool such as `gdcm2vtk`.

### 9.1 SYNOPSIS

```
gdcmimg [options] file-in file-out
```

### 9.2 DESCRIPTION

The **`gdcmimg`** command line tool can be used in two fashions:

- 1. Converting a recognized file format into its encapsulated DICOM counterpart,
- 2. Anonymizing a rectangular portion of a DICOM file.

### 9.3 PARAMETERS

```
file-in    input filename (non-DICOM)
```

```
file-out   DICOM output filename
```

### 9.4 OPTIONS

#### 9.4.1 PARAMETERS

```
-i --input      Input filename  
-o --output     Output filename
```

## 9.4.2 OPTIONS

```
--endian %s      Endianness (LSB/MSB).
-d --depth %d    Depth (Either 8/16/32 or BitsAllocated eg. 12 when known).
--sign %s        Pixel sign (0/1).
--spp %d         Sample Per Pixel (1/3).
-s --size %d,%d  Size.
-C --sop-class-uid SOP Class UID (name or value).
-T --study-uid   Study UID.
-S --series-uid  Series UID.
--root-uid       Root UID.
```

## 9.4.3 fill options

```
-R --region %d,%d Region.
-F --fill %d      Fill with pixel value specified.
```

## 9.4.4 general options

```
-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information
```

## 9.4.5 environment variable

```
GDCM_ROOT_UID Root UID
```

## 9.5 Supported File Format (appropriate file extension) gdcming

will base it's conversion process based on the file extension. Follows the list of recognized file extension. When no extension is found, DICOM file is assumed.

input format

```
* RAW      (raw, rawl, gray, rgb)
* RLE      (rle)
* PNM      (pgm, pnm, ppm)
* JPEG-LS  (jls)
* JPEG 2000 (jp2, j2k, j2c, jpx, jpc)
* JPEG     (jpg, jpeg, ljpg, ljpeg)
* DICOM    ()
```

output format:

```
* PGM      (pgm, pnm, ppm)
* DICOM    ()
```

For RAW file format, you should take special care of the `--endian` option. For the (old) JPEG file format, both the lossy and lossless format are supported, user should pay attention to the `--sign` option. For file format such as RLE or RAW, user is expected to fill in information required to find the dimension and type of input data as there is no other way to find this information. For all other file format, the properties are derived from the file format itself.

PNM file are supposed to be big endian (important for depth > 8)

## 9.6 Typical usage

### 9.6.1 Remove a rectangular part of the image

To fill the region [0,100]x[0,100] of a DICOM image simply do:

```
$ gdcimg --fill 0 --region 0,100,0,100 -i input.dcm -o output_black.dcm
```

Warning: if the Pixel Data is compressed, the image is first decompressed so that pixel can be set to 0, but it is not re-compressed.

### 9.6.2 Convert RAW to DICOM

Recognized extension is `.raw`, `.rawl`, `.gray` or `.rgb` (case insensitive)

```
$ gdcimg --size 512,512 --depth 16 -i input.raw -o output.dcm
```

the image will be a Secondary Capture.

When the input is 3 component, one need to specify explicitly the Samples Per Pixel:

```
$ gdcimg --size 512,512 --spp 3 input_rgb.raw output_rgb.dcm
```

When the filename contains `.rgb` as file extension output is automatically recognized as RGB no need to specify `--spp`

```
$ gdcimg --size 512,512 input.rgb output_rgb.dcm
```

You can use the `dd` cmd line to skip any header you would like to discard, for instance, if you would like to skip the first 108 bytes, simply do:

```
$ dd skip=108 bs=1 if=input.raw of=output.raw
```

`.raw` and `.rawl` extension are equivalent. You need to explicitly specify the endianness manually:

```
$ gdcimg --endian MSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

or

```
$ gdcimg --endian LSB --size 512,512 --depth 16 -i input.raw -o output.dcm
```

### 9.6.3 Convert PGM/PNM/PPM to DICOM

Recognized extensions are .pgm, .pnm, .ppm (case insensitive)

```
$ gdcming -i input.pgm -o output.dcm
```

the image will be a Secondary Capture

### 9.6.4 Convert RLE to DICOM

Recognized extension is .rle (case insensitive)

```
$ gdcming --size 512,512 --depth 16 -i input.rle -o output.dcm
```

the image will be a Secondary Capture

### 9.6.5 Convert JPEG to DICOM

Recognized extensions are .jpg, .jpeg, .ljpg, .ljpeg (case insensitive)

```
$ gdcming -i input.ljpeg -o output.dcm
```

the image will be a Secondary Capture

### 9.6.6 Convert J2K to DICOM

Recognized extensions are .j2k, .jp2, .jpc, .jpx, .j2c (case insensitive)

```
$ gdcming -i input.j2k -o output.dcm
```

the image will be a Secondary Capture.

All Pixel informations (Bits Stored/Allocated...) will be derived from the image itself, and not from the command line options.

### 9.6.7 Specifying a SOP Class UID

Instead of the default Secondary Capture Image Storage, one may want to specify, say VL Photographic Image Storage.

```
$ gdcming --sop-class-uid 1.2.840.10008.5.1.4.1.1.77.1.4 input.jpg output.dcm
```

## 9.7 Multiple Files

gdcming handle nicely a set of files (for instance jpeg):

```
$ gdcming 1.jpg 2.jpg 3.jpg 4.jpg output.dcm
```



## 9.8 Start Offset

In some case, one may want to create a 2D slice from an arbitrary volume (e.g 3D). In which case `--offset` becomes handy:

```
$ gdcming --offset 4954104330 --size 1673,1673 Input3D_1673_1673_1775.raw slice_1770.dcm
```

## 9.9 Warning

There are a couple of issues with `gdcming` implementation:

For RAW file, one should pay attention that when using `--endian MSB` the Pixel Data will be encapsulated as is (not touched by `gdcming`). Therefore the only possible transfer syntax available is Implicit VR Big Endian DLX (G.E Private). GDCM does handle this private Transfer Syntax. So if you need to convert this Transfer Syntax to another one (and allow Pixel Data manipulation), you can use:

```
$ gdcconv --raw --force input_big_endian_dlx.raw -o output_implicit_vr_little_endian.dcm
```

For JFIF file and JP2 file (with header) the header is copied into the Pixel Data element which is illegal for JP2. Use `gdcconv` to properly re-encode a JP2/JFIF file into J2K/JPG.

```
$ gdcming input.jp2 output_jp2.dcm
$ gdcconv --j2k --force output_jp2.dcm output_j2k.dcm
```

For RLE file, no check is done for crossing the row boundary. It is recommended to use `gdcconv -rle` to re-encode into a proper RLE file in case of doubt.

Of course if the compression is not ok with your setup, you can always de-encapsulated the DICOM file (typically JPEG) to a non-encapsulated form, using `gdcconv`:

```
$ gdcconv --raw input_jpeg.dcm output_raw.dcm
```

## 9.10 SEE ALSO

**`gdcmdump(1)`, `gdc2vtk(1)`, `gdc2raw(1)`, `convert(1)`, `dd(1)`**

## 9.11 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 10

# Display meta info about the input DICOM file.

### 10.1 SYNOPSIS

```
gdcminfo [options] file-in
```

### 10.2 DESCRIPTION

The **gdcminfo** command line program takes as input a DICOM file, or a directory and process it to extract meta-information about the DICOM file processed.

### 10.3 PARAMETERS

```
file-in    DICOM input filename
```

### 10.4 OPTIONS

#### 10.4.1 OPTIONS

-r --recursive	recursive.
-d --check-deflated	check if file is proper deflated syntax.
--resources-path	Resources path.
--md5sum	Compute md5sum of Pixel Data attribute value.
--check-compression	check the encapsulated stream compression (lossless/lossy).

#### 10.4.2 general options

-h	--help	print this help text and exit
-v	--version	print version information and exit
-V	--verbose	verbose mode (warning+error).

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

### 10.4.3 environment variable

GDCM\_RESOURCES\_PATH path pointing to resources files (Part3.xml, ...)

## 10.5 Simple usage

### 10.5.1 gdcmdata

Using data from gdcmdata:

```
$ gdcminfo gdcmdata/012345.002.050.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
NumberOfDimensions: 2
Dimensions: (256,256)
Origin: (-85,21.6,108.7)
Spacing: (0.664062,0.664062,1.5)
DirectionCosines: (1,0,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel      :1
BitsAllocated        :16
BitsStored           :16
HighBit              :15
PixelRepresentation:1
Orientation Label: CORONAL
```

### 10.5.2 Davie Clunie datasets:

Using data from David Clunie datasets:

```
$ gdcminfo BRTUM001.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4.1 [Enhanced MR Image Storage]
NumberOfDimensions: 3
Dimensions: (256,256,15)
Origin: (40,-105,105)
Spacing: (0.820312,0.820312,6)
DirectionCosines: (0,1,0,0,0,-1)
Rescale Intercept/Slope: (0,1)
SamplesPerPixel      :1
BitsAllocated        :16
BitsStored           :16
HighBit              :15
PixelRepresentation:1
Orientation Label: SAGITTAL
```

### 10.5.3 Checking the md5sum of the Pixel Data

After compressing a DICOM file (see `gdcmconv`) using a lossless compression algorithm, it is fairly easy to compare the two files for differences at DICOM attribute level. However one operation is slightly easier to do: how to make sure the compression was actually lossless ? In this case one could use the `--md5sum` operation.

Take an uncompressed DICOM image file:

```
$ gdcminfo --md5sum SIEMENS_ImageLocationUN.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

Now compress this file:

```
$ gdcmmconv --jpegls SIEMENS_ImageLocationUN.dcm lossless_compressed.dcm
```

and then check again the md5sum:

```
$ gdcminfo --md5sum lossless_compressed.dcm
```

The tool return: 0621954acd5815e0b4f7b65fcc6506b1

### 10.5.4 Checking if Pixel Data is lossless

In some environment one wish to check whether or not the DICOM file is lossless. It is fairly easy to do that in most cases. Only in two occasion this is not clear from the sole DICOM Attribute. When the Transfer Syntax is JPEG 2000 Image Compression (1.2.840.10008.1.2.4.91) and when the Transfer Syntax is JPEG-LS Lossy (Near-Lossless) Image Compression (1.2.840.10008.1.2.4.81).

In this case, the only solution is to open the Pixel Data element, read the specific JPEG header and check whether or not the JPEG transformation was lossless or not:

```
$ gdcminfo --check-compression gdcmmData/MAROTECH_CT_JP2Lossy.dcm
```

The tool returns: "Encapsulated Stream was found to be: lossy"

## 10.6 SEE ALSO

`gdcmdump(1)`, `gdcmraw(1)`, `gdcmconv(1)`

## 10.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 11

# Tool to convert PAPYRUS 3.0 to DICOM.

### 11.1 SYNOPSIS

```
gdcmconv [options] file-in file-out
```

### 11.2 DESCRIPTION

The **gdcmconv** command line program takes as input a PAPYRUS 3.0 file (file-in) and process it to generate an output (pseudo) DICOM file (file-out). The command line option dictate the type of operation(s) gdcmconv will use to generate the output file.

### 11.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out    DICOM output filename
```

### 11.4 OPTIONS

#### 11.4.1 PARAMETERS

```
-i --input      DICOM filename
-o --output      DICOM filename
```

#### 11.4.2 OPTIONS

```
-S --split      Split multiframe PAPYRUS 3.0 into multiples DICOM files
--decomp-pap3   Use PAPYRUS 3.0 for decompressing (can be combined with --split).
--check-iop     Check that the Image Orientation (Patient) Attribute is ok (see --split).
```

#### 11.4.3 general options

```
-h --help
```

```
    print this help text and exit

-v  --version
    print version information and exit

-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

#### 11.4.4 environment variable

```
GDCM_ROOT_UID Root UID
```

### 11.5 Simple usage

**gdcmap3** is a great tool to convert broken PAPYRUS 3.0 implementation into properly parsable DICOM file. Usage is simply:

```
$ gdcmap3 input.pa3 output.dcm
```

or if you prefer being explicit:

```
$ gdcmap3 -i input.pa3 -o output.dcm
```

Even though **gdcmap3** can overwrite directly on the same file (input.pa3 = output.dcm), it is recommended that user should first convert into a different file to make sure the bug is properly handled by GDCM.

### 11.6 SEE ALSO

**gdcmdump**(1), **gdcmap3**(1), **gdcminfo**(1)

### 11.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 12

# Tool to convert PDF to PDF/DICOM.

### 12.1 SYNOPSIS

```
gdcmpdf [options] file-in file-out
```

### 12.2 DESCRIPTION

The **gdcmpdf** tool convert a PDF file (any PDF version) into an encapsulated PDF/DICOM file. By default it will try to read the PDF meta information stored in the PDF and convert this information to some specific DICOM fields (see below). However it may fails (eg. wrong password on encrypted PDF file) in which case empty value are used.

### 12.3 PARAMETERS

file-in    PDF input filename

file-out   DICOM output filename

### 12.4 OPTIONS

#### 12.4.1 general options

```
-h    --help  
      print this help text and exit  
  
-v    --version  
      print version information and exit  
  
-V    --verbose  
      verbose mode (warning+error).  
  
-W    --warning  
      warning mode, print warning information  
  
-E    --error  
      error mode, print error information  
  
-D    --debug
```

```
debug mode, print debug information
```

## 12.5 Usage Example

```
$ wget http://gdcm.sourceforge.net/gdcm.pdf
$ gdcmpdf gdcm.pdf gdcm.dcm
```

To re-extract the encapsulated PDF file:

```
$ gdcmmraw -i gdcm.dcm -t 42,11 -o gdcm.dcm.pdf
$ diff gdcm.pdf gdcm.dcm.pdf
```

## 12.6 PDF Info Mapping

Here is how the PDF info is mapped to DICOM information (typical pdfinfo output):

```
Title:      GDCM Reference Manual
Subject:    Grassroots DICOM API reference
Keywords:   GDCM,DICOM,JPEG, Lossless JPEG,JPEG-LS,J2K, JPEG 2000,RLE
Author:     Mathieu Malaterre and co.
Creator:    LaTeX with hyperref package
Producer:   pdfTeX-1.21a
CreationDate: Tue Apr 28 15:34:26 2009
Tagged:     no
Pages:      1188
Encrypted:  no
Page size:  612 x 792 pts (letter)
File size:  13756841 bytes
Optimized:  yes
PDF version: 1.4
```

Converted to DICOM this leads to:

```
# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0005) CS [ISO_IR 100] # 10, 1 SpecificCharacterSet
(0008,0012) DA [20090428] # 8, 1 InstanceCreationDate
(0008,0013) TM [182550.302631] # 14, 1 InstanceCreationTime
(0008,0016) UI =EncapsulatedPDFStorage # 30, 1 SOPClassUID
(0008,0018) UI [1.2.826.0.1.3680043.2.1143.776842935192792959289022034349197114] # 64, 1 SOPInstanceUID
(0008,0020) DA [20090428] # 8, 1 StudyDate
(0008,0023) DA [20090428] # 8, 1 ContentDate
(0008,002a) DT [20090428153437.000000] # 22, 1 AcquisitionDateTime
(0008,0030) TM [182550.302160] # 14, 1 StudyTime
(0008,0033) TM [153426.000000] # 14, 1 ContentTime
(0008,0050) SH (no value available) # 0, 0 AccessionNumber
(0008,0060) CS [OT] # 2, 1 Modality
(0008,0064) CS [WSD] # 4, 1 ConversionType
(0008,0070) LO [LaTeX with hyperref package] # 28, 1 Manufacturer
(0008,0090) PN (no value available) # 0, 0 ReferringPhysiciansName
(0010,0010) PN [Mathieu Malaterre and co.] # 26, 1 PatientsName
(0010,0020) LO (no value available) # 0, 0 PatientID
(0010,0030) DA (no value available) # 0, 0 PatientsBirthDate
(0010,0040) CS (no value available) # 0, 0 PatientsSex
(0018,1020) LO [pdfTeX-1.21a] # 14, 1 SoftwareVersions
(0020,000d) UI [1.2.826.0.1.3680043.2.1143.1868121832223417351654232480755123133] # 64, 1 StudyInstanceUID
(0020,000e) UI [1.2.826.0.1.3680043.2.1143.1330099150825746617507846107663964311] # 64, 1 SeriesInstanceUID
(0020,0010) SH (no value available) # 0, 0 StudyID
(0020,0011) IS [1] # 2, 1 SeriesNumber
```

```

(0020,0013) IS [1] # 2, 1 InstanceNumber
(0028,0301) CS [YES] # 4, 1 BurnedInAnnotation
(0040,a043) SQ (Sequence with explicit length #=0) # 0, 1 ConceptNameCodeSequence
(ffff,e0dd) na (SequenceDelimitationItem for re-encod.) # 0, 0 SequenceDelimitationItem
(0042,0010) ST [GDCM Reference Manual] # 22, 1 DocumentTitle
(0042,0011) OB 25\\50\\44\\46\\2d\\31\\2e\\34\\0a\\25\\e7\\f3\\cf\\d3\\0a\\33\\32\\30\\37\\37\\20\\30... # 137568
(0042,0012) LO [application/pdf] # 16, 1 MIMETimeTypeOfEncapsulatedDocument

```

```

$ stat gdc.m.pdf
  File: `gdc.m.pdf'
  Size: 13756841      Blocks: 26912      IO Block: 4096   regular file
Device: fe01h/65025d Inode: 2675750      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1002/mmalaterre)   Gid: ( 1002/mmalaterre)
Access: 2009-04-28 16:05:00.000000000 +0200
Modify: 2009-04-28 15:34:37.000000000 +0200
Change: 2009-04-28 16:05:00.000000000 +0200

```

Explanation for the different Date/Time mappings:

- Study Date/Time, Instance Creation Date/Time are both equal to the current time gdc.mpdf tool was run,
- Acquisition Date Time is set to the Modify Time of the actual PDF file,
- Content Date/Time are set from the actual PDF header info: CreationDate.

## 12.7 SEE ALSO

**gdc.mconv(1)**, **gdc.mraw(1)**, **pdfinfo(1)**

## 12.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 13

# Extract Data Element Value Field.

### 13.1 SYNOPSIS

```
gdcmmraw [options] file-in file-out
```

### 13.2 DESCRIPTION

The **gdcmmraw** tool is mostly used for development purpose. It is used to extract a specific binary field from a DICOM DataSet.

### 13.3 PARAMETERS

```
file-in    DICOM input filename
```

```
file-out    output filename
```

### 13.4 OPTIONS

#### 13.4.1 PARAMETERS

```
-i --input      Input filename
-o --output      Output filename
-t --tag        Specify tag to extract value from.
```

#### 13.4.2 OPTIONS

```
-S --split-frags  Split fragments into multiple files.
-p --pattern      Specify trailing file pattern (see split-frags).
-P --pixel-data   Pixel Data trailing 0.
```

#### 13.4.3 general options

```
-h    --help
```





```
-rw-r--r-- 1 mathieu mathieu 81512 2008-08-08 22:10 jpeg03.ljpeg  
-rw-r--r-- 1 mathieu mathieu 81694 2008-08-08 22:10 jpeg02.ljpeg  
-rw-r--r-- 1 mathieu mathieu 81564 2008-08-08 22:10 jpeg01.ljpeg  
-rw-r--r-- 1 mathieu mathieu 79970 2008-08-08 22:10 jpeg00.ljpeg
```

## 13.6 Footnote about JPEG files

It is a common misunderstanding to interchange 'JPEG 8bits lossy' with simply JPEG file. The JPEG specification is much broader than simply the common lossy 8bits file (as found on internet).

You can have:

- JPEG Lossy 8bits
- JPEG Lossy 12bits
- JPEG Lossless 2-16bits

Those are what is defined in ITU-T T.81, ISO/IEC IS 10918-1.

## 13.7 SEE ALSO

**gdcmdump(1)**, **gdcmrw(1)**

## 13.8 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 14

# Scan a directory containing DICOM files.

### 14.1 SYNOPSIS

```
gdcmscanner [options] directory
```

### 14.2 DESCRIPTION

The **gdcmscanner** is a command line tool to quickly extract value from a set of DICOM attribute in a DICOM File-Set.

#### 14.2.1 PARAMETERS

```
-d --dir          DICOM directory
-t --tag %d,%d    DICOM tag(s) to look for
```

#### 14.2.2 OPTIONS

```
-p --print        Print output.
-r --recursive    Recursively descend directory.
```

#### 14.2.3 general options

```
-h  --help
    print this help text and exit

-v  --version
    print version information and exit

-V  --verbose
    verbose mode (warning+error).

-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

## 14.3 Typical usage

## 14.4 Simple usage

In order to display all the value for Patient Name (0010,0010) in the directory name **gdcMData**, simply do:

```
$ gdcmscanner -t 10,10 -d gdcMData -p
```

## 14.5 Complex usage

Because gdcmscanner does not support progress, you have to wait until all files are traversed to see any results. This is quite cumbersome, on UNIX this can be worked around with the following trick:

```
$ find gdcMData -type d -exec gdcmscanner -t 10,10 -d {} -p ';'`
```

So all directory are locally traversed (no child directory are recursively traversed), which means results comes out much faster.

## 14.6 SEE ALSO

**gdcmdump(1)**, **gdcmrw(1)**

## 14.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 15

# Tool to execute a DICOM Query/Retrieve operation

### 15.1 SYNOPSIS

```
gdcmscu [OPTION]...[OPERATION]...HOSTNAME...[PORT]...
```

Execute a DICOM Q/R operation to HOSTNAME, using port PORT (104 when not specified)

### 15.2 DESCRIPTION

The **gdcmscu** command line program is the tool to execute DICOM Query/Retrieve operation. It supports:

- C-ECHO (SCU)
- C-FIND (SCU)
- C-STORE (SCU)
- C-MOVE (SCU/SCP) C-MOVE operation are executed using two different ports (one for the SCU and one for the SCP).

### 15.3 PARAMETERS

### 15.4 OPTIONS

#### 15.4.1 OPTIONS

```
-H --hostname    %s  Hostname.  
-p --port        %d  Port number.  
    --aetitle    %s  Set calling AE Title.  
    --call       %s  Set called AE Title.
```

#### 15.4.2 mode options

```
--echo          C-ECHO (default when none).  
--store         C-STORE.
```

```
--find      C-FIND.
--move      C-MOVE.
```

### 15.4.3 C-STORE options

```
-i --input      %s  DICOM filename
-r --recursive  recursively process (sub-)directories
--store-query %s  Store constructed query in file
```

### 15.4.4 C-FIND/C-MOVE options

```
--patientroot  C-FIND Patient Root Model.
--studyroot    C-FIND Study Root Model.

--patient      C-FIND Query on Patient Info (cannot be used with --studyroot).
--study        C-FIND Query on Study Info.
--series       C-FIND Query on Series Info.
--image        C-FIND Query on Image Info.
--key %d,%d[%s] 0123,4567=VALUE for specifying search criteria (wildcard allowed)
                With --key, leave blank (ie, --key 10,20="" or --key 10,20) to retrieve values
```

### 15.4.5 C-MOVE options

```
-o --output      %s  DICOM filename / directory
--port-scp %d      Port for incoming associations
--key %d,%d[%s]    0123,4567=VALUE for specifying search criteria (wildcard not allowed)
                Note that C-MOVE supports the same queries as C-FIND, but no wildcards are allowed
```

### 15.4.6 general options

```
-h --help
    print this help text and exit

-v --version
    print version information and exit

-V --verbose
    verbose mode (warning+error).

-W --warning
    warning mode, print warning information

-E --error
    error mode, print error information

-D --debug
    debug mode, print debug information

-L --log-file
    specify a filename where to write logs

--queryhelp
    print query help
```

### 15.4.7 environment variable

```
GDCM_ROOT_UID Root UID
```

## 15.5 C-ECHO usage

**gdcm SCU** is a great tool to test if a DICOM server is up. For example to send a C-ECHO to server `dicom.example.com` using port 104, use:

```
$ gdcm SCU dicom.example.com
```

or if you prefer being explicit:

```
$ gdcm SCU --echo dicom.example.com 104
```

Using basic security your DICOM server might require that you set the appropriate called AE-TITLE

```
$ gdcm SCU --echo dicom.example.com 11112 --call SERVSCP
```

If you want to specify your own AE-TITLE (default is GDCMSCU), simply use:

```
$ gdcm SCU --echo dicom.example.com 11112 --call SERVSCP --aetitle MYSCU
```

For example you could test on the DICOM server provided by DICOMObject team:

```
$ gdcm SCU www.dicomserver.co.uk 11112
```

## 15.6 C-STORE usage

C-STORE is the operation that allow sending a DICOM file to a remote DICOM server. For instance to send a file called `myfile.dcm`

```
$ gdcm SCU --store dicom.example.com 104 myfile.dcm
```

or if you prefer being explicit:

```
$ gdcm SCU --store dicom.example.com 104 -i myfile.dcm
```

You can even send multiple files using the same association:

```
$ gdcm SCU --store dicom.example.com 104 myfile1.dcm myfile2.dcm myfile3.dcm ...
```

## 15.7 C-FIND usage

**gdcm SCU** also allow querying a DICOM server. This is the C-FIND operation, for example to find all DICOM Instance where PatientsName match a particular pattern, usage is simply:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --patientroot --key 10,10,"A*"
```

We also support a DCMTK compatible convention:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --patientroot --key 10,10="A*"
```

When an attribute is set without a value it will be part of the output result:

```
$ gdcm SCU --find --patient dicom.example.com 11112 --call MI2B2 --patientroot -k 10,10="A*" -k 10,20
```

## 15.8 C-MOVE usage

C-MOVE is the operation to retrieve a DICOM instance from a remote DICOM server. Most of the time, it is a subsequent operation after a C-FIND query. To retrieve a DICOM instance where PatientID is ABCD1234, simply execute:

```
$ gdcmscu --move --patient --aetitle ACME1 --call ACME_STORE dicom.example.com 5678 --patientroot -k 10,20="ABCD1234"
```

**WARNING** For this operation to work you need information from the DICOM server you are communicating with. Only the DICOM server you are sending a C-MOVE query will be responsible for sending back incoming associations (the actual C-STORE SCP). Therefore you need to make sure that you mapping of (AE-TITLE,PortNumber) is properly set on the DICOM server side as well as the port for incoming association (`--port-scp`).

**gdcmscu** does not currently support external C-STORE association (C-STORE request sent to an external SCP application).

## 15.9 patientroot notes

The flag `--patientroot` is just simply a wrapper around the syntax `--key 8,52=PATIENT` For instance one would write using DCMTK syntax:

```
$ findscu --patient dicom.example.com 11112 --key 8,52=PATIENT --key 10,10="F*"
```

This would become using GDCM syntax:

```
$ gdcmscu --find --patient dicom.example.com 11112 --patientroot --key 10,10="F*"
```

## 15.10 Debugging

This is sometime difficult to investigate why a connection to a remote DICOM server cannot be done. Some recommendations follows:

Always try to do a simple C-ECHO at first. If you cannot get the C-ECHO to work none of the other operations will work. Before trying to a C-MOVE operation, make sure you can execute the C-FIND equivalent query first.

When doing a C-MOVE operation you really need to communicate with the PACS admin as the C-MOVE operation is different from the other lower level operation such as HTTP/GET. When doing a C-MOVE, the server will communicate back using another channel (could be different port) using it's internal database to map an AE-TITLE back to the destination IP.

Indeed the C-MOVE operation by design does not always use your incoming IP address to send back the resulting dataset. Instead it uses a mapping of AE-TITLE to IP address to send back any results. So pay particular attention to the spelling of your AE-TITLE and your incoming port (which may be different from the port to connect to the server).

## 15.11 Port Warning

Watch out that port ranging [1-1024] are reserved for admin and not easily accessible unless granted special privileges. Therefore the default 104 DICOM port might not be accessible to all your users.

## 15.12 C-STORE Warnings

When constructing a C-STORE operation, `gdcm SCU` will always use the Media Storage SOP Class UID as found in the file to be sent. For encapsulated DICOM file (eg. RLE Lossless) the receiving SCP server might not support this compression and will legitimately refuse the C-STORE operation. In this case users have to manually convert to a non-compressed form this particular file:

```
$ gdcmconv --raw compressed.dcm non_compressed.dcm
```

## 15.13 C-MOVE Warnings

At the moment `gdcm SCU` only supports non-compressed transfer syntax. It will always request DataSet using Implicit VR Little Endian Transfer Syntax during a C-MOVE operation (both incoming and outgoing associations). This make `gdcm SCU -move` equivalent to DCMTK `movescu` syntax:

```
$ movescu -xi +xi ...
```

## 15.14 C-FIND IMAGE level (Composite Object Instance)

One should pay attention that `gdcm SCU -find` and `findscu` are not completely equivalent. Using `gdcm SCU -find`, all Unique Keys will be added automatically. One can therefore execute something like this:

```
$ gdcm SCU --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112
```

instead of the more explicit form

```
$ gdcm SCU --find --patientroot --image --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

This would also be equivalent to:

```
$ findscu --patient --key 8,52=IMAGE --key 8,18=1.2.3.4.5.6 dicom.example.com 11112 --key 10,20 --key 20,d --key
```

## 15.15 Storing the Query

It is also possible to store the query:

```
gdcm SCU --find --patient --patientroot dicom.example.com 11112 --key 10,20="*" --key 10,10 --store-query query.dcm
```

One can then check the DataSet values send for the query:

```
$ gdcmdump query.dcm
# Dicom-File-Format

# Dicom-Meta-Information-Header
# Used TransferSyntax:

# Dicom-Data-Set
# Used TransferSyntax: 1.2.840.10008.1.2
(0008,0005) ?? (CS) [ISO_IR 192] # 10,1-n Specific Character Set
(0008,0052) ?? (CS) [PATIENT ] # 8,1 Query/Retrieve Level
(0010,0010) ?? (PN) (no value) # 0,1 Patient's Name
(0010,0020) ?? (LO) [* ] # 2,1 Patient ID
```

The Specific Character Set was set to "ISO\_IR 192" as the locale encoding of the system was found automatically by gdcmscu to be UTF-8.

This means that the following command line will properly setup the Query with the appropriate Charset to be executed correctly:

```
$ gdcmscu --find --patient --patientroot dicom.example.com 11112 --key 10,10="*Jérôme"
```

The query is always executed on the server side (SCP), some implementations does not support string matching with different Character Set.

## 15.16 DICOM Public Servers

An up to date list of DICOM Public Servers can be found at:

<http://www.dclunie.com/medical-image-faq/html/part8.html#DICOMPublicServers>

## 15.17 SEE ALSO

**gdcmscu**(1)

## 15.18 COPYRIGHT

Copyright Insight Software Consortium



## Chapter 16

# Concatenate/Extract DICOM files.

### 16.1 SYNOPSIS

```
gdcmtar [options] file-in file-out
```

### 16.2 DESCRIPTION

The **gdcmtar** is a command line tool used to tar/untar multi-frames images (including SIEMENS MOSAIC file)

### 16.3 PARAMETERS

file-in DICOM input filename

file-out DICOM output filename

### 16.4 OPTIONS

#### 16.4.1 OPTIONS

```
--enhance      enhance (default)
-U --unenhance  unenhance
-M --mosaic     Split SIEMENS Mosaic image into multiple frames.
-p --pattern    Specify trailing file pattern.
--root-uid      Root UID.
```

#### 16.4.2 general options

```
-h --help      print this help text and exit
-v --version    print version information and exit
-V --verbose    verbose mode (warning+error).
```

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

### 16.4.3 environment variable

GDCM\_ROOT\_UID Root UID

## 16.5 Typical usage

### 16.5.1 SIEMENS Mosaic

```
$ gdcminfo MR-sonata-3D-as-Tile.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]
NumberOfDimensions: 2
Dimensions: (384,384,1)
\&...
```

```
$ gdcmtar --mosaic -i MR-sonata-3D-as-Tile.dcm -o mosaic --pattern %03d.dcm
```

Will output:

```
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic000.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic001.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic002.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic003.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic004.dcm
-rw-r--r-- 1 mathieu mathieu 72886 2009-08-10 11:14 mosaic005.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic006.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic007.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic008.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic009.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic010.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic011.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic012.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic013.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic014.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic015.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic016.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic017.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic018.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic019.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic020.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic021.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic022.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic023.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic024.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic025.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic026.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic027.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic028.dcm
-rw-r--r-- 1 mathieu mathieu 72884 2009-08-10 11:14 mosaic029.dcm
-rw-r--r-- 1 mathieu mathieu 72882 2009-08-10 11:14 mosaic030.dcm
```

```
$ gdcminfo mosaic000.dcm
```

```
MediaStorage is 1.2.840.10008.5.1.4.1.1.4 [MR Image Storage]  
TransferSyntax is 1.2.840.10008.1.2.1 [Explicit VR Little Endian]  
NumberOfDimensions: 2  
Dimensions: (64,64,1)  
\&...
```

## 16.6 SEE ALSO

**gdcmdump(1)**, **gdcmrw(1)**, **gdcminfo(1)**

## 16.7 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 17

# Simple DICOM viewer.

### 17.1 SYNOPSIS

```
gdcviewer [options] file-in
```

### 17.2 DESCRIPTION

The **gdcviewer** is a simple tool that show how to use [vtkGDCMImageReader](#). The class that use gdc to make a layer to VTK. **gdcviewer** is basically only just a wrapper around VTK/GDCM.

This tool is meant for testing integration of GDCM in VTK. You should see it as a demo tool. It does compile with VTK ranging from 4.2 to 5.10, but only with VTK 5.2 (or above) can play with the widgets (as described below).

### 17.3 PARAMETERS

```
file-in    DICOM input filename
```

### 17.4 OPTIONS

#### 17.4.1 OPTIONS

<code>--force-rescale</code>	force rescale (advanced users)
<code>--force-spacing</code>	force spacing (advanced users)
<code>-r --recursive</code>	Recursively descend directory

#### 17.4.2 general options

<code>-h</code>	<code>--help</code>	print this help text and exit
<code>-v</code>	<code>--version</code>	print version information and exit
<code>-V</code>	<code>--verbose</code>	verbose mode (warning+error).

```
-W  --warning
    warning mode, print warning information

-E  --error
    error mode, print error information

-D  --debug
    debug mode, print debug information
```

## 17.5 Typical usage

## 17.6 Simple usage

For now `gdcmviewer` should be started from a command line prompt. The next argument should be the name of the DICOM file you wish to read. For instance:

```
$ gdcmviewer -V 012345.002.050.dcm
```

`gdcmviewer` will try to read your file, and then print the `vtk` information associated with this file. Basically what kind of image you are looking at.

- `ScalarType` is the DICOM Real World Value type
- `Dimensions` is the dimension of the image
- `Spacing` is the spacing of the image
- `NumberOfScalarComponents` should be 1 for grayscale & `PALETTE COLOR` and 3 for `RGB`, `YBR` data.

## 17.7 Wiki Link

The wiki page, with color pictures can be found at: <http://gdcm.sourceforge.net/wiki/index.php/Gdcmviewer>

## 17.8 SEE ALSO

`gdcmdump(1)`, `gdcm2vtk(1)`

## 17.9 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre

## Chapter 18

**provides a tool to convert a DICOM file into a XML info set and vice-versa.**

### 18.1 SYNOPSIS

```
gdcmxml [options] file-in[DICOM or XML] file-out[XML or DICOM]
```

### 18.2 DESCRIPTION

The **gdcmxml** command line program converts a DICOM file (DataSet) into an XML file (according to the Native DICOM Model) or vice-versa. For those familiar with DCMTK, this provides binary capabilities (i.e. functionality of both dcm2xml and xml2dcm).

The XML info set which is from the DICOM file gdcmXMLPrintet Class. This is in strict compliance with the Native DICOM Model as given in Supp 118.

### 18.3 PARAMETERS

```
file-in    DICOM or XML input filename ( cannot be absent)
```

```
file-out   output filename (can be absent)
```

### 18.4 OPTIONS

#### 18.4.1 PARAMETERS

```
-i --input      DICOM filename  
-o --output     DICOM filename
```

#### 18.4.2 Options for DICOM to XML:

```
-B --loadBulkData  Loads bulk data into a binary file named "UUID" (by default UUID are written).
```

### 18.4.3 Options for XML to DICOM:

```
-B --loadBulkData  Loads bulk data from a binary file named as the "UUID" in XML file (by default UUID are writ  
-T --TransferSyntax Loads transfer syntax from file (default is LittleEndianImplicit)
```

### 18.4.4 general options

```
-h  --help  
    print this help text and exit  
  
-v  --version  
    print version information and exit  
  
-V  --verbose  
    verbose mode (warning+error).  
  
-W  --warning  
    warning mode, print warning information  
  
-E  --error  
    error mode, print error information  
  
-D  --debug  
    debug mode, print debug information
```

## 18.5 SEE ALSO

**gdcmdump(1), gdcconv(1)**

## 18.6 COPYRIGHT

Copyright (c) 2006-2011 Mathieu Malaterre



## Chapter 19

# 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 20

# Deprecated List

**Member `gdcm::CompositeNetworkFunctions::ConstructQuery`** (ERootType inRootType, EQueryLevel inQueryLevel, const KeyValuePairArrayType &keys, bool inMove=false)

**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 21

# Bug List

### Class `gdcm::DICOMDIRGenerator`

: There is a current limitation of not handling Referenced SOP Class UID / Referenced SOP Instance UID simply because the `gdcm::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:

- Gantry Tilt is not considered



## Chapter 22

# Namespace Index

### 22.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">gdc</a>	111
<a href="#">gdc::network</a>	133
<a href="#">gdc::SegmentHelper</a>	138
<a href="#">gdc::terminal</a>	
Class for Terminal Allow one to print in color in a shell	138





## Chapter 23

# Hierarchical Index

### 23.1 Class Hierarchy

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

gdcn::network::AbstractSyntax	152
gdcn::network::ApplicationContext	162
gdcn::ApplicationEntity	163
gdcn::network::ARTIMTimer	168
gdcn::ASN1	169
gdcn::network::AsynchronousOperationsWindowSub	170
gdcn::Attribute< Group, Element, TVR, TVM >	171
gdcn::Attribute< Group, Element, TVR, VM::VM1 >	178
gdcn::Attribute< Group, Element, TVR, VM::VM1_n >	185
gdcn::Attribute< Group, Element, TVR, VM::VM1_3 >	183
gdcn::Attribute< Group, Element, TVR, VM::VM1_8 >	184
gdcn::Attribute< Group, Element, TVR, VM::VM2_n >	191
gdcn::Attribute< Group, Element, TVR, VM::VM2_2n >	190
gdcn::Attribute< Group, Element, TVR, VM::VM3_n >	194
gdcn::Attribute< Group, Element, TVR, VM::VM3_3n >	193
gdcn::Base64	198
gdcn::network::BaseCompositeMessage	199
gdcn::network::CEchoRQ	236
gdcn::network::CEchoRSP	238
gdcn::network::CFindCancelRQ	240
gdcn::network::CFindRQ	241
gdcn::network::CFindRSP	242
gdcn::network::CMoveCancelRq	243
gdcn::network::CMoveRQ	245
gdcn::network::CMoveRSP	246
gdcn::network::CStoreRQ	280
gdcn::network::CStoreRSP	281
gdcn::network::BasePDU	201
gdcn::network::AAabortPDU	141
gdcn::network::AAssociateACPDU	143
gdcn::network::AAssociateRJPDU	146
gdcn::network::AAssociateRQPDU	147
gdcn::network::AReleaseRPPDU	165

gdcmm::network::AReleaseRQPDU . . . . .	166
gdcmm::network::PDataTFPDU . . . . .	564
std::basic_string< Char >	
std::string	
gdcmm::String< TDelimiter, TMaxLength, TPadChar > . . . . .	714
gdcmm::SegmentHelper::BasicCodedEntry . . . . .	207
gdcmm::BitmapToBitmapFilter . . . . .	220
gdcmm::PixmapToPixmapFilter . . . . .	591
gdcmm::ImageToImageFilter . . . . .	456
gdcmm::ImageApplyLookupTable . . . . .	423
gdcmm::ImageChangePhotometricInterpretation . . . . .	426
gdcmm::ImageChangePlanarConfiguration . . . . .	430
gdcmm::ImageChangeTransferSyntax . . . . .	433
gdcmm::ImageFragmentSplitter . . . . .	443
gdcmm::ByteBuffer . . . . .	225
gdcmm::ByteSwap< T > . . . . .	226
gdcmm::ByteSwapFilter . . . . .	227
gdcmm::network::CFind . . . . .	239
gdcmm::Coder . . . . .	248
gdcmm::Codec . . . . .	247
gdcmm::AudioCodec . . . . .	196
gdcmm::ImageCodec . . . . .	437
gdcmm::DeltaEncodingCodec . . . . .	311
gdcmm::JPEG2000Codec . . . . .	484
gdcmm::JPEGCodec . . . . .	490
gdcmm::JPEG12Codec . . . . .	480
gdcmm::JPEG16Codec . . . . .	482
gdcmm::JPEG8Codec . . . . .	487
gdcmm::JPEGLSCCodec . . . . .	494
gdcmm::KAKADUCCodec . . . . .	498
gdcmm::PGXCodec . . . . .	575
gdcmm::PNMCodec . . . . .	596
gdcmm::PVRGCodec . . . . .	618
gdcmm::RAWCodec . . . . .	632
gdcmm::RLECodec . . . . .	645
gdcmm::PDFCodec . . . . .	571
gdcmm::CodeString . . . . .	250
gdcmm::network::CompositeMessageFactory . . . . .	256
gdcmm::CompositeNetworkFunctions . . . . .	257
gdcmm::ConstCharWrapper . . . . .	261
gdcmm::CryptoFactory . . . . .	263
gdcmm::CAPICryptoFactory . . . . .	233
gdcmm::OpenSSLCryptoFactory . . . . .	545
gdcmm::OpenSSLP7CryptoFactory . . . . .	548
gdcmm::CryptographicMessageSyntax . . . . .	265
gdcmm::CAPICryptographicMessageSyntax . . . . .	234
gdcmm::OpenSSLCryptographicMessageSyntax . . . . .	546
gdcmm::OpenSSLP7CryptographicMessageSyntax . . . . .	550
gdcmm::CSAElement . . . . .	267
gdcmm::CSAHeader . . . . .	272
gdcmm::CSAHeaderDict . . . . .	276
gdcmm::CSAHeaderDictEntry . . . . .	277

gdcm::DataElement . . . . .	286
gdcm::CP246ExplicitDataElement . . . . .	261
gdcm::ExplicitDataElement . . . . .	367
gdcm::ExplicitImplicitDataElement . . . . .	368
gdcm::Fragment . . . . .	408
gdcm::BasicOffsetTable . . . . .	210
gdcm::ImplicitDataElement . . . . .	463
gdcm::Item . . . . .	475
gdcm::UNExplicitDataElement . . . . .	842
gdcm::UNExplicitImplicitDataElement . . . . .	843
gdcm::VR16ExplicitDataElement . . . . .	868
gdcm::DataSet . . . . .	297
gdcm::CommandDataSet . . . . .	254
gdcm::FileMetaInformation . . . . .	384
gdcm::DataSetHelper . . . . .	307
gdcm::Decoder . . . . .	307
gdcm::Codec . . . . .	247
gdcm::DefinedTerms . . . . .	309
gdcm::Defs . . . . .	309
gdcm::DICOMDIR . . . . .	313
gdcm::DICOMDIRGenerator . . . . .	314
gdcm::Dict . . . . .	316
gdcm::DictConverter . . . . .	318
gdcm::DictEntry . . . . .	321
gdcm::Dicts . . . . .	325
gdcm::network::DIMSE . . . . .	328
gdcm::DirectionCosines . . . . .	329
gdcm::Directory . . . . .	331
gdcm::DirectoryHelper . . . . .	334
gdcm::DummyValueGenerator . . . . .	335
gdcm::Element< TVR, TVM > . . . . .	338
gdcm::Element< TVR, VM::VM1_n > . . . . .	342
gdcm::Element< TVR, VM::VM1_2 > . . . . .	341
gdcm::Element< TVR, VM::VM2_n > . . . . .	347
gdcm::Element< TVR, VM::VM2_2n > . . . . .	345
gdcm::Element< TVR, VM::VM3_n > . . . . .	350
gdcm::Element< TVR, VM::VM3_3n > . . . . .	348
gdcm::Element< VR::AS, VM::VM5 > . . . . .	351
gdcm::Element< VR::OB, VM::VM1_n > . . . . .	338
gdcm::Element< VR::OB, VM::VM1 > . . . . .	352
gdcm::Element< VR::OW, VM::VM1_n > . . . . .	338
gdcm::Element< VR::OW, VM::VM1 > . . . . .	353
gdcm::ElementDisableCombinations< TVR, TVM > . . . . .	355
gdcm::ElementDisableCombinations< VR::OB, VM::VM1_n > . . . . .	356
gdcm::ElementDisableCombinations< VR::OW, VM::VM1_n > . . . . .	356
gdcm::EncapsulatedDocument . . . . .	356
gdcm::EncodingImplementation< T > . . . . .	357
gdcm::EncodingImplementation< VR::VRASCII > . . . . .	357
gdcm::EncodingImplementation< VR::VRBINARY > . . . . .	358
gdcm::EnumeratedValues . . . . .	360
gdcm::Event . . . . .	361
gdcm::AnyEvent . . . . .	161

gdcmm::AbortEvent . . . . .	151
gdcmm::AnonymizeEvent . . . . .	153
gdcmm::DataEvent . . . . .	295
gdcmm::DataSetEvent . . . . .	304
gdcmm::EndEvent . . . . .	359
gdcmm::ExitEvent . . . . .	365
gdcmm::FileNameEvent . . . . .	392
gdcmm::InitializeEvent . . . . .	465
gdcmm::IterationEvent . . . . .	478
gdcmm::ModifiedEvent . . . . .	527
gdcmm::ProgressEvent . . . . .	616
gdcmm::StartEvent . . . . .	704
gdcmm::UserEvent . . . . .	848
gdcmm::NoEvent . . . . .	542
std::exception	
gdcmm::CSAHeaderDictException . . . . .	279
gdcmm::DataElementException . . . . .	294
gdcmm::Exception . . . . .	364
gdcmm::ParseException . . . . .	560
gdcmm::Fiducials . . . . .	370
gdcmm::FileDerivation . . . . .	380
gdcmm::FileExplicitFilter . . . . .	382
gdcmm::Filename . . . . .	390
gdcmm::FilenameGenerator . . . . .	394
gdcmm::FileSet . . . . .	397
gdcmm::Global . . . . .	410
gdcmm::GroupDict . . . . .	412
gdcmm::IconImageFilter . . . . .	414
gdcmm::IconImageGenerator . . . . .	416
gdcmm::ignore_char . . . . .	419
gdcmm::ImageConverter . . . . .	442
gdcmm::ImageHelper . . . . .	446
gdcmm::network::ImplementationClassUIDSub . . . . .	461
gdcmm::network::ImplementationUIDSub . . . . .	461
gdcmm::network::ImplementationVersionNameSub . . . . .	462
gdcmm::IOD . . . . .	467
gdcmm::IODEntry . . . . .	468
gdcmm::IODs . . . . .	470
gdcmm::JSON . . . . .	497
gdcmm::Scanner::ltstr . . . . .	507
gdcmm::Macro . . . . .	508
gdcmm::Macros . . . . .	509
gdcmm::network::MaximumLengthSub . . . . .	511
gdcmm::MD5 . . . . .	512
gdcmm::MediaStorage . . . . .	513
gdcmm::Module . . . . .	529
gdcmm::ModuleEntry . . . . .	531
gdcmm::NestedModuleEntries . . . . .	539
gdcmm::Modules . . . . .	533
gdcmm::Object . . . . .	542
gdcmm::BaseRootQuery . . . . .	203
gdcmm::FindPatientRootQuery . . . . .	403
gdcmm::FindStudyRootQuery . . . . .	406

gdcmm::MovePatientRootQuery . . . . .	535
gdcmm::MoveStudyRootQuery . . . . .	537
gdcmm::Bitmap . . . . .	212
gdcmm::Pixmap . . . . .	584
gdcmm::Image . . . . .	419
gdcmm::Curve . . . . .	283
gdcmm::File . . . . .	371
gdcmm::FileWithName . . . . .	402
gdcmm::LookupTable . . . . .	503
gdcmm::SegmentedPaletteColorLookupTable . . . . .	660
gdcmm::MeshPrimitive . . . . .	524
gdcmm::Overlay . . . . .	554
gdcmm::Segment . . . . .	655
gdcmm::Subject . . . . .	721
gdcmm::Anonymizer . . . . .	156
gdcmm::Command . . . . .	252
gdcmm::MemberCommand< T > . . . . .	520
gdcmm::SimpleMemberCommand< T > . . . . .	687
gdcmm::FileAnonymizer . . . . .	374
gdcmm::FileChangeTransferSyntax . . . . .	377
gdcmm::FileStreamer . . . . .	398
gdcmm::network::ULConnectionManager . . . . .	835
gdcmm::Scanner . . . . .	649
gdcmm::ServiceClassUser . . . . .	681
gdcmm::Surface . . . . .	723
gdcmm::Value . . . . .	853
gdcmm::ByteValue . . . . .	228
gdcmm::SequenceOfFragments . . . . .	666
gdcmm::SequenceOfItems . . . . .	671
gdcmm::Orientation . . . . .	552
gdcmm::Parser . . . . .	562
gdcmm::Patient . . . . .	564
gdcmm::PDBelement . . . . .	567
gdcmm::PDBHeader . . . . .	569
gdcmm::network::PDUFactory . . . . .	572
gdcmm::PersonName . . . . .	574
gdcmm::PhotometricInterpretation . . . . .	577
gdcmm::PixelFormat . . . . .	579
gdcmm::Preamble . . . . .	599
gdcmm::PresentationContext . . . . .	600
gdcmm::network::PresentationContextAC . . . . .	602
gdcmm::PresentationContextGenerator . . . . .	603
gdcmm::network::PresentationContextRQ . . . . .	605
gdcmm::network::PresentationDataValue . . . . .	607
gdcmm::Printer . . . . .	609
gdcmm::DictPrinter . . . . .	323
gdcmm::Dumper . . . . .	336
gdcmm::PrivateDict . . . . .	612
gdcmm::PythonFilter . . . . .	620
gdcmm::QueryBase . . . . .	621
gdcmm::QueryImage . . . . .	624
gdcmm::QueryPatient . . . . .	626
gdcmm::QuerySeries . . . . .	628

gdcmm::QueryStudy . . . . .	630
gdcmm::QueryFactory . . . . .	623
gdcmm::Reader . . . . .	635
gdcmm::PixmapReader . . . . .	587
gdcmm::ImageReader . . . . .	449
gdcmm::ImageRegionReader . . . . .	453
gdcmm::SegmentReader . . . . .	661
gdcmm::SurfaceReader . . . . .	732
gdcmm::Region . . . . .	640
gdcmm::BoxRegion . . . . .	222
gdcmm::Rescaler . . . . .	642
gdcmm::network::RoleSelectionSub . . . . .	648
gdcmm::SerieHelper::Rule . . . . .	648
gdcmm::SerieHelper . . . . .	677
gdcmm::Series . . . . .	679
gdcmm::network::ServiceClassApplicationInformation . . . . .	680
gdcmm::SHA1 . . . . .	686
gdcmm::SimpleSubjectWatcher . . . . .	690
gdcmm::SmartPointer< ObjectType > . . . . .	692
gdcmm::SmartPointer< gdcmm::Bitmap > . . . . .	692
gdcmm::SmartPointer< gdcmm::File > . . . . .	692
gdcmm::SmartPointer< gdcmm::gdcmm::Subject > . . . . .	692
gdcmm::SmartPointer< gdcmm::Image > . . . . .	692
gdcmm::SmartPointer< gdcmm::MemberCommand > . . . . .	692
gdcmm::SmartPointer< gdcmm::MeshPrimitive > . . . . .	692
gdcmm::SmartPointer< gdcmm::Pixmap > . . . . .	692
gdcmm::SmartPointer< gdcmm::SimpleMemberCommand > . . . . .	692
gdcmm::SmartPointer< LookupTable > . . . . .	692
gdcmm::SmartPointer< Segment > . . . . .	692
gdcmm::SmartPointer< Surface > . . . . .	692
gdcmm::SmartPointer< Value > . . . . .	692
gdcmm::network::SOPClassExtendedNegociationSub . . . . .	695
gdcmm::SOPClassUIDToIOD . . . . .	696
gdcmm::Sorter . . . . .	697
gdcmm::IPPSorter . . . . .	471
gdcmm::Spacing . . . . .	701
gdcmm::Spectroscopy . . . . .	703
gdcmm::SplitMosaicFilter . . . . .	703
gdcmm::static_assert_test< x > . . . . .	706
gdcmm::STATIC_ASSERTION_FAILURE< x > . . . . .	706
gdcmm::STATIC_ASSERTION_FAILURE< true > . . . . .	706
gdcmm::StreamImageReader . . . . .	706
gdcmm::StreamImageWriter . . . . .	709
String< '\', 64 > . . . . .	
gdcmm::LO . . . . .	501
gdcmm::StringFilter . . . . .	718
gdcmm::Study . . . . .	720
gdcmm::SurfaceHelper . . . . .	730
gdcmm::SwapCode . . . . .	736
gdcmm::SwapperDoOp . . . . .	738
gdcmm::SwapperNoOp . . . . .	739
gdcmm::System . . . . .	739
gdcmm::Table . . . . .	744

gdcm::TableEntry . . . . .	745
gdcm::TableReader . . . . .	745
gdcm::XMLDictReader . . . . .	934
gdcm::XMLPrivateDictReader . . . . .	938
gdcm::network::TableRow . . . . .	747
gdcm::Tag . . . . .	749
gdcm::PrivateTag . . . . .	614
gdcm::TagPath . . . . .	756
gdcm::Testing . . . . .	757
gdcm::Trace . . . . .	761
gdcm::TransferSyntax . . . . .	764
gdcm::network::TransferSyntaxSub . . . . .	768
gdcm::network::Transition . . . . .	769
gdcm::Type . . . . .	770
gdcm::UI . . . . .	772
gdcm::UIDGenerator . . . . .	773
gdcm::UIDs . . . . .	774
gdcm::network::ULAction . . . . .	793
gdcm::network::ULActionAA1 . . . . .	796
gdcm::network::ULActionAA2 . . . . .	797
gdcm::network::ULActionAA3 . . . . .	798
gdcm::network::ULActionAA4 . . . . .	799
gdcm::network::ULActionAA5 . . . . .	800
gdcm::network::ULActionAA6 . . . . .	801
gdcm::network::ULActionAA7 . . . . .	803
gdcm::network::ULActionAA8 . . . . .	804
gdcm::network::ULActionAE1 . . . . .	805
gdcm::network::ULActionAE2 . . . . .	806
gdcm::network::ULActionAE3 . . . . .	807
gdcm::network::ULActionAE4 . . . . .	808
gdcm::network::ULActionAE5 . . . . .	810
gdcm::network::ULActionAE6 . . . . .	811
gdcm::network::ULActionAE7 . . . . .	812
gdcm::network::ULActionAE8 . . . . .	813
gdcm::network::ULActionAR1 . . . . .	814
gdcm::network::ULActionAR10 . . . . .	815
gdcm::network::ULActionAR2 . . . . .	817
gdcm::network::ULActionAR3 . . . . .	818
gdcm::network::ULActionAR4 . . . . .	819
gdcm::network::ULActionAR5 . . . . .	820
gdcm::network::ULActionAR6 . . . . .	821
gdcm::network::ULActionAR7 . . . . .	822
gdcm::network::ULActionAR8 . . . . .	824
gdcm::network::ULActionAR9 . . . . .	825
gdcm::network::ULActionDT1 . . . . .	826
gdcm::network::ULActionDT2 . . . . .	827
gdcm::network::ULConnection . . . . .	830
gdcm::network::ULConnectionCallback . . . . .	832
gdcm::network::ULBasicCallback . . . . .	828
gdcm::network::ULWritingCallback . . . . .	840
gdcm::network::ULConnectionInfo . . . . .	834
gdcm::network::ULEvent . . . . .	838
gdcm::network::ULTransitionTable . . . . .	839

gdcm::Unpacker12Bits	845
gdcm::Usage	846
gdcm::network::UserInformation	849
gdcm::UUIDGenerator	850
gdcm::Validate	851
gdcm::ValueIO< TDE, TSwap, TType >	855
gdcm::Version	856
gdcm::VL	857
gdcm::VM	859
gdcm::VMToLength< T >	863
gdcm::VR	863
gdcm::VRToEncoding< T >	870
gdcm::VRToType< T >	870
gdcm::VRToType< TVR >	870
gdcm::VRVLSIZE< T >	871
gdcm::VRVLSIZE< 0 >	871
gdcm::VRVLSIZE< 1 >	871
vtkImageAlgorithm	
vtkImagePlanarComponentsToComponents	916
vtkImageMapToColors	
vtkImageMapToWindowLevelColors2	914
vtkImageWriter	
vtkGDCMImageWriter	883
vtkLookupTable	
vtkLookupTable16	921
vtkMedicalImageProperties	
vtkGDCMMedicalImageProperties	888
vtkMedicalImageReader2	
vtkGDCMImageReader	872
vtkGDCMThreadedImageReader	898
vtkGDCMImageReader2	878
vtkObject	
vtkGDCMTesting	896
vtkImageColorViewer	904
vtkRTStructSetProperties	923
vtkPolyDataAlgorithm	
vtkGDCMPolyDataReader	890
vtkPolyDataWriter	
vtkGDCMPolyDataWriter	893
vtkThreadedImageAlgorithm	
vtkGDCMThreadedImageReader2	901
vtkImageMapToColors16	911
vtkImageRGBToYBR	918
vtkImageYBRToRGB	920
gdcm::Waveform	928
gdcm::Writer	928
gdcm::PixmapWriter	593
gdcm::ImageWriter	458
gdcm::SegmentWriter	664
gdcm::SurfaceWriter	734
gdcm::XMLPrinter	935



## Chapter 24

# Class Index

### 24.1 Class List

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

<a href="#">gdcmm::network::AAAbortPDU</a>	
<a href="#">AAAbortPDU Table</a> 9-26 A-ABORT PDU FIELDS . . . . .	141
<a href="#">gdcmm::network::AAssociateACPDU</a>	
<a href="#">AAssociateACPDU Table</a> 9-17 ASSOCIATE-AC PDU fields . . . . .	143
<a href="#">gdcmm::network::AAssociateRJPDU</a>	
<a href="#">AAssociateRJPDU Table</a> 9-21 ASSOCIATE-RJ PDU FIELDS . . . . .	146
<a href="#">gdcmm::network::AAssociateRQPDU</a>	
<a href="#">AAssociateRQPDU Table</a> 9-11 ASSOCIATE-RQ PDU fields . . . . .	147
<a href="#">gdcmm::AbortEvent</a> . . . . .	151
<a href="#">gdcmm::network::AbstractSyntax</a>	
<a href="#">AbstractSyntax Table</a> 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS . . . . .	152
<a href="#">gdcmm::AnonymizeEvent</a>	
<a href="#">AnonymizeEvent</a> Special type of event triggered during the Anonymization process . . . . .	153
<a href="#">gdcmm::Anonymizer</a>	
<a href="#">Anonymizer</a> This class is a multi purpose anonymizer. It can work in 2 mode: . . . . .	156
<a href="#">gdcmm::AnyEvent</a> . . . . .	161
<a href="#">gdcmm::network::ApplicationContext</a>	
<a href="#">ApplicationContext Table</a> 9-12 APPLICATION CONTEXT ITEM FIELDS . . . . .	162
<a href="#">gdcmm::ApplicationEntity</a>	
<a href="#">ApplicationEntity</a> . . . . .	163
<a href="#">gdcmm::network::AReleaseRPPDU</a>	
<a href="#">AReleaseRPPDU Table</a> 9-25 A-RELEASE-RP PDU fields . . . . .	165
<a href="#">gdcmm::network::AReleaseRQPDU</a>	
<a href="#">AReleaseRQPDU Table</a> 9-24 A-RELEASE-RQ PDU FIELDS . . . . .	166
<a href="#">gdcmm::network::ARTIMTimer</a>	
<a href="#">ARTIMTimer</a> This file contains the code for the ARTIM timer . . . . .	168
<a href="#">gdcmm::ASN1</a>	
Class for <a href="#">ASN1</a> . . . . .	169
<a href="#">gdcmm::network::AsynchronousOperationsWindowSub</a>	
<a href="#">AsynchronousOperationsWindowSub</a> PS 3.7 <a href="#">Table</a> D.3-7 ASYNCHRONOUS OPERATIONS WIN↔ DOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ) . . . . .	170
<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 instanciation does not match the public dictionary . . . . .	171

gdcm::Attribute< Group, Element, TVR, VM::VM1 >	178
gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >	183
gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >	184
gdcm::Attribute< Group, Element, TVR, VM::VM1_n >	185
gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >	190
gdcm::Attribute< Group, Element, TVR, VM::VM2_n >	191
gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >	193
gdcm::Attribute< Group, Element, TVR, VM::VM3_n >	194
gdcm::AudioCodec	
AudioCodec	196
gdcm::Base64	
Class for Base64	198
gdcm::network::BaseCompositeMessage	
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	199
gdcm::network::BasePDU	
BasePDU base class for PDUs	201
gdcm::BaseRootQuery	
BaseRootQuery contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root	203
gdcm::SegmentHelper::BasicCodedEntry	
This structure defines a basic coded entry with all of its attributes	207
gdcm::BasicOffsetTable	
Class to represent a BasicOffsetTable	210
gdcm::Bitmap	
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)	212
gdcm::BitmapToBitmapFilter	
BitmapToBitmapFilter class Super class for all filter taking an image and producing an output image	220
gdcm::BoxRegion	
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)	222
gdcm::ByteBuffer	
ByteBuffer	225
gdcm::ByteSwap< T >	
ByteSwap	226
gdcm::ByteSwapFilter	
ByteSwapFilter In place byte-swapping of a dataset FIXME: FL status ??	227
gdcm::ByteValue	
Class to represent binary value (array of bytes)	228
gdcm::CAPICryptoFactory	233
gdcm::CAPICryptographicMessageSyntax	234
gdcm::network::CEchoRQ	
CEchoRQ this file defines the messages for the cecho action	236
gdcm::network::CEchoRSP	
CEchoRSP this file defines the messages for the cecho action	238
gdcm::network::CFind	239
gdcm::network::CFindCancelRQ	
CFindCancelRQ this file defines the messages for the cfind action	240
gdcm::network::CFindRQ	
CFindRQ this file defines the messages for the cfind action	241
gdcm::network::CFindRSP	
CFindRSP this file defines the messages for the cfind action	242

<a href="#">gdcm::network::CMoveCancelRq</a>	243
<a href="#">gdcm::network::CMoveRQ</a>	
CMoveRQ this file defines the messages for the cmove action	245
<a href="#">gdcm::network::CMoveRSP</a>	
CMoveRSP this file defines the messages for the cmove action	246
<a href="#">gdcm::Codec</a>	
Codec class	247
<a href="#">gdcm::Coder</a>	
Coder	248
<a href="#">gdcm::CodeString</a>	
CodeString This is an implementation of DICOM VR: CS The cstor will properly Trim so that operator== is correct	250
<a href="#">gdcm::Command</a>	
Command superclass for callback/observer methods	252
<a href="#">gdcm::CommandDataSet</a>	
Class to represent a <a href="#">Command DataSet</a>	254
<a href="#">gdcm::network::CompositeMessageFactory</a>	
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)	256
<a href="#">gdcm::CompositeNetworkFunctions</a>	
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:	257
<a href="#">gdcm::ConstCharWrapper</a>	
Do not use me	261
<a href="#">gdcm::CP246ExplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as CP246Explicit Data <a href="#">Element</a>	261
<a href="#">gdcm::CryptoFactory</a>	
Class to do handle the crypto factory	263
<a href="#">gdcm::CryptographicMessageSyntax</a>	265
<a href="#">gdcm::CSAElement</a>	
Class to represent a CSA <a href="#">Element</a>	267
<a href="#">gdcm::CSAHeader</a>	
Class for <a href="#">CSAHeader</a>	272
<a href="#">gdcm::CSAHeaderDict</a>	
Class to represent a map of <a href="#">CSAHeaderDictEntry</a>	276
<a href="#">gdcm::CSAHeaderDictEntry</a>	
Class to represent an Entry in the <a href="#">Dict</a> Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from <a href="#">gdcm::Tag</a> to the needed information	277
<a href="#">gdcm::CSAHeaderDictException</a>	279
<a href="#">gdcm::network::CStoreRQ</a>	
CStoreRQ this file defines the messages for the cecho action	280
<a href="#">gdcm::network::CStoreRSP</a>	
CStoreRSP this file defines the messages for the cecho action	281

<a href="#">gdcmm::Curve</a>	
<a href="#">Curve</a> class to handle element 50xx,3000 <a href="#">Curve</a> Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004	283
<a href="#">gdcmm::DataElement</a>	
Class to represent a Data <a href="#">Element</a> either Implicit or Explicit	286
<a href="#">gdcmm::DataElementException</a>	294
<a href="#">gdcmm::DataEvent</a>	
<a href="#">DataEvent</a>	295
<a href="#">gdcmm::DataSet</a>	
Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information <a href="#">Object</a>	297
<a href="#">gdcmm::DataSetEvent</a>	
<a href="#">DataSetEvent</a> Special type of event triggered during the <a href="#">DataSet</a> store/move process	304
<a href="#">gdcmm::DataSetHelper</a>	
<a href="#">DataSetHelper</a> (internal class, not intended for user level)	307
<a href="#">gdcmm::Decoder</a>	
<a href="#">Decoder</a>	307
<a href="#">gdcmm::DefinedTerms</a>	
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 <a href="#">Element</a> with Defined Terms that does not contain a <a href="#">Value</a> equivalent to one of the Values currently specified in this standard shall not be considered to have an invalid value. Note: Interpretation <a href="#">Type</a> ID (4008,0210) is an example of a Data <a href="#">Element</a> having Defined Terms. It is defined to have a <a href="#">Value</a> that may be one of the set of standard Values; REPORT or AMENDMENT (see PS 3.3). Because this Data <a href="#">Element</a> has Defined Terms other Interpretation <a href="#">Type</a> IDs may be defined by the implementor	309
<a href="#">gdcmm::Defs</a>	
FIXME I do not like the name ' <a href="#">Defs</a> '	309
<a href="#">gdcmm::DeltaEncodingCodec</a>	
<a href="#">DeltaEncodingCodec</a> compression used by some private vendor	311
<a href="#">gdcmm::DICOMDIR</a>	
<a href="#">DICOMDIR</a> class	313
<a href="#">gdcmm::DICOMDIRGenerator</a>	
<a href="#">DICOMDIRGenerator</a> class This is a STD-GEN-CD <a href="#">DICOMDIR</a> generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles	314
<a href="#">gdcmm::Dict</a>	
Class to represent a map of <a href="#">DictEntry</a>	316
<a href="#">gdcmm::DictConverter</a>	
Class to convert a .dic file into something else:	318
<a href="#">gdcmm::DictEntry</a>	
Class to represent an Entry in the <a href="#">Dict</a> Does not really exist within the DICOM definition, just a way to minimize storage and have a mapping from <a href="#">gdcmm::Tag</a> to the needed information	321
<a href="#">gdcmm::DictPrinter</a>	
<a href="#">DictPrinter</a> class	323
<a href="#">gdcmm::Dicts</a>	
Class to manipulate the sum of knowledge (all the dict user load)	325
<a href="#">gdcmm::network::DIMSE</a>	
<a href="#">DIMSE</a> PS 3.7 - 2009 Annex E <a href="#">Command</a> Dictionary (Normative) E.1 REGISTRY OF DICOM CO↔MMAND ELEMENTS <a href="#">Table</a> E.1-1 COMMAND FIELDS (PART 1)	328
<a href="#">gdcmm::DirectionCosines</a>	
Class to handle <a href="#">DirectionCosines</a>	329
<a href="#">gdcmm::Directory</a>	
Class for manipulation directories	331

<a href="#">gdcm::DirectoryHelper</a>	
<a href="#">DirectoryHelper</a>	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 . . . . .
	334
<a href="#">gdcm::DummyValueGenerator</a>	
Class for generating dummy value . . . . .	335
<a href="#">gdcm::Dumper</a>	
Codec class . . . . .	336
<a href="#">gdcm::Element&lt; TVR, TVM &gt;</a>	
Element class . . . . .	338
<a href="#">gdcm::Element&lt; TVR, VM::VM1_2 &gt;</a>	341
<a href="#">gdcm::Element&lt; TVR, VM::VM1_n &gt;</a>	342
<a href="#">gdcm::Element&lt; TVR, VM::VM2_2n &gt;</a>	345
<a href="#">gdcm::Element&lt; TVR, VM::VM2_n &gt;</a>	347
<a href="#">gdcm::Element&lt; TVR, VM::VM3_3n &gt;</a>	348
<a href="#">gdcm::Element&lt; TVR, VM::VM3_n &gt;</a>	350
<a href="#">gdcm::Element&lt; VR::AS, VM::VM5 &gt;</a>	351
<a href="#">gdcm::Element&lt; VR::OB, VM::VM1 &gt;</a>	352
<a href="#">gdcm::Element&lt; VR::OW, VM::VM1 &gt;</a>	353
<a href="#">gdcm::ElementDisableCombinations&lt; TVR, TVM &gt;</a>	
A class which is used to produce compile errors for an invalid combination of template parameters . . . . .	355
<a href="#">gdcm::ElementDisableCombinations&lt; VR::OB, VM::VM1_n &gt;</a>	356
<a href="#">gdcm::ElementDisableCombinations&lt; VR::OW, VM::VM1_n &gt;</a>	356
<a href="#">gdcm::EncapsulatedDocument</a>	
EncapsulatedDocument . . . . .	356
<a href="#">gdcm::EncodingImplementation&lt; T &gt;</a>	
EncodingImplementation . . . . .	357
<a href="#">gdcm::EncodingImplementation&lt; VR::VRASCII &gt;</a>	357
<a href="#">gdcm::EncodingImplementation&lt; VR::VRBINARY &gt;</a>	358
<a href="#">gdcm::EndEvent</a>	359
<a href="#">gdcm::EnumeratedValues</a>	
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: . . . . .	360
<a href="#">gdcm::Event</a>	
Superclass for callback/observer methods . . . . .	361
<a href="#">gdcm::Exception</a>	
Exception . . . . .	364
<a href="#">gdcm::ExitEvent</a>	365
<a href="#">gdcm::ExplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a> . . . . .	367
<a href="#">gdcm::ExplicitImplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a> . . . . .	368
<a href="#">gdcm::Fiducials</a>	
Fiducials . . . . .	370
<a href="#">gdcm::File</a>	
DICOM <a href="#">File</a> See PS 3.10 <a href="#">File</a> : A <a href="#">File</a> 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 <a href="#">File</a> . Files are identified by a unique <a href="#">File</a> ID and may be written, read and/or deleted . . . . .	371
<a href="#">gdcm::FileAnonymizer</a>	
FileAnonymizer . . . . .	374

<a href="#">gdcm::FileChangeTransferSyntax</a>	
<a href="#">FileChangeTransferSyntax</a> . . . . .	377
<a href="#">gdcm::FileDerivation</a>	
<a href="#">FileDerivation</a> class See PS 3.16 - 2008 For the list of Code <a href="#">Value</a> that can be used for in Derivation	
Code Sequence . . . . .	380
<a href="#">gdcm::FileExplicitFilter</a>	
<a href="#">FileExplicitFilter</a> class After changing a file from Implicit to Explicit representation (see <a href="#">Image↔ChangeTransferSyntax</a> ) one operation is to make sure the <a href="#">VR</a> 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 <a href="#">VR</a> is not stored directly in the file . . . . .	382
<a href="#">gdcm::FileMetaInformation</a>	
Class to represent a <a href="#">File</a> Meta Information . . . . .	384
<a href="#">gdcm::Filename</a>	
Class to manipulate file name's . . . . .	390
<a href="#">gdcm::FileNameEvent</a>	
<a href="#">FileNameEvent</a> Special type of event triggered during processing of <a href="#">FileSet</a> . . . . .	392
<a href="#">gdcm::FilenameGenerator</a>	
<a href="#">FilenameGenerator</a> . . . . .	394
<a href="#">gdcm::FileSet</a>	
File-set: A File-set is a collection of DICOM Files (and possibly non-DICOM Files) that share a common naming space within which <a href="#">File</a> IDs are unique . . . . .	397
<a href="#">gdcm::FileStreamer</a>	
<a href="#">FileStreamer</a> This class let a user create a massive DICOM <a href="#">DataSet</a> from a template DICOM file, by appending chunks of data . . . . .	398
<a href="#">gdcm::FileWithName</a>	
<a href="#">FileWithName</a> . . . . .	402
<a href="#">gdcm::FindPatientRootQuery</a>	
<a href="#">PatientRootQuery</a> contains: the class which will produce a dataset for c-find with patient root . . . .	403
<a href="#">gdcm::FindStudyRootQuery</a>	
<a href="#">FindStudyRootQuery</a> contains: the class which will produce a dataset for C-FIND with study root . .	406
<a href="#">gdcm::Fragment</a>	
Class to represent a <a href="#">Fragment</a> . . . . .	408
<a href="#">gdcm::Global</a>	
<a href="#">Global</a> . . . . .	410
<a href="#">gdcm::GroupDict</a>	
Class to represent the mapping from group number to its abbreviation and name . . . . .	412
<a href="#">gdcm::IconImageFilter</a>	
<a href="#">IconImageFilter</a> This filter will extract icons from a <a href="#">gdcm::File</a> This filter will loop over all known sequence (public and private) that may contains an <a href="#">IconImage</a> 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	414
<a href="#">gdcm::IconImageGenerator</a>	
<a href="#">IconImageGenerator</a> This filter will generate a valid Icon from the Pixel Data element (an instance of <a href="#">gdcm::Pixmap</a> ). To generate a valid Icon, one is only allowed the following Photometric Interpretation: . . . . .	416
<a href="#">gdcm::ignore_char</a> . . . . .	419
<a href="#">gdcm::Image</a>	
<a href="#">Image</a> This is the container for an <a href="#">Image</a> in the general sense. From this container you should be able to request information like: . . . . .	419
<a href="#">gdcm::ImageApplyLookupTable</a>	
<a href="#">ImageApplyLookupTable</a> class It applies the LUT the PixelData (only PALETTE_COLOR images) Output will be a <a href="#">PhotometricInterpretation</a> =RGB image . . . . .	423

gdcm::ImageChangePhotometricInterpretation	
ImageChangePhotometricInterpretation class Class to change the Photometric Interpretation of an input DICOM	426
gdcm::ImageChangePlanarConfiguration	
ImageChangePlanarConfiguration class Class to change the Planar configuration of an input DICOM	
By default it will change into the more usual representation: PlanarConfiguration = 0	430
gdcm::ImageChangeTransferSyntax	
ImageChangeTransferSyntax class Class to change the transfer syntax of an input DICOM	433
gdcm::ImageCodec	
ImageCodec	437
gdcm::ImageConverter	
Image Converter	442
gdcm::ImageFragmentSplitter	
ImageFragmentSplitter class For single frame image, DICOM standard allow splitting the frame into multiple fragments	443
gdcm::ImageHelper	
ImageHelper (internal class, not intended for user level)	446
gdcm::ImageReader	
ImageReader	449
gdcm::ImageRegionReader	
ImageRegionReader	453
gdcm::ImageToImageFilter	
ImageToImageFilter class Super class for all filter taking an image and producing an output image	456
gdcm::ImageWriter	
ImageWriter	458
gdcm::network::ImplementationClassUIDSub	
ImplementationClassUIDSub PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELD (A-ASSOCIATE-RQ)	461
gdcm::network::ImplementationUIDSub	
ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	461
gdcm::network::ImplementationVersionNameSub	
ImplementationVersionNameSub Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)	462
gdcm::ImplicitDataElement	
Class to represent an <i>Implicit VR</i> Data Element	463
gdcm::InitializeEvent	465
gdcm::IOD	
Class for representing a IOD	467
gdcm::IODEntry	
Class for representing a IODEntry	468
gdcm::IODs	
Class for representing a IODs	470
gdcm::IPPSorter	
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	471
gdcm::Item	
Class to represent an Item A component of the value of a Data Element that is of Value Representation Sequence of Items. An Item contains a Data Set. See PS 3.5 7.5.1 Item Encoding Rules Each Item of a Data Element of VR SQ shall be encoded as a DICOM Standard Data Element with a specific Data Element Tag of Value (FFFE,E000). The Item Tag is followed by a 4 byte Item Length field encoded in one of the following two ways Explicit/ Implicit	475
gdcm::IterationEvent	478

<a href="#">gdcm::JPEG12Codec</a>	
Class to do JPEG 12bits (lossy & lossless)	480
<a href="#">gdcm::JPEG16Codec</a>	
Class to do JPEG 16bits (lossless)	482
<a href="#">gdcm::JPEG2000Codec</a>	
Class to do JPEG 2000	484
<a href="#">gdcm::JPEG8Codec</a>	
Class to do JPEG 8bits (lossy & lossless)	487
<a href="#">gdcm::JPEGCodec</a>	
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: <a href="#">gdcm::JPEG8Codec</a> , <a href="#">gdcm::JPEG12Codec</a> & <a href="#">gdcm::JPEG16Codec</a>	
It also support inconsistency in between DICOM header and JPEG compressed stream <a href="#">ImageCodec</a> implementation for the JPEG case	490
<a href="#">gdcm::JPEGLSCodec</a>	
JPEG-LS	494
<a href="#">gdcm::JSON</a>	497
<a href="#">gdcm::KAKADUCodec</a>	
KAKADUCodec	498
<a href="#">gdcm::LO</a>	
LO	501
<a href="#">gdcm::LookupTable</a>	
LookupTable class	503
<a href="#">gdcm::Scanner::ltstr</a>	507
<a href="#">gdcm::Macro</a>	
Class for representing a <a href="#">Macro</a>	508
<a href="#">gdcm::Macros</a>	
Class for representing a <a href="#">Modules</a>	509
<a href="#">gdcm::network::MaximumLengthSub</a>	
MaximumLengthSub Annex D <a href="#">Table D.1-1</a> MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATION↵ TE-RQ)	511
<a href="#">gdcm::MD5</a>	
Class for MD5	512
<a href="#">gdcm::MediaStorage</a>	
MediaStorage	513
<a href="#">gdcm::MemberCommand&lt; T &gt;</a>	
Command subclass that calls a pointer to a member function	520
<a href="#">gdcm::MeshPrimitive</a>	
This class defines surface mesh primitives. It is designed from surface mesh primitives macro	524
<a href="#">gdcm::ModifiedEvent</a>	527
<a href="#">gdcm::Module</a>	
Class for representing a <a href="#">Module</a>	529
<a href="#">gdcm::ModuleEntry</a>	
Class for representing a <a href="#">ModuleEntry</a>	531
<a href="#">gdcm::Modules</a>	
Class for representing a <a href="#">Modules</a>	533
<a href="#">gdcm::MovePatientRootQuery</a>	
MovePatientRootQuery contains: the class which will produce a dataset for c-move with patient root	535
<a href="#">gdcm::MoveStudyRootQuery</a>	
MoveStudyRootQuery contains: the class which will produce a dataset for C-MOVE with study root	537
<a href="#">gdcm::NestedModuleEntries</a>	
Class for representing a <a href="#">NestedModuleEntries</a>	539
<a href="#">gdcm::NoEvent</a>	542
<a href="#">gdcm::Object</a>	
Object	542



<a href="#">gdcm::OpenSSLCryptoFactory</a>	545
<a href="#">gdcm::OpenSSLCryptographicMessageSyntax</a>	546
<a href="#">gdcm::OpenSSLP7CryptoFactory</a>	548
<a href="#">gdcm::OpenSSLP7CryptographicMessageSyntax</a>	
Class for <a href="#">CryptographicMessageSyntax</a> encryption. This is just a simple wrapper around openssl PKCS7_encrypt functionalities	550
<a href="#">gdcm::Orientation</a>	
Class to handle <a href="#">Orientation</a>	552
<a href="#">gdcm::Overlay</a>	
Overlay class	554
<a href="#">gdcm::ParseException</a>	
<a href="#">ParseException</a> Standard exception handling object	560
<a href="#">gdcm::Parser</a>	
Parser ala XML_Parser from expat (SAX)	562
<a href="#">gdcm::Patient</a>	
See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54	564
<a href="#">gdcm::network::PDataTFPDU</a>	
<a href="#">PDataTFPDU</a> Table 9-22 P-DATA-TF PDU FIELDS	564
<a href="#">gdcm::PDBElement</a>	
Class to represent a PDB <a href="#">Element</a>	567
<a href="#">gdcm::PDBHeader</a>	
Class for <a href="#">PDBHeader</a>	569
<a href="#">gdcm::PDFCodec</a>	
<a href="#">PDFCodec</a> class	571
<a href="#">gdcm::network::PDUFactory</a>	
<a href="#">PDUFactory</a> 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	572
<a href="#">gdcm::PersonName</a>	
<a href="#">PersonName</a> class	574
<a href="#">gdcm::PGXCodec</a>	
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images	575
<a href="#">gdcm::PhotometricInterpretation</a>	
Class to represent an <a href="#">PhotometricInterpretation</a>	577
<a href="#">gdcm::PixelFormat</a>	
<a href="#">PixelFormat</a>	579
<a href="#">gdcm::Pixmap</a>	
<a href="#">Pixmap</a> class A bitmap based image. Used as parent for both <a href="#">IconImage</a> and the main <a href="#">Pixel Data Image</a> It does not contains any World Space information (IPP, IOP)	584
<a href="#">gdcm::PixmapReader</a>	
<a href="#">PixmapReader</a>	587
<a href="#">gdcm::PixmapToPixmapFilter</a>	
<a href="#">PixmapToPixmapFilter</a> class Super class for all filter taking an image and producing an output image	591
<a href="#">gdcm::PixmapWriter</a>	
<a href="#">PixmapWriter</a> This class will takes two inputs:	593
<a href="#">gdcm::PNMCodec</a>	
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <a href="http://netpbm.sourceforge.net/">http://netpbm.sourceforge.net/</a>	596
<a href="#">gdcm::Preamble</a>	
DICOM <a href="#">Preamble</a> (Part 10)	599
<a href="#">gdcm::PresentationContext</a>	
<a href="#">PresentationContext</a>	600
<a href="#">gdcm::network::PresentationContextAC</a>	
<a href="#">PresentationContextAC</a> Table 9-18 PRESENTATION CONTEXT ITEM FIELDS	602

<a href="#">gdcm::PresentationContextGenerator</a>	
<a href="#">PresentationContextGenerator</a>	This class is responsible for generating the proper <a href="#">PresentationContext</a> 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 . . . . .
	603
<a href="#">gdcm::network::PresentationContextRQ</a>	
<a href="#">PresentationContextRQ</a>	<a href="#">Table 9-13 PRESENTATION CONTEXT ITEM FIELDS</a> . . . . .
	605
<a href="#">gdcm::network::PresentationDataValue</a>	
<a href="#">PresentationDataValue</a>	<a href="#">Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS</a> . . . . .
	607
<a href="#">gdcm::Printer</a>	
<a href="#">Printer</a>	class . . . . .
	609
<a href="#">gdcm::PrivateDict</a>	
<a href="#">Private</a>	<a href="#">Dict</a> . . . . .
	612
<a href="#">gdcm::PrivateTag</a>	
	Class to represent a Private DICOM Data <a href="#">Element</a> ( <a href="#">Attribute</a> ) <a href="#">Tag</a> (Group, <a href="#">Element</a> , Owner) . . . . .
	614
<a href="#">gdcm::ProgressEvent</a>	
<a href="#">ProgressEvent</a>	Special type of event triggered during . . . . .
	616
<a href="#">gdcm::PVRGCodec</a>	
<a href="#">PVRGCodec</a>	. . . . .
	618
<a href="#">gdcm::PythonFilter</a>	
<a href="#">PythonFilter</a>	<a href="#">PythonFilter</a> is the class that make gdcm2.x looks more like gdcm1 and transform the binary blob contained in a <a href="#">DataElement</a> into a string, typically this is a nice feature to have for wrapped language . . . . .
	620
<a href="#">gdcm::QueryBase</a>	
<a href="#">QueryBase</a>	contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE . . . . .
	621
<a href="#">gdcm::QueryFactory</a>	
<a href="#">QueryFactory.h</a>	. . . . .
	623
<a href="#">gdcm::QueryImage</a>	
<a href="#">QueryImage</a>	contains: class to construct an image-based query for C-FIND and C-MOVE . . . . .
	624
<a href="#">gdcm::QueryPatient</a>	
<a href="#">QueryPatient</a>	contains: class to construct a patient-based query for c-find and c-move . . . . .
	626
<a href="#">gdcm::QuerySeries</a>	
<a href="#">QuerySeries</a>	contains: class to construct a series-based query for c-find and c-move . . . . .
	628
<a href="#">gdcm::QueryStudy</a>	
<a href="#">QueryStudy.h</a>	contains: class to construct a study-based query for C-FIND and C-MOVE . . . . .
	630
<a href="#">gdcm::RAWCodec</a>	
<a href="#">RAWCodec</a>	class . . . . .
	632
<a href="#">gdcm::Reader</a>	
<a href="#">Reader</a>	ala DOM (Document <a href="#">Object</a> Model) . . . . .
	635
<a href="#">gdcm::Region</a>	
	Class for manipulation region . . . . .
	640
<a href="#">gdcm::Rescaler</a>	
<a href="#">Rescale</a>	class This class is meant to apply the linear transform of Stored Pixel <a href="#">Value</a> to Real World <a href="#">Value</a> . 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 <a href="#">Type</a> 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 <a href="#">Value</a> will be 16 bits signed type . . . . .
	642
<a href="#">gdcm::RLECodec</a>	
	Class to do RLE . . . . .
	645

<a href="#">gdcm::network::RoleSelectionSub</a>	
<a href="#">RoleSelectionSub</a> PS 3.7 <a href="#">Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)</a>	648
<a href="#">gdcm::SerieHelper::Rule</a>	648
<a href="#">gdcm::Scanner</a>	
<a href="#">Scanner</a> This filter is meant for quickly browsing a <a href="#">FileSet</a> (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 <a href="#">Attribute</a>	649
<a href="#">gdcm::Segment</a>	
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface	655
<a href="#">gdcm::SegmentedPaletteColorLookupTable</a>	
<a href="#">SegmentedPaletteColorLookupTable</a> class	660
<a href="#">gdcm::SegmentReader</a>	
This class defines a segment reader. It reads attributes of group 0x0062	661
<a href="#">gdcm::SegmentWriter</a>	
This class defines a segment writer. It writes attributes of group 0x0062	664
<a href="#">gdcm::SequenceOfFragments</a>	
Class to represent a Sequence Of Fragments	666
<a href="#">gdcm::SequenceOfItems</a>	
Class to represent a Sequence Of Items (value representation : SQ)	671
<a href="#">gdcm::SerieHelper</a>	
<a href="#">SerieHelper</a> 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	677
<a href="#">gdcm::Series</a>	
<a href="#">Series</a>	679
<a href="#">gdcm::network::ServiceClassApplicationInformation</a>	680
<a href="#">gdcm::ServiceClassUser</a>	
<a href="#">ServiceClassUser</a>	681
<a href="#">gdcm::SHA1</a>	
Class for <a href="#">SHA1</a>	686
<a href="#">gdcm::SimpleMemberCommand&lt; T &gt;</a>	
<a href="#">Command</a> subclass that calls a pointer to a member function	687
<a href="#">gdcm::SimpleSubjectWatcher</a>	
<a href="#">SimpleSubjectWatcher</a> This is a typical <a href="#">Subject</a> Watcher class. It will observe all events	690
<a href="#">gdcm::SmartPointer&lt; ObjectType &gt;</a>	
Class for Smart Pointer	692
<a href="#">gdcm::network::SOPClassExtendedNegociationSub</a>	
<a href="#">SOPClassExtendedNegociationSub</a> PS 3.7 <a href="#">Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)</a>	695
<a href="#">gdcm::SOPClassUIDToIOD</a>	
Class convert a class SOP Class UID into <a href="#">IOD</a>	696
<a href="#">gdcm::Sorter</a>	
<a href="#">Sorter</a> General class to do sorting using a custom function You simply need to provide a function of type: <a href="#">Sorter::SortFunction</a>	697
<a href="#">gdcm::Spacing</a>	
Class for <a href="#">Spacing</a>	701
<a href="#">gdcm::Spectroscopy</a>	
<a href="#">Spectroscopy</a> class	703
<a href="#">gdcm::SplitMosaicFilter</a>	
<a href="#">SplitMosaicFilter</a> class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA <a href="#">Image</a> Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture	703
<a href="#">gdcm::StartEvent</a>	704

<a href="#">gdcmm::static_assert_test&lt; x &gt;</a>	706
<a href="#">gdcmm::STATIC_ASSERTION_FAILURE&lt; x &gt;</a>	706
<a href="#">gdcmm::STATIC_ASSERTION_FAILURE&lt; true &gt;</a>	706
<a href="#">gdcmm::StreamImageReader</a>	
<a href="#">StreamImageReader</a>	706
<a href="#">gdcmm::StreamImageWriter</a>	
<a href="#">StreamImageReader</a>	709
<a href="#">gdcmm::String&lt; TDelimiter, TMaxLength, TPadChar &gt;</a>	
<a href="#">String</a>	714
<a href="#">gdcmm::StringFilter</a>	
<a href="#">StringFilter</a> <a href="#">StringFilter</a> is the class that make gdcmm2.x looks more like gdcmm1 and transform the binary blob contained in a <a href="#">DataElement</a> into a string, typically this is a nice feature to have for wrapped language	718
<a href="#">gdcmm::Study</a>	
<a href="#">Study</a>	720
<a href="#">gdcmm::Subject</a>	
<a href="#">Subject</a>	721
<a href="#">gdcmm::Surface</a>	
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes	723
<a href="#">gdcmm::SurfaceHelper</a>	
<a href="#">SurfaceHelper</a> Helper class for <a href="#">Surface</a> object	730
<a href="#">gdcmm::SurfaceReader</a>	
This class defines a SURFACE IE reader. It reads surface mesh module attributes	732
<a href="#">gdcmm::SurfaceWriter</a>	
This class defines a SURFACE IE writer. It writes surface mesh module attributes	734
<a href="#">gdcmm::SwapCode</a>	
<a href="#">SwapCode</a> representation	736
<a href="#">gdcmm::SwapperDoOp</a>	738
<a href="#">gdcmm::SwapperNoOp</a>	739
<a href="#">gdcmm::System</a>	
Class to do system operation	739
<a href="#">gdcmm::Table</a>	
<a href="#">Table</a>	744
<a href="#">gdcmm::TableEntry</a>	
<a href="#">TableEntry</a>	745
<a href="#">gdcmm::TableReader</a>	
Class for representing a <a href="#">TableReader</a>	745
<a href="#">gdcmm::network::TableRow</a>	747
<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> ). Basically an uint32_t which can also be expressed as two uint16_t (group and element)	749
<a href="#">gdcmm::TagPath</a>	
Class to handle a path of tag	756
<a href="#">gdcmm::Testing</a>	
Class for testing	757
<a href="#">gdcmm::Trace</a>	
<a href="#">Trace</a>	761
<a href="#">gdcmm::TransferSyntax</a>	
Class to manipulate Transfer Syntax	764
<a href="#">gdcmm::network::TransferSyntaxSub</a>	
<a href="#">TransferSyntaxSub</a> <a href="#">Table</a> 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS	768
<a href="#">gdcmm::network::Transition</a>	769

gdcmm::Type	
Type	770
gdcmm::UI	772
gdcmm::UIDGenerator	
Class for generating unique UID	773
gdcmm::UIDs	
All known uids	774
gdcmm::network::ULAction	
ULAction A <a href="#">ULConnection</a> in a given ULState can perform certain ULActions. This base class provides the interface for running those ULActions on a given <a href="#">ULConnection</a>	793
gdcmm::network::ULActionAA1	796
gdcmm::network::ULActionAA2	797
gdcmm::network::ULActionAA3	798
gdcmm::network::ULActionAA4	799
gdcmm::network::ULActionAA5	800
gdcmm::network::ULActionAA6	801
gdcmm::network::ULActionAA7	803
gdcmm::network::ULActionAA8	804
gdcmm::network::ULActionAE1	805
gdcmm::network::ULActionAE2	806
gdcmm::network::ULActionAE3	807
gdcmm::network::ULActionAE4	808
gdcmm::network::ULActionAE5	810
gdcmm::network::ULActionAE6	811
gdcmm::network::ULActionAE7	812
gdcmm::network::ULActionAE8	813
gdcmm::network::ULActionAR1	814
gdcmm::network::ULActionAR10	815
gdcmm::network::ULActionAR2	817
gdcmm::network::ULActionAR3	818
gdcmm::network::ULActionAR4	819
gdcmm::network::ULActionAR5	820
gdcmm::network::ULActionAR6	821
gdcmm::network::ULActionAR7	822
gdcmm::network::ULActionAR8	824
gdcmm::network::ULActionAR9	825
gdcmm::network::ULActionDT1	826
gdcmm::network::ULActionDT2	827
gdcmm::network::ULBasicCallback	
ULBasicCallback This is the most basic of callbacks for how the <a href="#">ULConnectionManager</a> 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 <a href="#">ULConnectionManager</a>	828
gdcmm::network::ULConnection	
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	830
gdcmm::network::ULConnectionCallback	832
gdcmm::network::ULConnectionInfo	
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	834

<a href="#">gdcm::network::ULConnectionManager</a>	
<a href="#">ULConnectionManager</a>	The <a href="#">ULConnectionManager</a> performs actions on the <a href="#">ULConnection</a> 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) . . . . .
	835
<a href="#">gdcm::network::ULEvent</a>	
<a href="#">ULEvent</a>	base class for network events . . . . .
	838
<a href="#">gdcm::network::ULTransitionTable</a>	
<a href="#">ULTransitionTable</a>	The transition table of all the <a href="#">ULEvents</a> , new <a href="#">ULActions</a> , and <a href="#">ULStates</a> . . . . .
	839
<a href="#">gdcm::network::ULWritingCallback</a>	. . . . .
	840
<a href="#">gdcm::UNExplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as UNExplicit Data <a href="#">Element</a>	. . . . .
	842
<a href="#">gdcm::UNExplicitImplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as ExplicitImplicit Data <a href="#">Element</a>	This class gather two known bugs: . . . . .
	843
<a href="#">gdcm::Unpacker12Bits</a>	
Pack/Unpack 12 bits pixel into 16bits . . . . .	845
<a href="#">gdcm::Usage</a>	
Usage . . . . .	846
<a href="#">gdcm::UserEvent</a>	. . . . .
	848
<a href="#">gdcm::network::UserInformation</a>	
<a href="#">UserInformation</a> Table 9-16 USER INFORMATION ITEM FIELDS . . . . .	849
<a href="#">gdcm::UUIDGenerator</a>	
Class for generating unique UUID generate DCE 1.1 uid . . . . .	850
<a href="#">gdcm::Validate</a>	
Validate class . . . . .	851
<a href="#">gdcm::Value</a>	
Class to represent the value of a Data <a href="#">Element</a> . . . . .	853
<a href="#">gdcm::ValueIO&lt; TDE, TSwap, TType &gt;</a>	
Class to dispatch template calls . . . . .	855
<a href="#">gdcm::Version</a>	
Major/minor and build version . . . . .	856
<a href="#">gdcm::VL</a>	
Value Length . . . . .	857
<a href="#">gdcm::VM</a>	
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 . . . . .	859
<a href="#">gdcm::VMToLength&lt; T &gt;</a>	. . . . .
	863
<a href="#">gdcm::VR</a>	
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 . . . . .	863
<a href="#">gdcm::VR16ExplicitDataElement</a>	
Class to read/write a <a href="#">DataElement</a> as Explicit Data <a href="#">Element</a> . . . . .	868
<a href="#">gdcm::VRToEncoding&lt; T &gt;</a>	. . . . .
	870
<a href="#">gdcm::VRToType&lt; T &gt;</a>	. . . . .
	870
<a href="#">gdcm::VRVLSize&lt; T &gt;</a>	. . . . .
	871
<a href="#">gdcm::VRVLSize&lt; 0 &gt;</a>	. . . . .
	871
<a href="#">gdcm::VRVLSize&lt; 1 &gt;</a>	. . . . .
	871
<a href="#">vtkGDCMImageReader</a>	. . . . .
	872
<a href="#">vtkGDCMImageReader2</a>	. . . . .
	878
<a href="#">vtkGDCMImageWriter</a>	. . . . .
	883
<a href="#">vtkGDCMMedicalImageProperties</a>	. . . . .
	888
<a href="#">vtkGDCMPolyDataReader</a>	. . . . .
	890
<a href="#">vtkGDCMPolyDataWriter</a>	. . . . .
	893

<a href="#">vtkGDCMTesting</a>	896
<a href="#">vtkGDCMThreadedImageReader</a>	898
<a href="#">vtkGDCMThreadedImageReader2</a>	901
<a href="#">vtkImageColorViewer</a>	904
<a href="#">vtkImageMapToColors16</a>	911
<a href="#">vtkImageMapToWindowLevelColors2</a>	914
<a href="#">vtkImagePlanarComponentsToComponents</a>	916
<a href="#">vtkImageRGBToYBR</a>	918
<a href="#">vtkImageYBRToRGB</a>	920
<a href="#">vtkLookupTable16</a>	921
<a href="#">vtkRTStructSetProperties</a>	923
<a href="#">gdcm::Waveform</a>	
<a href="#">Waveform</a> class	928
<a href="#">gdcm::Writer</a>	
<a href="#">Writer</a> ala DOM (Document <a href="#">Object</a> Model) This class is a non-validating writer, it will only performs well-formedness check only	928
<a href="#">gdcm::XMLDictReader</a>	
Class for representing a <a href="#">XMLDictReader</a>	934
<a href="#">gdcm::XMLPrinter</a>	935
<a href="#">gdcm::XMLPrivateDictReader</a>	
Class for representing a <a href="#">XMLPrivateDictReader</a>	938





## Chapter 25

# File Index

### 25.1 File List

Here is a list of all files with brief descriptions:

<a href="#">gdc2pnm.man</a>	941
<a href="#">gdc2vtk.man</a>	941
<a href="#">gdcmAAabortPDU.h</a>	941
<a href="#">gdcmAAAssociateACPDU.h</a>	942
<a href="#">gdcmAAAssociateRJPDU.h</a>	942
<a href="#">gdcmAAAssociateRQPDU.h</a>	943
<a href="#">gdcmAbstractSyntax.h</a>	944
<a href="#">gdcmanon.man</a>	945
<a href="#">gdcmAnonymizeEvent.h</a>	945
<a href="#">gdcmAnonymizer.h</a>	947
<a href="#">gdcmApplicationContext.h</a>	947
<a href="#">gdcmApplicationEntity.h</a>	948
<a href="#">gdcmAReleaseRPPDU.h</a>	949
<a href="#">gdcmAReleaseRQPDU.h</a>	950
<a href="#">gdcmARTIMTimer.h</a>	951
<a href="#">gdcmASN1.h</a>	952
<a href="#">gdcmAsynchronousOperationsWindowSub.h</a>	953
<a href="#">gdcmAttribute.h</a>	953
<a href="#">gdcmAudioCodec.h</a>	955
<a href="#">gdcmBase64.h</a>	955
<a href="#">gdcmBaseCompositeMessage.h</a>	956
<a href="#">gdcmBasePDU.h</a>	957
<a href="#">gdcmBaseRootQuery.h</a>	958
<a href="#">gdcmBasicOffsetTable.h</a>	960
<a href="#">gdcmBitmap.h</a>	961
<a href="#">gdcmBitmapToBitmapFilter.h</a>	962
<a href="#">gdcmBoxRegion.h</a>	963
<a href="#">gdcmByteBuffer.h</a>	963
<a href="#">gdcmByteSwap.h</a>	964
<a href="#">gdcmByteSwapFilter.h</a>	965
<a href="#">gdcmByteValue.h</a>	966
<a href="#">gdcmCAPICryptoFactory.h</a>	967
<a href="#">gdcmCAPICryptographicMessageSyntax.h</a>	967
<a href="#">gdcmCEchoMessages.h</a>	968

gdcmCFindMessages.h	969
gdcmCMoveMessages.h	970
gdcmCodec.h	971
gdcmCoder.h	972
gdcmCodeString.h	973
gdcmCommand.h	974
gdcmCommandDataSet.h	976
gdcmCompositeMessageFactory.h	976
gdcmCompositeNetworkFunctions.h	977
gdcmConstCharWrapper.h	978
gdcmconv.man	978
gdcmCP246ExplicitDataElement.h	979
gdcmCryptoFactory.h	979
gdcmCryptographicMessageSyntax.h	980
gdcmCSAElement.h	981
gdcmCSAHeader.h	982
gdcmCSAHeaderDict.h	983
gdcmCSAHeaderDictEntry.h	985
gdcmCStoreMessages.h	986
gdcmCurve.h	987
gdcmDataElement.h	988
gdcmDataEvent.h	990
gdcmDataSet.h	991
gdcmDataSetEvent.h	992
gdcmDataSetHelper.h	992
gdcmDecoder.h	993
gdcmDefinedTerms.h	994
gdcmDeflateStream.h	995
gdcmDefs.h	995
gdcmDeltaEncodingCodec.h	997
gdcmDICOMDIR.h	997
gdcmDICOMDIRGenerator.h	998
gdcmDict.h	999
gdcmDictConverter.h	1001
gdcmDictEntry.h	1001
gdcmDictPrinter.h	1003
gdcmDicts.h	1003
gdcmdiff.man	1004
gdcmDIMSE.h	1005
gdcmDirectionCosines.h	1005
gdcmDirectory.h	1006
gdcmDirectoryHelper.h	1007
gdcmDummyValueGenerator.h	1008
gdcmdump.man	1008
gdcmDumper.h	1009
gdcmElement.h	1009
gdcmEncapsulatedDocument.h	1011
gdcmEnumeratedValues.h	1011
gdcmEvent.h	1012
gdcmException.h	1014
gdcmExplicitDataElement.h	1015
gdcmExplicitImplicitDataElement.h	1015
gdcmFiducials.h	1016
gdcmFile.h	1017

<a href="#">gdcmFileAnonymizer.h</a>	1018
<a href="#">gdcmFileChangeTransferSyntax.h</a>	1018
<a href="#">gdcmFileDerivation.h</a>	1019
<a href="#">gdcmFileExplicitFilter.h</a>	1020
<a href="#">gdcmFileMetaInformation.h</a>	1021
<a href="#">gdcmFilename.h</a>	1022
<a href="#">gdcmFileNameEvent.h</a>	1022
<a href="#">gdcmFilenameGenerator.h</a>	1023
<a href="#">gdcmFileSet.h</a>	1024
<a href="#">gdcmFileStreamer.h</a>	1026
<a href="#">gdcmFindPatientRootQuery.h</a>	1026
<a href="#">gdcmFindStudyRootQuery.h</a>	1027
<a href="#">gdcmFragment.h</a>	1028
<a href="#">gdcmgendir.man</a>	1030
<a href="#">gdcmGlobal.h</a>	1030
<a href="#">gdcmGroupDict.h</a>	1031
<a href="#">gdcmIconImage.h</a>	1031
<a href="#">gdcmIconImageFilter.h</a>	1032
<a href="#">gdcmIconImageGenerator.h</a>	1033
<a href="#">gdcmImage.h</a>	1034
<a href="#">gdcmImageApplyLookupTable.h</a>	1035
<a href="#">gdcmImageChangePhotometricInterpretation.h</a>	1036
<a href="#">gdcmImageChangePlanarConfiguration.h</a>	1037
<a href="#">gdcmImageChangeTransferSyntax.h</a>	1038
<a href="#">gdcmImageCodec.h</a>	1039
<a href="#">gdcmImageConverter.h</a>	1040
<a href="#">gdcmImageFragmentSplitter.h</a>	1041
<a href="#">gdcmImageHelper.h</a>	1042
<a href="#">gdcmImageReader.h</a>	1043
<a href="#">gdcmImageRegionReader.h</a>	1044
<a href="#">gdcmImageToImageFilter.h</a>	1045
<a href="#">gdcmImageWriter.h</a>	1046
<a href="#">gdcmimg.man</a>	1047
<a href="#">gdcmImplementationClassUIDSub.h</a>	1047
<a href="#">gdcmImplementationUIDSub.h</a>	1048
<a href="#">gdcmImplementationVersionNameSub.h</a>	1049
<a href="#">gdcmImplicitDataElement.h</a>	1051
<a href="#">gdcminfo.man</a>	1051
<a href="#">gdcmIOD.h</a>	1051
<a href="#">gdcmIODEntry.h</a>	1053
<a href="#">gdcmIODs.h</a>	1056
<a href="#">gdcmIPPSorter.h</a>	1057
<a href="#">gdcmItem.h</a>	1058
<a href="#">gdcmJPEG12Codec.h</a>	1060
<a href="#">gdcmJPEG16Codec.h</a>	1060
<a href="#">gdcmJPEG2000Codec.h</a>	1061
<a href="#">gdcmJPEG8Codec.h</a>	1062
<a href="#">gdcmJPEGCodec.h</a>	1063
<a href="#">gdcmJPEGLSCodec.h</a>	1064
<a href="#">gdcmJSON.h</a>	1065
<a href="#">gdcmKAKADUCodec.h</a>	1066
<a href="#">gdcmLegacyMacro.h</a>	1067
<a href="#">gdcmLO.h</a>	1068
<a href="#">gdcmLookupTable.h</a>	1069

<a href="#">gdcmMacro.h</a>	1070
<a href="#">gdcmMacroEntry.h</a>	1072
<a href="#">gdcmMacros.h</a>	1074
<a href="#">gdcmMaximumLengthSub.h</a>	1076
<a href="#">gdcmMD5.h</a>	1077
<a href="#">gdcmMediaStorage.h</a>	1078
<a href="#">gdcmMeshPrimitive.h</a>	1079
<a href="#">gdcmModule.h</a>	1081
<a href="#">gdcmModuleEntry.h</a>	1082
<a href="#">gdcmModules.h</a>	1084
<a href="#">gdcmMovePatientRootQuery.h</a>	1085
<a href="#">gdcmMoveStudyRootQuery.h</a>	1086
<a href="#">gdcmNestedModuleEntries.h</a>	1087
<a href="#">gdcmNetworkEvents.h</a>	1089
<a href="#">gdcmNetworkStateID.h</a>	1090
<a href="#">gdcmObject.h</a>	1091
<a href="#">gdcmOpenSSLCryptoFactory.h</a>	1092
<a href="#">gdcmOpenSSLCryptographicMessageSyntax.h</a>	1092
<a href="#">gdcmOpenSSLP7CryptoFactory.h</a>	1093
<a href="#">gdcmOpenSSLP7CryptographicMessageSyntax.h</a>	1094
<a href="#">gdcmOrientation.h</a>	1096
<a href="#">gdcmOverlay.h</a>	1096
<a href="#">gdcmpap3.man</a>	1097
<a href="#">gdcmParseException.h</a>	1097
<a href="#">gdcmParser.h</a>	1099
<a href="#">gdcmPatient.h</a>	1099
<a href="#">gdcmPDataTFPDU.h</a>	1100
<a href="#">gdcmPDBElement.h</a>	1101
<a href="#">gdcmPDBHeader.h</a>	1103
<a href="#">gdcmpdf.man</a>	1103
<a href="#">gdcmPDFCodec.h</a>	1103
<a href="#">gdcmPDUFactory.h</a>	1104
<a href="#">gdcmPersonName.h</a>	1105
<a href="#">gdcmPGXCodec.h</a>	1106
<a href="#">gdcmPhotometricInterpretation.h</a>	1106
<a href="#">gdcmPixelFormat.h</a>	1107
<a href="#">gdcmPixmap.h</a>	1108
<a href="#">gdcmPixmapReader.h</a>	1109
<a href="#">gdcmPixmapToPixmapFilter.h</a>	1111
<a href="#">gdcmPixmapWriter.h</a>	1111
<a href="#">gdcmPNMCodec.h</a>	1112
<a href="#">gdcmPreamble.h</a>	1113
<a href="#">gdcmPresentationContext.h</a>	1114
<a href="#">gdcmPresentationContextAC.h</a>	1115
<a href="#">gdcmPresentationContextGenerator.h</a>	1117
<a href="#">gdcmPresentationContextRQ.h</a>	1117
<a href="#">gdcmPresentationDataValue.h</a>	1118
<a href="#">gdcmPrinter.h</a>	1119
<a href="#">gdcmPrivateTag.h</a>	1120
<a href="#">gdcmProgressEvent.h</a>	1122
<a href="#">gdcmPVRGCodec.h</a>	1122
<a href="#">gdcmPythonFilter.h</a>	1123
<a href="#">gdcmQueryBase.h</a>	1124
<a href="#">gdcmQueryFactory.h</a>	1126

gdcmQueryImage.h	1127
gdcmQueryPatient.h	1127
gdcmQuerySeries.h	1128
gdcmQueryStudy.h	1129
gdcmraw.man	1130
gdcmRAWCodec.h	1130
gdcmReader.h	1131
gdcmRegion.h	1133
gdcmRescaler.h	1134
gdcmRLECodec.h	1134
gdcmRoleSelectionSub.h	1135
gdcmScanner.h	1136
gdcmscanner.man	1137
gdcmscu.man	1137
gdcmSegment.h	1137
gdcmSegmentedPaletteColorLookupTable.h	1139
gdcmSegmentHelper.h	1139
gdcmSegmentReader.h	1141
gdcmSegmentWriter.h	1142
gdcmSequenceOfFragments.h	1143
gdcmSequenceOfItems.h	1144
gdcmSerieHelper.h	1144
gdcmSeries.h	1146
gdcmServiceClassApplicationInformation.h	1147
gdcmServiceClassUser.h	1148
gdcmSHA1.h	1148
gdcmSimpleSubjectWatcher.h	1149
gdcmSmartPointer.h	1150
gdcmSOPClassExtendedNegociationSub.h	1151
gdcmSOPClassUIDToIOD.h	1152
gdcmSorter.h	1153
gdcmSpacing.h	1155
gdcmSpectroscopy.h	1155
gdcmSplitMosaicFilter.h	1156
gdcmStaticAssert.h	1157
gdcmStreamImageReader.h	1158
gdcmStreamImageWriter.h	1158
gdcmString.h	1159
gdcmStringFilter.h	1160
gdcmStudy.h	1161
gdcmSubject.h	1162
gdcmSurface.h	1163
gdcmSurfaceHelper.h	1164
gdcmSurfaceReader.h	1165
gdcmSurfaceWriter.h	1166
gdcmSwapCode.h	1167
gdcmSwapper.h	1168
gdcmSystem.h	1169
gdcmTable.h	1170
gdcmTableEntry.h	1171
gdcmTableReader.h	1172
gdcmTag.h	1174
gdcmTagPath.h	1175
gdcmTagToVR.h	1175

gdcmtar.man	1175
gdcmTerminal.h	1176
gdcmTestDriver.h	1177
gdcmTesting.h	1177
gdcmTrace.h	1178
gdcmTransferSyntax.h	1181
gdcmTransferSyntaxSub.h	1182
gdcmType.h	1183
gdcmTypes.h	1185
gdcmUIDGenerator.h	1185
gdcmUIDs.h	1186
gdcmULAction.h	1187
gdcmULActionAA.h	1188
gdcmULActionAE.h	1189
gdcmULActionAR.h	1190
gdcmULActionDT.h	1190
gdcmULBasicCallback.h	1191
gdcmULConnection.h	1192
gdcmULConnectionCallback.h	1193
gdcmULConnectionInfo.h	1194
gdcmULConnectionManager.h	1195
gdcmULEvent.h	1196
gdcmULTransitionTable.h	1197
gdcmULWritingCallback.h	1198
gdcmUNExplicitDataElement.h	1198
gdcmUNExplicitImplicitDataElement.h	1199
gdcmUnpacker12Bits.h	1200
gdcmUsage.h	1200
gdcmUserInformation.h	1203
gdcmUUIDGenerator.h	1204
gdcmValidate.h	1204
gdcmValue.h	1205
gdcmValueIO.h	1206
gdcmVersion.h	1207
gdcmviewer.man	1208
gdcmVL.h	1208
gdcmVM.h	1209
gdcmVR.h	1210
gdcmVR16ExplicitDataElement.h	1213
gdcmWaveform.h	1213
gdcmWin32.h	1214
gdcmWriter.h	1214
gdcmxml.man	1215
gdcmXMLDictReader.h	1215
gdcmXMLPrinter.h	1216
gdcmXMLPrivateDictReader.h	1216
vtkGDCMImageReader.h	1217
vtkGDCMImageReader2.h	1218
vtkGDCMImageWriter.h	1219
vtkGDCMMedicalImageProperties.h	1220
vtkGDCMPolyDataReader.h	1221
vtkGDCMPolyDataWriter.h	1221
vtkGDCMTesting.h	1222
vtkGDCMThreadedImageReader.h	1222

<a href="#">vtkGDCMThreadedImageReader2.h</a>	1223
<a href="#">vtkImageColorViewer.h</a>	1224
<a href="#">vtkImageMapToColors16.h</a>	1224
<a href="#">vtkImageMapToWindowLevelColors2.h</a>	1225
<a href="#">vtkImagePlanarComponentsToComponents.h</a>	1225
<a href="#">vtkImageRGBToYBR.h</a>	1226
<a href="#">vtkImageYBRToRGB.h</a>	1226
<a href="#">vtkLookupTable16.h</a>	1227
<a href="#">vtkRTStructSetProperties.h</a>	1227





## Chapter 26

# Namespace Documentation

### 26.1 gdcM Namespace Reference

#### Namespaces

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

*Class for Terminal Allow one to print in color in a shell.*

#### Classes

- class [AbortEvent](#)
- class [AnonymizeEvent](#)  
*[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.*
- class [Anonymizer](#)  
*[Anonymizer](#) This class is a multi purpose anonymizer. It can work in 2 mode:*
- 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 instanciation 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 [BaseRootQuery](#)
  - [BaseRootQuery](#) contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.
- class [BasicOffsetTable](#)
  - Class to represent a [BasicOffsetTable](#).
- class [Bitmap](#)
  - [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)
- class [BitmapToBitmapFilter](#)
  - [BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.
- class [BoxRegion](#)
  - 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)
- class [ByteBuffer](#)
  - [ByteBuffer](#).
- class [ByteSwap](#)
  - [ByteSwap](#).
- class [ByteSwapFilter](#)
  - [ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??
- class [ByteValue](#)
  - Class to represent binary value (array of bytes)
- class [CAPICryptoFactory](#)
- class [CAPICryptographicMessageSyntax](#)
- class [Codec](#)
  - [Codec](#) class.
- class [Coder](#)
  - [Coder](#).
- class [CodeString](#)
  - [CodeString](#) This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that operator== is correct.
- class [Command](#)
  - [Command](#) superclass for callback/observer methods.
- class [CommandDataSet](#)
  - Class to represent a [Command DataSet](#).
- class [CompositeNetworkFunctions](#)
  - [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:
- 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](#) 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.*
- class [CSAHeaderDictException](#)
- class [Curve](#)
  - [Curve](#) class to handle element 50xx,3000 [Curve](#) Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.*
- 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) A Data Set represents an instance of a real world Information [Object](#).*
- class [DataSetEvent](#)
  - [DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.*
- 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 This is a STD-GEN-CD [DICOMDIR](#) generator. ref: PS 3.11-2008 Annex D (Normative) - General Purpose CD-R and DVD Interchange Profiles.*
- 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* 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.

- 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* 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.

- 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 [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](#) 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.*
- class [FileAnonymizer](#)  
*FileAnonymizer.*
- class [FileChangeTransferSyntax](#)  
*FileChangeTransferSyntax.*
- class [FileDerivation](#)  
*FileDerivation class See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence.*
- class [FileExplicitFilter](#)  
*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.*
- class [FileMetaInformation](#)  
*Class to represent a [File](#) Meta Information.*
- class [Filename](#)  
*Class to manipulate file name's.*
- class [FileNameEvent](#)  
*FileNameEvent Special type of event triggered during processing of [FileSet](#).*
- class [FilenameGenerator](#)  
*FilenameGenerator.*
- class [FileSet](#)  
*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.*
- class [FileStreamer](#)  
*FileStreamer This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.*
- class [FileWithName](#)  
*FileWithName.*
- class [FindPatientRootQuery](#)  
*PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.*
- class [FindStudyRootQuery](#)  
*FindStudyRootQuery contains: the class which will produce a dataset for C-FIND with study root.*
- 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](#) This filter will extract icons from a [gdcm::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.*

- class [IconImageGenerator](#)

*[IconImageGenerator](#) This filter will generate a valid Icon from the Pixel Data element (an instance of [gdcm::Pixmap](#)). To generate a valid Icon, one is only allowed the following Photometric Interpretation:*

- struct [ignore\\_char](#)

- class [Image](#)

*[Image](#) This is the container for an [Image](#) in the general sense. From this container you should be able to request information like:*

- class [ImageApplyLookupTable](#)

*[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image.*

- class [ImageChangePhotometricInterpretation](#)

*[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.*

- class [ImageChangePlanarConfiguration](#)

*[ImageChangePlanarConfiguration](#) class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: [PlanarConfiguration](#) = 0.*

- class [ImageChangeTransferSyntax](#)

*[ImageChangeTransferSyntax](#) class Class to change the transfer syntax of an input DICOM.*

- class [ImageCodec](#)

*[ImageCodec](#).*

- class [ImageConverter](#)

*[Image](#) Converter.*

- class [ImageFragmentSplitter](#)

*[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.*

- class [ImageHelper](#)

*[ImageHelper](#) (internal class, not intended for user level)*

- class [ImageReader](#)

*[ImageReader](#).*

- class [ImageRegionReader](#)

*[ImageRegionReader](#).*

- class [ImageToImageFilter](#)

*[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.*

- 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](#) 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.*

- class [Item](#)

*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.*
- 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 to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.*
- class [JPEGLSCCodec](#)

*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. It is designed from surface mesh primitives macro.*
- 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 contains: the class which will produce a dataset for c-move with patient root.*

- class [MoveStudyRootQuery](#)  
*MoveStudyRootQuery* contains: the class which will produce a dataset for C-MOVE with study root.
- class [NestedModuleEntries](#)  
*Class for representing a NestedModuleEntries.*
- class [NoEvent](#)
- class [Object](#)  
*Object.*
- class [OpenSSLCryptoFactory](#)
- class [OpenSSLCryptographicMessageSyntax](#)
- class [OpenSSLP7CryptoFactory](#)
- class [OpenSSLP7CryptographicMessageSyntax](#)  
*Class for CryptographicMessageSyntax encryption. This is just a simple wrapper around openssl PKCS7\_encrypt functionalities.*
- 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 See PGX as used in JPEG 2000 implementation and reference images.*
- class [PhotometricInterpretation](#)  
*Class to represent an PhotometricInterpretation.*
- class [PixelFormat](#)  
*PixelFormat.*
- class [Pixmap](#)  
*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)*
- class [PixmapReader](#)  
*PixmapReader.*
- class [PixmapToPixmapFilter](#)  
*PixmapToPixmapFilter class Super class for all filter taking an image and producing an output image.*
- class [PixmapWriter](#)  
*PixmapWriter This class will takes two inputs:*
- class [PNMCodec](#)



Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.

- class [Preamble](#)  
*DICOM Preamble (Part 10)*
- class [PresentationContext](#)  
*PresentationContext.*
- class [PresentationContextGenerator](#)  
*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.
- 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* Special type of event triggered during.
- 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* contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.
- class [QueryFactory](#)  
*QueryFactory.h.*
- class [QueryImage](#)  
*QueryImage* contains: class to construct an image-based query for C-FIND and C-MOVE.
- class [QueryPatient](#)  
*QueryPatient* contains: class to construct a patient-based query for c-find and c-move.
- class [QuerySeries](#)  
*QuerySeries* contains: class to construct a series-based query for c-find and c-move.
- class [QueryStudy](#)  
*QueryStudy.h* contains: class to construct a study-based query for C-FIND and C-MOVE.
- class [RAWCodec](#)  
*RAWCodec* class.
- class [Reader](#)  
*Reader* ala DOM (Document Object Model)
- class [Region](#)  
*Class for manipulation region.*
- class [Rescaler](#)  
*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.

- class [RLECodec](#)  
*Class to do RLE.*
- class [Scanner](#)  
*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](#).*
- class [Segment](#)  
*This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.*
- class [SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable class.*
- class [SegmentReader](#)  
*This class defines a segment reader. It reads attributes of group 0x0062.*
- class [SegmentWriter](#)  
*This class defines a segment writer. It writes attributes of group 0x0062.*
- class [SequenceOfFragments](#)  
*Class to represent a Sequence Of Fragments.*
- class [SequenceOfItems](#)  
*Class to represent a Sequence Of Items (value representation : SQ)*
- 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 This is a typical [Subject](#) Watcher class. It will observe all events.*
- class [SmartPointer](#)  
*Class for Smart Pointer.*
- class [SOPClassUIDToIOD](#)  
*Class convert a class SOP Class UID into [IOD](#).*
- class [Sorter](#)  
*Sorter General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::SortFunction](#).*
- class [Spacing](#)  
*Class for Spacing.*
- class [Spectroscopy](#)  
*Spectroscopy class.*
- class [SplitMosaicFilter](#)  
*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.*
- class [StartEvent](#)
- struct [static\\_assert\\_test](#)
- struct [STATIC\\_ASSERTION\\_FAILURE](#)
- struct [STATIC\\_ASSERTION\\_FAILURE< true >](#)

- class [StreamImageReader](#)  
*StreamImageReader.*
- class [StreamImageWriter](#)  
*StreamImageReader.*
- class [String](#)  
*String.*
- class [StringFilter](#)  
*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.
- class [Study](#)  
*Study.*
- class [Subject](#)  
*Subject.*
- class [Surface](#)  
*This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.*
- class [SurfaceHelper](#)  
*SurfaceHelper* Helper class for *Surface* object.
- class [SurfaceReader](#)  
*This class defines a SURFACE IE reader. It reads surface mesh module attributes.*
- class [SurfaceWriter](#)  
*This class defines a SURFACE IE writer. It writes surface mesh module attributes.*
- 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](#)). Basically an uint32\_t which can also be expressed as two uint16\_t (group and 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](#) This class gather two known bugs:*
- class [Unpacker12Bits](#)
  - Pack/Unpack 12 bits pixel into 16bits.*
- class [Usage](#)
  - Usage.*
- class [UserEvent](#)
- class [UUIDGenerator](#)
  - Class for generating unique UUID generate DCE 1.1 uid.*
- 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 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.*
- 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 [Writer](#)
  - [Writer](#) ala DOM (Document [Object](#) Model) This class is a non-validating writer, it will only performs well- formedness check only.*
- 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 > [UICComp](#)
- typedef [String](#)<"\", 64 > [UTComp](#)

## Enumerations

- enum [CompOperators](#) {  
    [GDCM\\_EQUAL](#) = 0,  
    [GDCM\\_DIFFERENT](#),  
    [GDCM\\_GREATER](#),  
    [GDCM\\_GREATEROREQUAL](#),  
    [GDCM\\_LESS](#),  
    [GDCM\\_LESSCOREQUAL](#) }
- enum [ECharSet](#) {  
    [eLatin1](#) = 0,  
    [eLatin2](#),  
    [eLatin3](#),  
    [eLatin4](#),  
    [eCyrillic](#),  
    [eArabic](#),  
    [eGreek](#),  
    [eHebrew](#),  
    [eLatin5](#),  
    [eJapanese](#),  
    [eThai](#),  
    [eJapaneseKanjiMultibyte](#),  
    [eJapaneseSupplementaryKanjiMultibyte](#),  
    [eKoreanHangulHanjaMultibyte](#),  
    [eUTF8](#),  
    [eGB18030](#) }
- enum [EQueryLevel](#) {  
    [ePatient](#) = 0,  
    [eStudy](#) = 1,  
    [eSeries](#) = 2,  
    [eImage](#) = 3 }

- enum [EQueryType](#) {  
    [eFind](#) = 0,  
    [eMove](#) }
- enum [ERootType](#) {  
    [ePatientRootType](#),  
    [eStudyRootType](#) }
- enum [LodModeType](#) {  
    [LD\\_ALL](#) = 0x00000000,  
    [LD\\_NOSEQ](#) = 0x00000001,  
    [LD\\_NOSHADOW](#) = 0x00000002,  
    [LD\\_NOSHADOWSEQ](#) = 0x00000004 }

## Functions

- [ignore\\_char](#) const [backslash](#) ('\\')
  - [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 [PDSElement](#) &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 [Sorter](#) &s)
  - std::ostream & [operator<<](#) (std::ostream &os, const [CSAHeaderDictEntry](#) &val)
  - 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 [Module](#) &\_val)
  - std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &val)
  - std::ostream & [operator<<](#) (std::ostream &os, const [Directory](#) &d)

- `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 Dict &val)`
- `std::ostream & operator<< (std::ostream &_os, const MediaStorage &ms)`
- `std::ostream & operator<< (std::ostream &_os, const VR &val)`
- `std::ostream & operator<< (std::ostream &os, const PixelFormat &pf)`
- `std::ostream & operator<< (std::ostream &os, const Fragment &val)`
- `std::ostream & operator<< (std::ostream &_os, const UI &_val)`
- `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 DataSet &val)`
- `std::ostream & operator<< (std::ostream &os, const Item &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 Float >  
std::string to_string (Float data)`
- `TYPETOENCODING (SQ, VRBINARY, unsigned char) TYPETOENCODING(UN`

## Variables

- static `Global GlobalInstance`
- `VRBINARY`

### 26.1.1 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.

## 26.1.2 Typedef Documentation

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

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

26.1.2.3 `typedef bool(* gdcm::BOOL_FUNCTION_PFILE_PFILE_POINTER)(File *, File *)`

26.1.2.4 `typedef String<'\\',16> gdcm::CSCComp`

26.1.2.5 `typedef String<'\\',64> gdcm::DACComp`

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

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

26.1.2.8 `typedef Bitmap gdcm::IconImage`

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

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

26.1.2.11 `typedef ModuleEntry gdcm::MacroEntry`

26.1.2.12 `typedef NestedModuleEntries gdcm::NestedMacroEntries`

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

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

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

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

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

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

## 26.1.3 Enumeration Type Documentation

26.1.3.1 `enum gdcm::CompOperators`

Enumerator

***GDCM\_EQUAL***



***GDCM\_DIFFERENT***  
***GDCM\_GREATER***  
***GDCM\_GREATEROREQUAL***  
***GDCM\_LESS***  
***GDCM\_LESOREQUAL***

#### 26.1.3.2 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***

#### 26.1.3.3 enum gdcm::EQueryLevel

Enumerator

***ePatient***  
***eStudy***  
***eSeries***  
***eImage***

#### 26.1.3.4 enum gdcm::EQueryType

Enumerator

***eFind***  
***eMove***

## 26.1.3.5 enum gdcm::ERootType

Enumerator

***ePatientRootType***

***eStudyRootType***

## 26.1.3.6 enum gdcm::LodModeType

Enumerator

***LD\_ALL***

***LD\_NOSEQ***

***LD\_NOSHADOW***

***LD\_NOSHADOWSEQ***

## 26.1.4 Function Documentation

## 26.1.4.1 ignore\_char const gdcm::backslash ( '\ ' )

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

## 26.1.4.2 VR::VRType gdcm::GetVRFromTag ( Tag const &amp; tag )

## 26.1.4.3 bool gdcm::operator!= ( const CodeString &amp; ref, const CodeString &amp; cs ) [inline]

## 26.1.4.4 bool gdcm::operator!= ( const DataElement &amp; lhs, const DataElement &amp; rhs ) [inline]

## 26.1.4.5 std::ostream&amp; gdcm::operator&lt;&lt; ( std::ostream &amp; os, const Version &amp; v ) [inline]

References gdcm::Version::Print().

## 26.1.4.6 std::ostream&amp; gdcm::operator&lt;&lt; ( std::ostream &amp; \_os, const NestedModuleEntries &amp; \_val ) [inline]

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

## 26.1.4.7 std::ostream&amp; gdcm::operator&lt;&lt; ( std::ostream &amp; os, const SwapCode &amp; sc ) [inline]

References gdcm::SwapCode::GetSwapCodeString().

## 26.1.4.8 std::ostream&amp; gdcm::operator&lt;&lt; ( std::ostream &amp; os, const FileSet &amp; f ) [inline]

## 26.1.4.9 std::ostream&amp; gdcm::operator&lt;&lt; ( std::ostream &amp; os, const Region &amp; r ) [inline]

References gdcm::Region::Print().

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

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

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

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

References `gdcm::PDBElement::NameField`, and `gdcm::PDBElement::ValueField`.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

References `gdcmm::Sorter::Print()`.

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

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

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

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

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

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

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

References `gdcmm::PDBHeader::Print()`.

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

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

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

References `gdcmm::PhotometricInterpretation::GetPIString()`.

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

References `gdcmm::Directory::Print()`.

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

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

References `gdcmm::Object::Print()`.

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

References `gdcmm::DataElement::GetByteValue()`, `gdcmm::DataElement::ValueField`, and `gdcmm::DataElement::Value↵LengthField`.

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

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

26.1.4.41 `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`.

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

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

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

References `gdcm::TransferSyntax::GetTSSString()`.

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

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

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

References `gdcm::VM::GetVMString()`.

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

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

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

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

References `gdcm::MediaStorage::GetMSString()`.

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

References `gdcm::VR::GetVRString()`.

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

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

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

References `gdcm::DataElement::TagField`, `gdcm::DataElement::ValueField`, and `gdcm::DataElement::ValueLengthField`.

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

References `gdcmm::UI::Internal`.

26.1.4.53 `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`.

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

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

References `gdcmm::DataSet::Print()`.

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

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

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

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

References `gdcmm::UIDs::GetName()`, and `gdcmm::UIDs::GetString()`.

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

Examples:

[DumpPhilipsECHO.cxx](#).

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

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

References `gdcmm::ignore_char::m_char`.

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

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

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

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

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

## 26.1.5 Variable Documentation

26.1.5.1 Global gdcmm::GlobalInstance [static]

26.1.5.2 gdcmm::VRBINARY

## 26.2 gdcmm::network Namespace Reference

### Classes

- class [AAbortPDU](#)  
*[AAbortPDU Table](#) 9-26 A-ABORT PDU FIELDS.*
- class [AAssociateACPDU](#)  
*[AAssociateACPDU Table](#) 9-17 ASSOCIATE-AC PDU fields.*
- class [AAssociateRJPDU](#)  
*[AAssociateRJPDU Table](#) 9-21 ASSOCIATE-RJ PDU FIELDS.*
- class [AAssociateRQPDU](#)  
*[AAssociateRQPDU Table](#) 9-11 ASSOCIATE-RQ PDU fields.*
- class [AbstractSyntax](#)  
*[AbstractSyntax Table](#) 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.*
- class [ApplicationContext](#)  
*[ApplicationContext Table](#) 9-12 APPLICATION CONTEXT ITEM FIELDS.*
- class [AReleaseRPPDU](#)  
*[AReleaseRPPDU Table](#) 9-25 A-RELEASE-RP PDU fields.*
- class [AReleaseRQPDU](#)  
*[AReleaseRQPDU Table](#) 9-24 A-RELEASE-RQ PDU FIELDS.*
- class [ARTIMTimer](#)  
*[ARTIMTimer](#) This file contains the code for the ARTIM timer.*
- class [AsynchronousOperationsWindowSub](#)  
*[AsynchronousOperationsWindowSub](#) PS 3.7 [Table](#) D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIELDS (A-ASSOCIATE-RQ)↔*
- class [BaseCompositeMessage](#)  
*[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.*
- class [BasePDU](#)  
*[BasePDU](#) base class for PDUs.*
- class [CEchoRQ](#)  
*[CEchoRQ](#) this file defines the messages for the echo action.*
- class [CEchoRSP](#)  
*[CEchoRSP](#) this file defines the messages for the echo action.*
- class [CFind](#)
- class [CFindCancelRQ](#)  
*[CFindCancelRQ](#) this file defines the messages for the cfind action.*
- class [CFindRQ](#)  
*[CFindRQ](#) this file defines the messages for the cfind action.*

- class [CFindRSP](#)  
*CFindRSP* this file defines the messages for the cfind action.
- class [CMoveCancelRq](#)
- class [CMoveRQ](#)  
*CMoveRQ* this file defines the messages for the cmove action.
- class [CMoveRSP](#)  
*CMoveRSP* this file defines the messages for the cmove action.
- class [CompositeMessageFactory](#)  
*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).
- class [CStoreRQ](#)  
*CStoreRQ* this file defines the messages for the cecho action.
- class [CStoreRSP](#)  
*CStoreRSP* this file defines the messages for the cecho action.
- class [DIMSE](#)  
*DIMSE* PS 3.7 - 2009 Annex E [Command](#) Dictionary (Normative) E.1 REGISTRY OF DICOM COMMAND ELEMENTS  
[Table E.1-1 COMMAND FIELDS \(PART 1\)](#)
- class [ImplementationClassUIDSub](#)  
*ImplementationClassUIDSub* PS 3.7 [Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [ImplementationUIDSub](#)  
*ImplementationUIDSub* [Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS \(A-ASSOCIATE-AC\)](#)
- class [ImplementationVersionNameSub](#)  
*ImplementationVersionNameSub* [Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [MaximumLengthSub](#)  
*MaximumLengthSub* Annex D [Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [PDataTFPDU](#)  
*PDataTFPDU* [Table 9-22 P-DATA-TF PDU FIELDS.](#)
- class [PDUFactory](#)  
*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.
- class [PresentationContextAC](#)  
*PresentationContextAC* [Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.](#)
- class [PresentationContextRQ](#)  
*PresentationContextRQ* [Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.](#)
- class [PresentationDataValue](#)  
*PresentationDataValue* [Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.](#)
- class [RoleSelectionSub](#)  
*RoleSelectionSub* PS 3.7 [Table D.3-9 SCP/SCU ROLE SELECTION SUB-ITEM FIELDS \(A-ASSOCIATE-RQ\)](#)
- class [ServiceClassApplicationInformation](#)
- class [SOPClassExtendedNegociationSub](#)  
*SOPClassExtendedNegociationSub* PS 3.7 [Table D.3-11 SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS \(A-ASSOCIATE-RQ and A-ASSOCIATE-AC\)](#)
- class [TableRow](#)
- class [TransferSyntaxSub](#)  
*TransferSyntaxSub* [Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.](#)



- struct [Transition](#)
- class [ULAction](#)

*[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](#).*

- 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](#) 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](#).*

- class [ULConnection](#)
- class [ULConnectionCallback](#)
- class [ULConnectionInfo](#)

*[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.*

- class [ULConnectionManager](#)

*[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).*

- class [ULEvent](#)

*[ULEvent](#) base class for network events.*

- class [ULTransitionTable](#)

*ULTransitionTable* The transition table of all the ULEvents, new ULActions, and ULStates.

- class [ULWritingCallback](#)
- class [UserInformation](#)

*UserInformation Table* 9-16 USER INFORMATION ITEM FIELDS.

## Enumerations

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

## Functions

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

## Variables

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

- const int [cMaxStateID](#) = 13

## 26.2.1 Enumeration Type Documentation

### 26.2.1.1 enum gdcmm::network::EEventID

Enumerator

***eAASSOCIATERequestLocalUser***  
***eTransportConnConfirmLocal***  
***eASSOCIATE\_ACPDUreceived***  
***eASSOCIATE\_RJPDUreceived***  
***eTransportConnIndicLocal***  
***eAASSOCIATE\_RQPDUreceived***  
***eAASSOCIATEResponseAccept***  
***eAASSOCIATEResponseReject***  
***ePDATArequest***  
***ePDATATFPDU***  
***eARELEASERequest***  
***eARELEASE\_RQPDUReceivedOpen***  
***eARELEASE\_RPPDUReceived***  
***eARELEASEResponse***  
***eAABORTRequest***  
***eAABORTPDUReceivedOpen***  
***eTransportConnectionClosed***  
***eARTIMTimerExpired***  
***eUnrecognizedPDUReceived***  
***eEventDoesNotExist***

### 26.2.1.2 enum gdcmm::network::EStateID

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

name and date: 16 sept 2010 mmr

Enumerator

***eStaDoesNotExist***  
***eSta1Idle***  
***eSta2Open***  
***eSta3WaitLocalAssoc***  
***eSta4LocalAssocDone***  
***eSta5WaitRemoteAssoc***  
***eSta6TransferReady***

***eSta7WaitRelease***  
***eSta8WaitLocalRelease***  
***eSta9ReleaseCollisionRqLocal***  
***eSta10ReleaseCollisionAc***  
***eSta11ReleaseCollisionRq***  
***eSta12ReleaseCollisionAcLocal***  
***eSta13AwaitingClose***

## 26.2.2 Function Documentation

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

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

## 26.2.3 Variable Documentation

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

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

Referenced by `gdcn::network::TableRow::TableRow()`, and `gdcn::network::TableRow::~~TableRow()`.

## 26.3 gdcn::SegmentHelper Namespace Reference

### Classes

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

## 26.4 gdcn::terminal Namespace Reference

Class for Terminal Allow one to print in color in a shell.

### 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)

### 26.4.1 Detailed Description

Class for Terminal Allow one to print in color in a shell.

- support VT100 compatible shell
- win32 console

### 26.4.2 Enumeration Type Documentation

#### 26.4.2.1 enum gdcmm::terminal::Attribute

Enumerator

***reset***  
***bright***  
***dim***  
***underline***  
***blink***  
***reverse***  
***hidden***

#### 26.4.2.2 enum gdcmm::terminal::Color

Enumerator

***black***  
***red***  
***green***

*yellow*

*blue*

*magenta*

*cyan*

*white*

#### 26.4.2.3 enum `gdcmm::terminal::Mode`

Enumerator

*CONSOLE*

*VT100*

### 26.4.3 Function Documentation

26.4.3.1 `GDCM_EXPORT std::string gdcmm::terminal::setattribute ( Attribute att )`

26.4.3.2 `GDCM_EXPORT std::string gdcmm::terminal::setbgcolor ( Color c )`

26.4.3.3 `GDCM_EXPORT std::string gdcmm::terminal::setfgcolor ( Color c )`

26.4.3.4 `GDCM_EXPORT void gdcmm::terminal::setmode ( Mode m )`

## Chapter 27

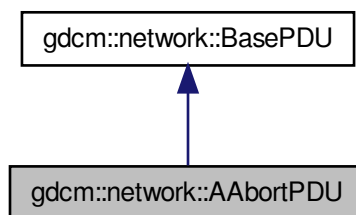
# Class Documentation

### 27.1 gdcm::network::AAabortPDU Class Reference

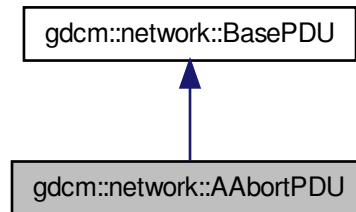
[AAabortPDU](#) [Table 9-26](#) A-ABORT PDU FIELDS.

```
#include <gdcmAAabortPDU.h>
```

Inheritance diagram for `gdcm::network::AAabortPDU`:



Collaboration diagram for `gdcm::network::AAbortPDU`:



## 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

### 27.1.1 Detailed Description

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

### 27.1.2 Constructor & Destructor Documentation

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

### 27.1.3 Member Function Documentation

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

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

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

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

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

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



27.1.3.4 void gdcm::network::AAabortPDU::SetReason ( const uint8\_t r )

27.1.3.5 void gdcm::network::AAabortPDU::SetSource ( const uint8\_t s )

27.1.3.6 size\_t gdcm::network::AAabortPDU::Size ( ) const [virtual]

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

27.1.3.7 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:

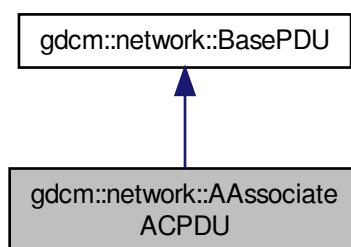
- [gdcmAAabortPDU.h](#)

## 27.2 gdcm::network::AAssociateACPDU Class Reference

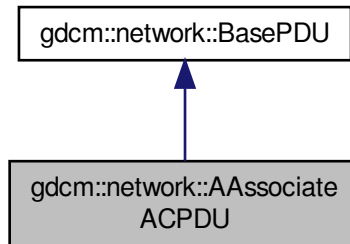
[AAssociateACPDU Table 9-17](#) ASSOCIATE-AC PDU fields.

```
#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 UserInfo & GetUserInfo () 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](#)

## 27.2.1 Detailed Description

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

## 27.2.2 Member Typedef Documentation

27.2.2.1 `typedef std::vector<PresentationContextAC>::size_type gdcmm::network::AAAssociateACPDU::SizeType`

## 27.2.3 Constructor & Destructor Documentation

27.2.3.1 `gdcmm::network::AAAssociateACPDU::AAAssociateACPDU ( )`

## 27.2.4 Member Function Documentation

27.2.4.1 `void gdcmm::network::AAAssociateACPDU::AddPresentationContextAC ( PresentationContextAC const & pcac )`

27.2.4.2 `SizeType gdcmm::network::AAAssociateACPDU::GetNumberOfPresentationContextAC ( ) const [inline]`

27.2.4.3 `const PresentationContextAC& gdcmm::network::AAAssociateACPDU::GetPresentationContextAC ( SizeType i ) [inline]`

27.2.4.4 `const UserInformation& gdcmm::network::AAAssociateACPDU::GetUserInformation ( ) const [inline]`

27.2.4.5 `void gdcmm::network::AAAssociateACPDU::InitFromRQ ( AAAssociateRQPDU const & rqpdu )`

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

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

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

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

27.2.4.8 `std::istream& gdcmm::network::AAAssociateACPDU::Read ( std::istream & is ) [virtual]`

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

27.2.4.9 `void gdcmm::network::AAAssociateACPDU::SetCalledAETitle ( const char calledaetitle[16] ) [protected]`

27.2.4.10 `void gdcmm::network::AAAssociateACPDU::SetCallingAETitle ( const char callingaetitle[16] ) [protected]`

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

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

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

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

## 27.2.5 Friends And Related Function Documentation

### 27.2.5.1 friend class **AAssociateRQPDU** [friend]

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

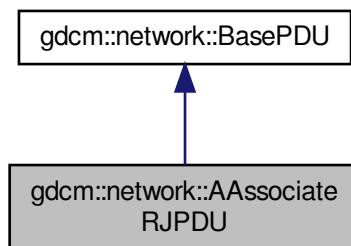
- [gdcmAAssociateACPDU.h](#)

## 27.3 gdcm::network::AAssociateRJPDU Class Reference

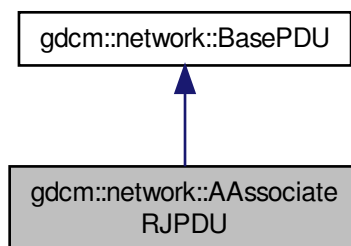
[AAssociateRJPDU](#) Table 9-21 ASSOCIATE-RJ PDU FIELDS.

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

Inheritance diagram for gdcm::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

### 27.3.1 Detailed Description

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

### 27.3.2 Constructor & Destructor Documentation

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

### 27.3.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

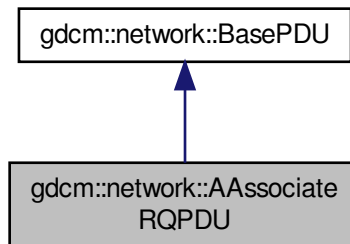
- [gdcmAAssociateRJPDU.h](#)

## 27.4 gdcm::network::AAssociateRQPDU Class Reference

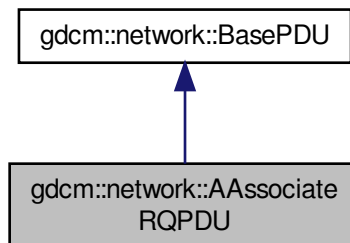
[AAssociateRQPDU Table](#) 9-11 ASSOCIATE-RQ PDU fields.

```
#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 &as) 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](#)

## 27.4.1 Detailed Description

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

## 27.4.2 Member Typedef Documentation

27.4.2.1 `typedef std::vector<PresentationContextRQ> gdcm::network::AAssociateRQPDU::PresentationContextArrayType`

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

## 27.4.3 Constructor & Destructor Documentation

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

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

### 27.4.4 Member Function Documentation

- 27.4.4.1 `void gdcn::network::AAssociateRQPDU::AddPresentationContext ( PresentationContextRQ const & pc )`
- 27.4.4.2 `std::string gdcn::network::AAssociateRQPDU::GetCalledAETitle ( ) const` `[inline]`
- 27.4.4.3 `std::string gdcn::network::AAssociateRQPDU::GetCallingAETitle ( ) const` `[inline]`
- 27.4.4.4 `SizeType gdcn::network::AAssociateRQPDU::GetNumberOfPresentationContext ( ) const` `[inline]`
- 27.4.4.5 `PresentationContextRQ const& gdcn::network::AAssociateRQPDU::GetPresentationContext ( SizeType i ) const` `[inline]`
- 27.4.4.6 `const PresentationContextRQ* gdcn::network::AAssociateRQPDU::GetPresentationContextByAbstractSyntax ( AbstractSyntax const & as ) const`
- 27.4.4.7 `const PresentationContextRQ* gdcn::network::AAssociateRQPDU::GetPresentationContextByID ( uint8_t i ) const`
- 27.4.4.8 `PresentationContextArrayType const& gdcn::network::AAssociateRQPDU::GetPresentationContexts ( )` `[inline]`
- 27.4.4.9 `std::string gdcn::network::AAssociateRQPDU::GetReserved43_74 ( ) const` `[protected]`
- 27.4.4.10 `const UserInformation& gdcn::network::AAssociateRQPDU::GetUserInformation ( ) const` `[inline]`
- 27.4.4.11 `static bool gdcn::network::AAssociateRQPDU::IsAETitleValid ( const char title[16] )` `[static]`

Check whether or not the.

Parameters

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

- 27.4.4.12 `bool gdcn::network::AAssociateRQPDU::IsLastFragment ( ) const` `[inline]`, `[virtual]`

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

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

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

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

- 27.4.4.14 `std::istream& gdcn::network::AAssociateRQPDU::Read ( std::istream & is )` `[virtual]`

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

- 27.4.4.15 `void gdcn::network::AAssociateRQPDU::SetCalledAETitle ( const char calledaetitle[16] )`

Set the Called AE Title.



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

Set the Calling AE Title.

27.4.4.17 void gdcm::network::AAssociateRQPDU::SetUserInformation ( **UserInformation** const & *ui* )

27.4.4.18 size\_t gdcm::network::AAssociateRQPDU::Size ( ) const [virtual]

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

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

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

## 27.4.5 Friends And Related Function Documentation

27.4.5.1 friend class **AAssociateACPDU** [friend]

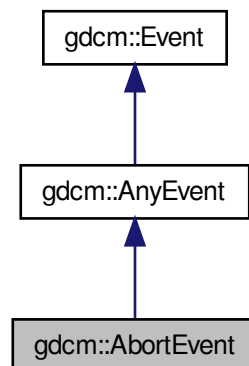
The documentation for this class was generated from the following file:

- [gdcmAAssociateRQPDU.h](#)

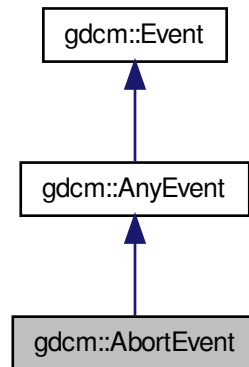
## 27.5 gdcm::AbortEvent Class Reference

```
#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](#)

## 27.6 gdcM::network::AbstractSyntax Class Reference

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

```
#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

#### 27.6.1 Detailed Description

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

## 27.6.2 Constructor & Destructor Documentation

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

## 27.6.3 Member Function Documentation

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

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

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

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

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

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

27.6.3.7 `void gdcm::network::AbstractSyntax::SetNameFromUID ( UIDs::TSName tsname )`

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

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

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

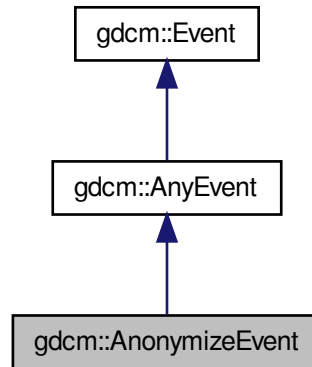
- [gdcmAbstractSyntax.h](#)

## 27.7 gdcm::AnonymizeEvent Class Reference

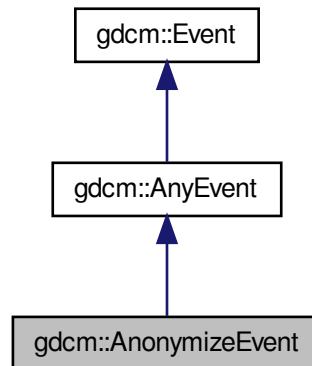
[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

```
#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 [::gdcmm::Event](#) \*e) const
- virtual const char \* [GetEventName](#) () const
- [Tag](#) const & [GetTag](#) () const
- virtual [::gdcmm::Event](#) \* [MakeObject](#) () const
- void [SetTag](#) (const [Tag](#) &t)

### 27.7.1 Detailed Description

[AnonymizeEvent](#) Special type of event triggered during the Anonymization process.

See also

[Anonymizer](#)

### 27.7.2 Member Typedef Documentation

27.7.2.1 typedef [AnonymizeEvent](#) [gdcmm::AnonymizeEvent::Self](#)

27.7.2.2 typedef [AnyEvent](#) [gdcmm::AnonymizeEvent::Superclass](#)

### 27.7.3 Constructor & Destructor Documentation

27.7.3.1 [gdcmm::AnonymizeEvent::AnonymizeEvent](#) ( [Tag](#) const & *tag* = 0 ) [\[inline\]](#)

27.7.3.2 virtual [gdcmm::AnonymizeEvent::~~AnonymizeEvent](#) ( ) [\[inline\]](#),[\[virtual\]](#)

27.7.3.3 [gdcmm::AnonymizeEvent::AnonymizeEvent](#) ( [const Self](#) & *s* ) [\[inline\]](#)

### 27.7.4 Member Function Documentation

27.7.4.1 virtual bool [gdcmm::AnonymizeEvent::CheckEvent](#) ( [const ::gdcmm::Event](#) \* *e* ) const [\[inline\]](#),[\[virtual\]](#)

27.7.4.2 virtual const char\* [gdcmm::AnonymizeEvent::GetEventName](#) ( ) const [\[inline\]](#),[\[virtual\]](#)

Return the StringName associated with the event.

Implements [gdcmm::Event](#).

27.7.4.3 [Tag](#) const& [gdcmm::AnonymizeEvent::GetTag](#) ( ) const [\[inline\]](#)

27.7.4.4 virtual [::gdcmm::Event](#)\* [gdcmm::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 [gdcmm::Event](#).

27.7.4.5 void [gdcmm::AnonymizeEvent::SetTag](#) ( [const Tag](#) & *t* ) [\[inline\]](#)

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

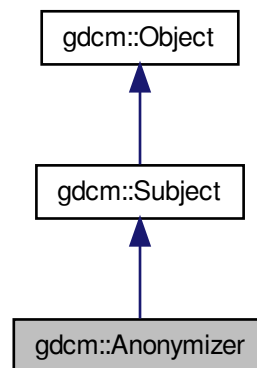
- [gdcmAnonymizeEvent.h](#)

## 27.8 gdcm::Anonymizer Class Reference

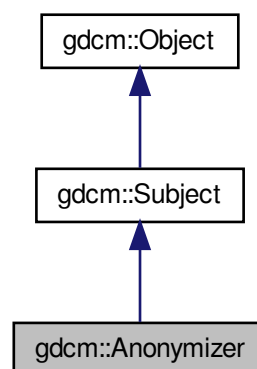
[Anonymizer](#) This class is a multi purpose anonymizer. It can work in 2 mode:

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

Inheritance diagram for gdcm::Anonymizer:



Collaboration diagram for gdcm::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)

### 27.8.1 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 [gdcm::Anonymizer](#) class when anonymizing a [FileSet](#). Once the [gdcm::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](#).

## 27.8.2 Constructor & Destructor Documentation

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

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

## 27.8.3 Member Function Documentation



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

27.8.3.2 `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

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

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

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

Warning

the mapping is definitely lost

27.8.3.5 `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](#).

27.8.3.6 `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](#).

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

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

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

for wrapped language: instantiate a reference counted object

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

27.8.3.11 `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

#### 27.8.3.12 `bool gdcM::Anonymizer::RemoveGroupLength ( )`

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

#### 27.8.3.13 `bool gdcM::Anonymizer::RemovePrivateTags ( )`

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

Examples:

[ClinicalTrialAnnotate.cxx](#).

#### 27.8.3.14 `bool gdcM::Anonymizer::RemoveRetired ( )`

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

#### 27.8.3.15 `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](#).

#### 27.8.3.16 `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

#### 27.8.3.17 `void gdcM::Anonymizer::SetCryptographicMessageSyntax ( CryptographicMessageSyntax * cms )`

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

#### 27.8.3.18 `void gdcM::Anonymizer::SetFile ( const File & f ) [inline]`

Set/Get [File](#).

Examples:

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

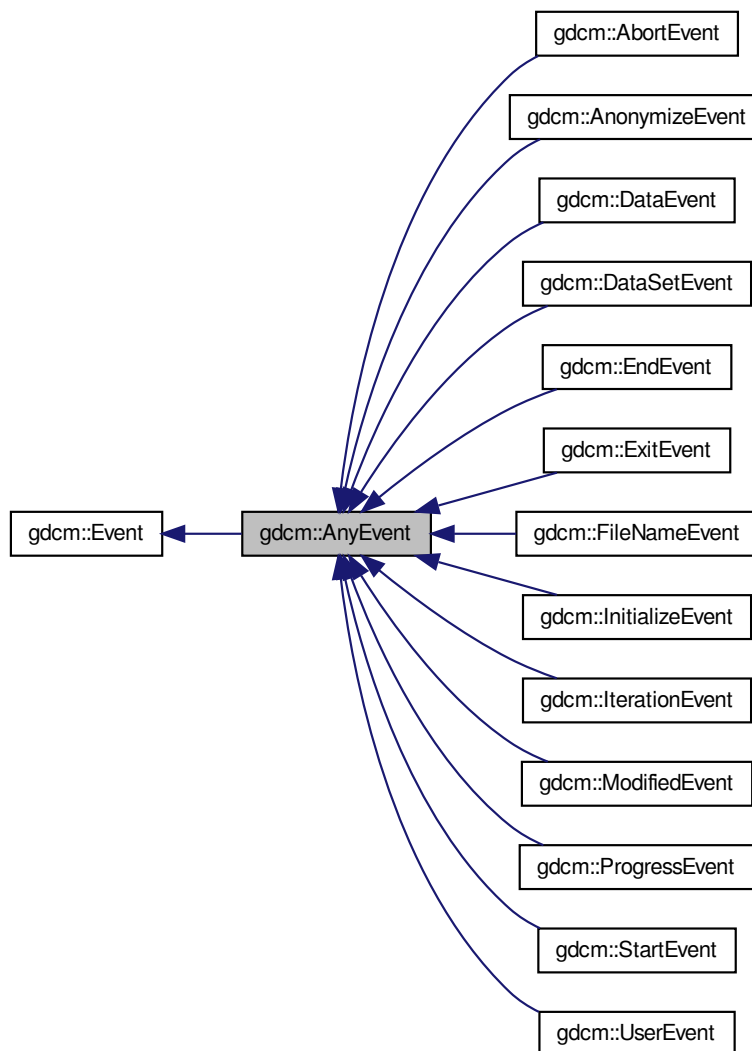
The documentation for this class was generated from the following file:

- [gdcMAnonymizer.h](#)

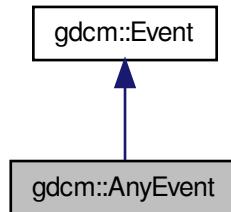
## 27.9 gdcM::AnyEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::AnyEvent:



Collaboration diagram for `gdcm::AnyEvent`:



### Additional Inherited Members

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

- [gdcmEvent.h](#)

## 27.10 `gdcm::network::ApplicationContext` Class Reference

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

```
#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`

### 27.10.1 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 )

## 27.10.2 Constructor & Destructor Documentation

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

## 27.10.3 Member Function Documentation

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

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

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

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

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

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

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

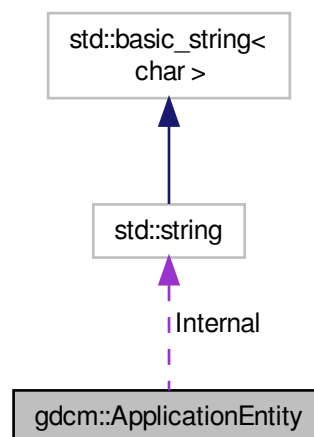
- [gdcmApplicationContext.h](#)

## 27.11 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](#) = ' '

### 27.11.1 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

### 27.11.2 Member Function Documentation

27.11.2.1 bool `gdcm::ApplicationEntity::IsValid ( )` const [inline]

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

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

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

### 27.11.3 Member Data Documentation

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

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

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

27.11.3.4 `const char gdcm::ApplicationEntity::Padding = ''` [static]

27.11.3.5 `const char gdcm::ApplicationEntity::Separator = ''` [static]

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

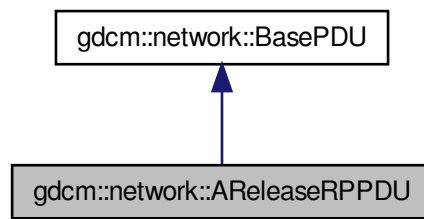
- [gdcmApplicationEntity.h](#)

## 27.12 gdcm::network::AReleaseRPPDU Class Reference

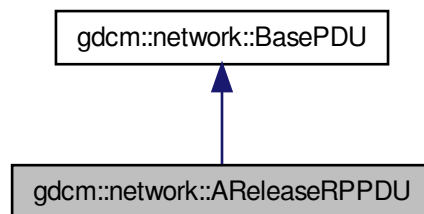
[AReleaseRPPDU](#) Table 9-25 A-RELEASE-RP PDU fields.

```
#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

### 27.12.1 Detailed Description

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

### 27.12.2 Constructor & Destructor Documentation

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

### 27.12.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

- [gdcmAReleaseRPPDU.h](#)

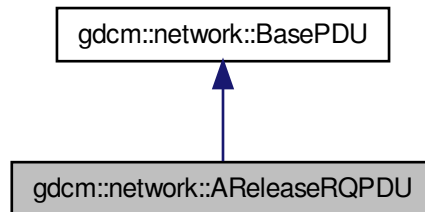
## 27.13 gdcmm::network::AReleaseRQPDU Class Reference

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

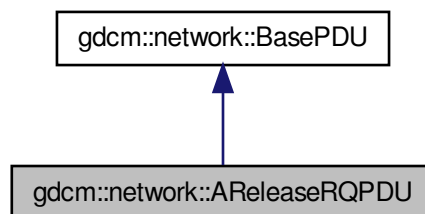
```
#include <gdcmAReleaseRQPDU.h>
```



Inheritance diagram for gdcmm::network::AReleaseRQPDU:



Collaboration diagram for gdcmm::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

### 27.13.1 Detailed Description

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

### 27.13.2 Constructor & Destructor Documentation

27.13.2.1 `gdcn::network::AReleaseRQPDU::AReleaseRQPDU ( )`

### 27.13.3 Member Function Documentation

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

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

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

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

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

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

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

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

27.13.3.5 `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](#)

## 27.14 `gdcn::network::ARTIMTimer` Class Reference

[ARTIMTimer](#) This file contains the code for the ARTIM timer.

```
#include <gdcnARTIMTimer.h>
```

### Public Member Functions

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

### 27.14.1 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.

### 27.14.2 Constructor & Destructor Documentation

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

### 27.14.3 Member Function Documentation

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

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

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

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

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

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

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

- [gdcmARTIMTimer.h](#)

## 27.15 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](#) ()

### 27.15.1 Detailed Description

Class for [ASN1](#).

### 27.15.2 Constructor & Destructor Documentation

27.15.2.1 `gdcn::ASN1::ASN1 ( )`

27.15.2.2 `gdcn::ASN1::~~ASN1 ( )`

### 27.15.3 Member Function Documentation

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

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

27.15.3.3 `int gdcn::ASN1::TestPBKDF2 ( )` `[protected]`

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

- [gdcnASN1.h](#)

## 27.16 gdcn::network::AsynchronousOperationsWindowSub Class Reference

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

```
#include <gdcnAsynchronousOperationsWindowSub.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

### 27.16.1 Detailed Description

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

## 27.16.2 Constructor & Destructor Documentation

27.16.2.1 gdcmm::network::AsynchronousOperationsWindowSub::AsynchronousOperationsWindowSub ( )

## 27.16.3 Member Function Documentation

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

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

27.16.3.3 size\_t gdcmm::network::AsynchronousOperationsWindowSub::Size ( ) const

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

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

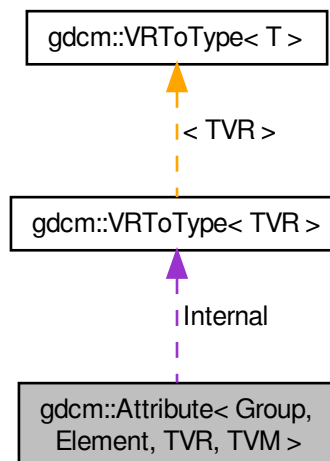
- [gdcmmAsynchronousOperationsWindowSub.h](#)

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

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

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

Collaboration diagram for gdcmm::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)

### 27.17.1 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:

[CreateJIPIDataSet.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenFakeIdentifyFile.cxx](#), [GetSequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndPrintAttributes.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), and [VolumeSorter.cxx](#).

### 27.17.2 Member Typedef Documentation

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

### 27.17.3 Member Enumeration Documentation

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

Enumerator

***VMType***

### 27.17.4 Member Function Documentation

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

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

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

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

References gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcmm::DataElement::GetVR(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcmm::Attribute< Group, Element, TVR, TVM >::Internal, gdcmm::DataElement::SetByteValue(), gdcmm::DataElement::SetVR(), gdcmm::VR::SQ, gdcmm::VR::UI, and gdcmm::VR::VRASCII.

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

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

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

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::GetValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator<(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<(), gdcmm::Attribute< Group, Element, TVR, TVM >::operator==(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==(), gdcmm::Attribute< Group, Element, TVR, TVM >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetValues().

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

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Print(), gdcmm::Attribute< Group, Element, TVR, TVM >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Set(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetFromDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetFromDataSet().



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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator[]()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::operator[]()`.

27.17.4.10 `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 >::GetValue ( unsigned int idx = 0 ) const [inline]`

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator==()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`.

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`.

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

27.17.4.15 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.17.4.16 `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()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.17.4.17 `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]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetValue()`.

27.17.4.18 `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 >::operator[] ( unsigned int idx ) const [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetValue()`.

27.17.4.19 `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 >::Print ( std::ostream & os ) const [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.17.4.20 `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 >::Set ( DataSet const & ds ) [inline]`

References `gdcM::DataSet::GetDataElement()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

27.17.4.21 `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 >::SetByteValue ( const ByteValue * bv ) [inline], [protected]`

References `gdcM::ByteValue::GetLength()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcM::ByteValue::GetPointer()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

Referenced by `gdcM::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`, and `gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`.

```
27.17.4.22 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(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), gdcmm::ByteValue::GetPointer(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

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

```
27.17.4.23 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
>::SetFromDataElement ( DataElement const & de ) [inline]
```

References gdcmm::DataElement::GetByteValue(), gdcmm::Tag::GetGroup(), gdcmm::DataElement::GetTag(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcmm::DataElement::GetVR(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR(), gdcmm::VR::INVALID, gdcmm::DataElement::IsEmpty(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap(), and gdcmm::VR::UN.

Referenced by gdcmm::Attribute< Group, Element, TVR, TVM >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set(), gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::Set(), gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetFromDataSet().

```
27.17.4.24 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 >::SetFromDataSet (
DataSet const & ds ) [inline]
```

References gdcmm::DataSet::FindDataElement(), gdcmm::DataSet::GetDataElement(), gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag(), gdcmm::DataElement::IsEmpty(), and gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement().

```
27.17.4.25 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 >::SetValue (
ArrayType v, unsigned int idx = 0 ) [inline]
```

References gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

```
27.17.4.26 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 >::SetValues ( const
ArrayType * array, unsigned int numel = VMType ) [inline]
```

Examples:

[LargeVRDSEExplicit.cxx](#).

References gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues(), and gdcmm::Attribute< Group, Element, TVR, TVM >::Internal.

Referenced by gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetByteValue(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::SetNumberOfValues().

### 27.17.5 Member Data Documentation

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

Referenced by `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator!=()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator<()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::operator==()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::operator==()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Print()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Print()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetByteValueNoSwap()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetValues()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetValues()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute()`.

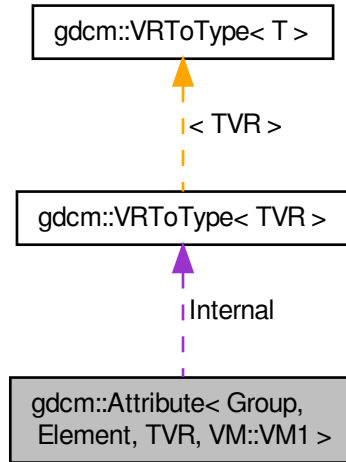
The documentation for this class was generated from the following file:

- [gdcmmAttribute.h](#)

## 27.18 `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >` Class Template Reference

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

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



## Public Types

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

## Public Member Functions

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

### 27.18.1 Member Typedef Documentation

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

### 27.18.2 Member Enumeration Documentation

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

#### Enumerator

***VMType***

### 27.18.3 Member Function Documentation

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

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

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

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

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcmm::DataElement::GetVR()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR()`,

gdcm::Attribute< Group, Element, TVR, TVM >::Internal, gdcm::DataElement::SetByteValue(), gdcm::DataElement::SetVR(), gdcm::VR::SQ, gdcm::VR::UI, and gdcm::VR::VRASCII.

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

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

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

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

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

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

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

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

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

References gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

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

References gdcm::VM::VM1.

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

27.18.3.15 `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 >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

27.18.3.16 `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 >::GetNumberOfValues(), gdcm::Attribute< Group, Element, TVR, TVM >::GetValues(), and gdcm::Attribute< Group, Element, TVR, TVM >::Internal.

**27.18.3.17** `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()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetValues()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

**27.18.3.19** `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()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

**27.18.3.20** `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()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::ByteValue::GetPointer()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

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

References `gdcmm::ByteValue::GetLength()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcmm::ByteValue::GetPointer()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.

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

References `gdcmm::DataElement::GetByteValue()`, `gdcmm::Tag::GetGroup()`, `gdcmm::DataElement::GetTag()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcmm::DataElement::GetVR()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcmm::VR::INVALID`, `gdcmm::DataElement::IsEmpty()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetByteValueNoSwap()`, and `gdcmm::VR::UN`.

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

References `gdcmm::DataSet::FindDataElement()`, `gdcmm::DataSet::GetDataElement()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcmm::DataElement::IsEmpty()`, and `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

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

References `gdcmm::Attribute< Group, Element, TVR, TVM >::Internal`.



#### 27.18.4 Member Data Documentation

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

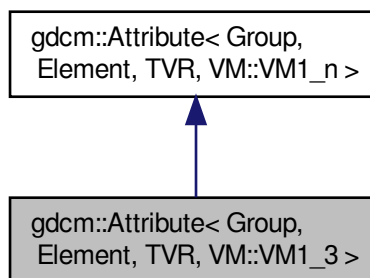
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

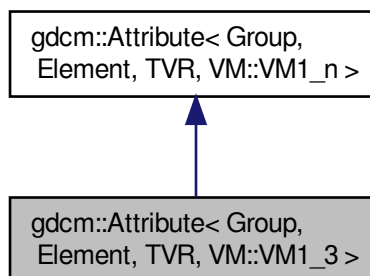
### 27.19 gdcM::Attribute< Group, Element, TVR, VM::VM1\_3 > Class Template Reference

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

Inheritance diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1\_3 >:



Collaboration diagram for gdcM::Attribute< Group, Element, TVR, VM::VM1\_3 >:



## Public Member Functions

- [VM GetVM](#) () const

## Additional Inherited Members

### 27.19.1 Member Function Documentation

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

References `gdcM::VM::VM1_3`.

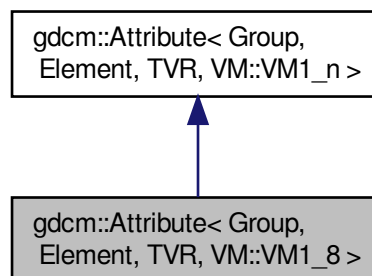
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

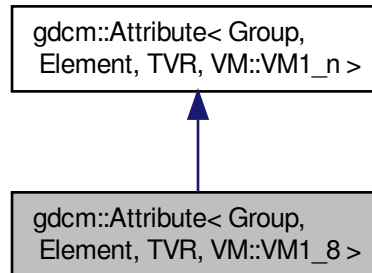
## 27.20 gdcM::Attribute< Group, Element, TVR, VM::VM1\_8 > Class Template Reference

```
#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

### 27.20.1 Member Function Documentation

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

References `gdcm::VM::VM1_8`.

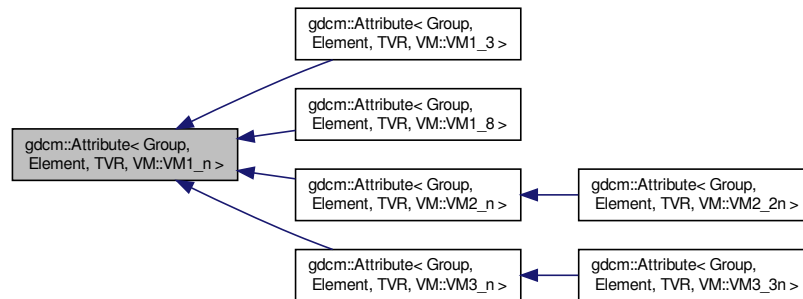
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

## 27.21 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)

### 27.21.1 Member Typedef Documentation

27.21.1.1 `template<uint16_t Group, uint16_t Element, int TVR> typedef VRToType<TVR>::Type gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::ArrayType`

### 27.21.2 Constructor & Destructor Documentation

27.21.2.1 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::Attribute ( ) [inline],[explicit]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.2.2 `template<uint16_t Group, uint16_t Element, int TVR> gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::~~Attribute ( ) [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

### 27.21.3 Member Function Documentation

27.21.3.1 `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)) )`

27.21.3.2 `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)) )`

27.21.3.3 `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)) )`

27.21.3.4 `template<uint16_t Group, uint16_t Element, int TVR> DataElement gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

27.21.3.5 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVM ( ) [inline],[static]`

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

27.21.3.6 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::GetDictVR ( ) [inline],[static]`

27.21.3.7 `template<uint16_t Group, uint16_t Element, int TVR> unsigned int gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetNumberOfValues ( ) const [inline]`

27.21.3.8 `template<uint16_t Group, uint16_t Element, int TVR> static Tag gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetTag ( ) [inline], [static]`

27.21.3.9 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetValue ( unsigned int idx = 0 ) [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.3.10 `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]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.3.11 `template<uint16_t Group, uint16_t Element, int TVR> const ArrayType* gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetValues ( ) const [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.3.12 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetVM ( ) [inline], [static]`

References `gdcM::VM::VM1_n`.

27.21.3.13 `template<uint16_t Group, uint16_t Element, int TVR> static VR gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::GetVR ( ) [inline], [static]`

27.21.3.14 `template<uint16_t Group, uint16_t Element, int TVR> ArrayType& gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::operator[] ( unsigned int idx ) [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetValue()`.

27.21.3.15 `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]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetValue()`.

27.21.3.16 `template<uint16_t Group, uint16_t Element, int TVR> void gdcM::Attribute< Group, Element, TVR, VM::VM1_n >::Print ( std::ostream & os ) const [inline]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetVM()`, `gdcM::Attribute< Group, Element, TVR, TVM >::GetVR()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.3.17 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

27.21.3.18 `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()`, `gdcm::ByteValue::GetPointer()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetValues()`.

27.21.3.19 `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::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::GetVR()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetVR()`, `gdcm::DataElement::IsEmpty()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetByteValue()`.

27.21.3.20 `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()`, `gdcm::Attribute< Group, Element, TVR, TVM >::GetTag()`, `gdcm::DataElement::IsEmpty()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`.

27.21.3.21 `template<uint16_t Group, uint16_t Element, int TVR> void gdcm::Attribute< Group, Element, TVR, VM::VM1_n >::SetNumberOfValues ( unsigned int numel ) [inline]`

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

27.21.3.22 `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]`

References `gdcm::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcm::Attribute< Group, Element, TVR, TVM >::Internal`.

27.21.3.23 `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()`.

27.21.3.24 `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]`

References `gdcM::Attribute< Group, Element, TVR, TVM >::GetNumberOfValues()`, and `gdcM::Attribute< Group, Element, TVR, TVM >::Internal`.

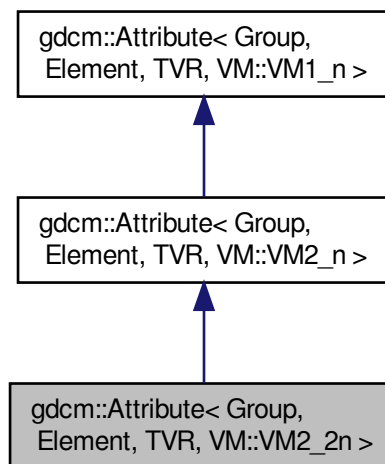
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

## 27.22 `gdcM::Attribute< Group, Element, TVR, VM::VM2_2n >` Class Template Reference

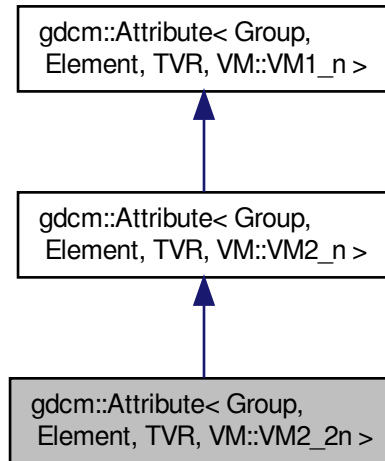
```
#include <gdcMAttribute.h>
```

Inheritance diagram for `gdcM::Attribute< Group, Element, TVR, VM::VM2_2n >`:





Collaboration diagram for gdcm::Attribute< Group, Element, TVR, VM::VM2\_n >:



### Static Public Member Functions

- static [VM GetVM](#) ()

### Additional Inherited Members

#### 27.22.1 Member Function Documentation

27.22.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >::GetVM ( ) [inline], [static]`

References `gdcm::VM::VM2_2n`.

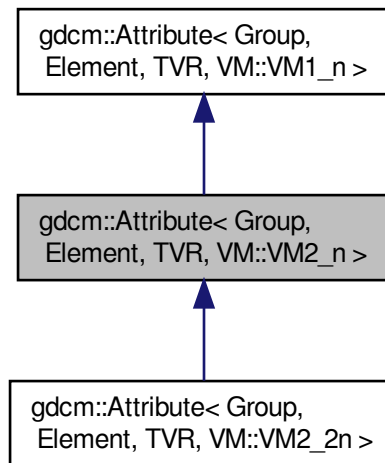
The documentation for this class was generated from the following file:

- [gdcmAttribute.h](#)

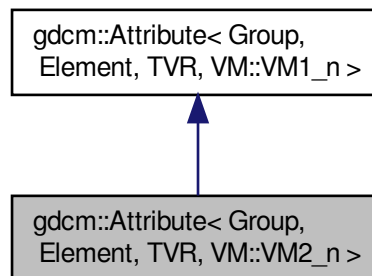
## 27.23 gdcm::Attribute< Group, Element, TVR, VM::VM2\_n > Class Template Reference

```
#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

### 27.23.1 Member Function Documentation

```
27.23.1.1  template<uint16_t Group, uint16_t Element, int TVR> VM gdcM::Attribute< Group, Element, TVR, VM::VM2_n
>::GetVM( ) const  [inline]
```

References gdcM::VM::VM2\_n.

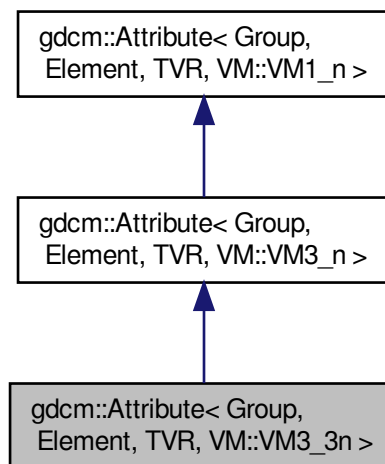
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

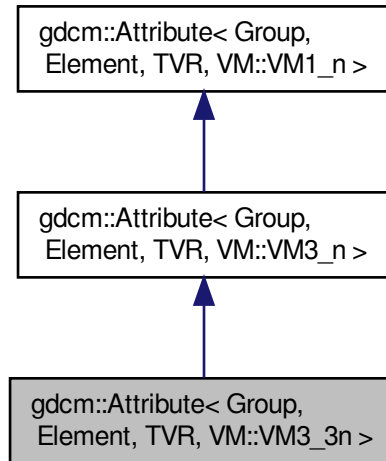
## 27.24 gdcM::Attribute< Group, Element, TVR, VM::VM3\_3n > Class Template Reference

```
#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

### 27.24.1 Member Function Documentation

27.24.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_3n >::GetVM ( ) [inline], [static]`

References `gdcM::VM::VM3_3n`.

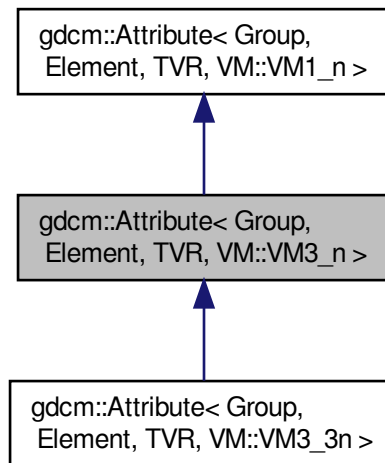
The documentation for this class was generated from the following file:

- [gdcMAttribute.h](#)

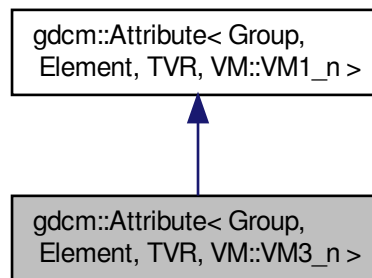
## 27.25 `gdcM::Attribute< Group, Element, TVR, VM::VM3_n >` Class Template Reference

```
#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

#### 27.25.1 Member Function Documentation

27.25.1.1 `template<uint16_t Group, uint16_t Element, int TVR> static VM gdcM::Attribute< Group, Element, TVR, VM::VM3_n >::GetVM( ) [inline],[static]`

References `gdcM::VM::VM3_n`.

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

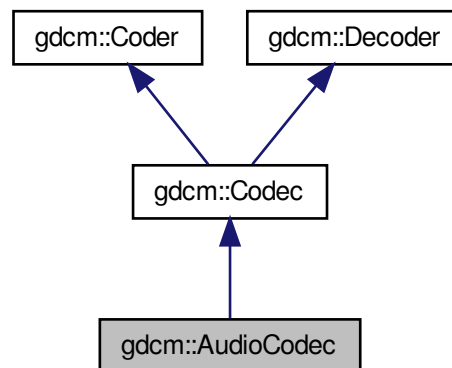
- [gdcMAttribute.h](#)

## 27.26 gdcM::AudioCodec Class Reference

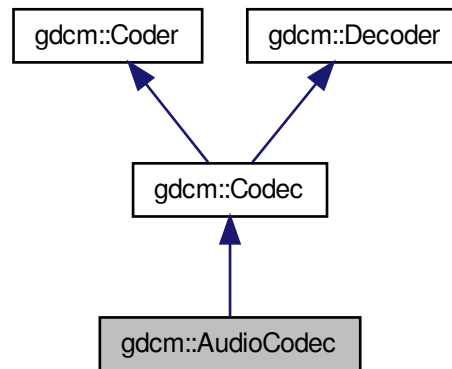
[AudioCodec](#).

```
#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

### 27.26.1 Detailed Description

[AudioCodec](#).

### 27.26.2 Constructor & Destructor Documentation

27.26.2.1 `gdcm::AudioCodec::AudioCodec ( )`

27.26.2.2 `gdcm::AudioCodec::~~AudioCodec ( )`

### 27.26.3 Member Function Documentation

27.26.3.1 `bool gdcm::AudioCodec::CanCode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

**27.26.3.2** `bool gdcm::AudioCodec::CanDecode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

**27.26.3.3** `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](#)

## 27.27 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`)

### 27.27.1 Detailed Description

Class for [Base64](#).

### 27.27.2 Member Function Documentation

**27.27.2.1** `static size_t gdcm::Base64::Decode ( char * dst, size_t dlen, const char * src, size_t slen )` `[static]`

Decode a base64-formatted buffer.

Parameters

<code>dst</code>	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](#).

**27.27.2.2** `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

**27.27.2.3** `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](#).

**27.27.2.4** `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:

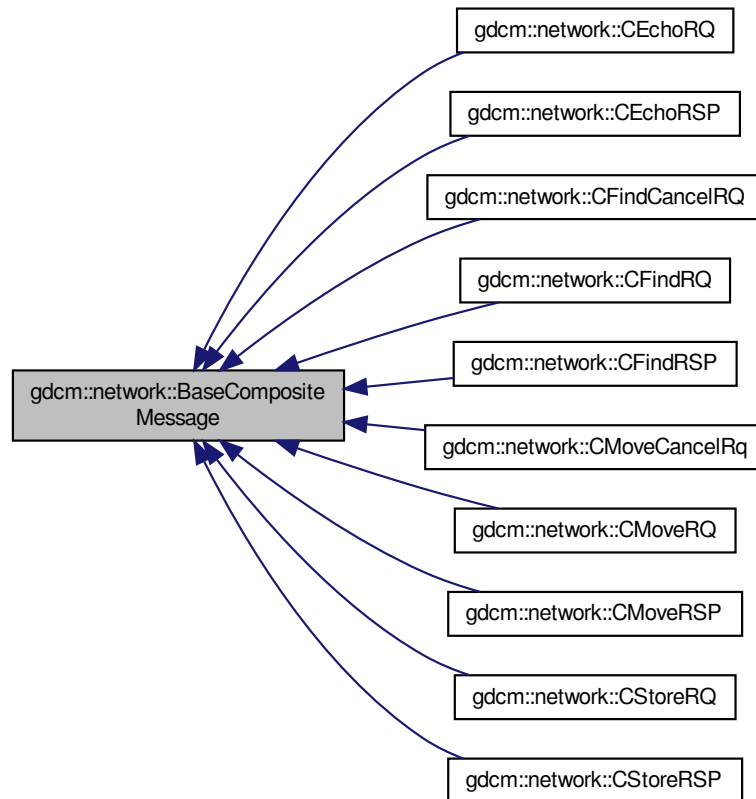
- [gdcmBase64.h](#)

## 27.28 gdcm::network::BaseCompositeMessage Class Reference

[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.

```
#include <gdcmBaseCompositeMessage.h>
```

Inheritance diagram for `gdcn::network::BaseCompositeMessage`:



## Public Member Functions

- virtual `~BaseCompositeMessage()`
- virtual `std::vector< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)=0`

### 27.28.1 Detailed Description

**BaseCompositeMessage** The Composite events described in section 3.7-2009 of the DICOM standard all use their own messages. These messages are constructed using Presentation Data Values, from section 3.8-2009 of the standard, and then fill in appropriate values in their datasets.

So, for the five composites:

- C-ECHO
- C-FIND

- C-MOVE
- C-GET
- C-STORE there are a series of messages. However, all of these messages are obtained as part of a PDataPDU, and all have to be placed there. Therefore, since they all have shared functionality and construction tropes, that will be put into a base class. Further, the base class will be then returned by the factory class, gdcmCompositePDUFactory.

This is an abstract class. It cannot be instantiated on its own.

## 27.28.2 Constructor & Destructor Documentation

27.28.2.1 `virtual gdcm::network::BaseCompositeMessage::~BaseCompositeMessage ( ) [inline], [virtual]`

## 27.28.3 Member Function Documentation

27.28.3.1 `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](#).

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

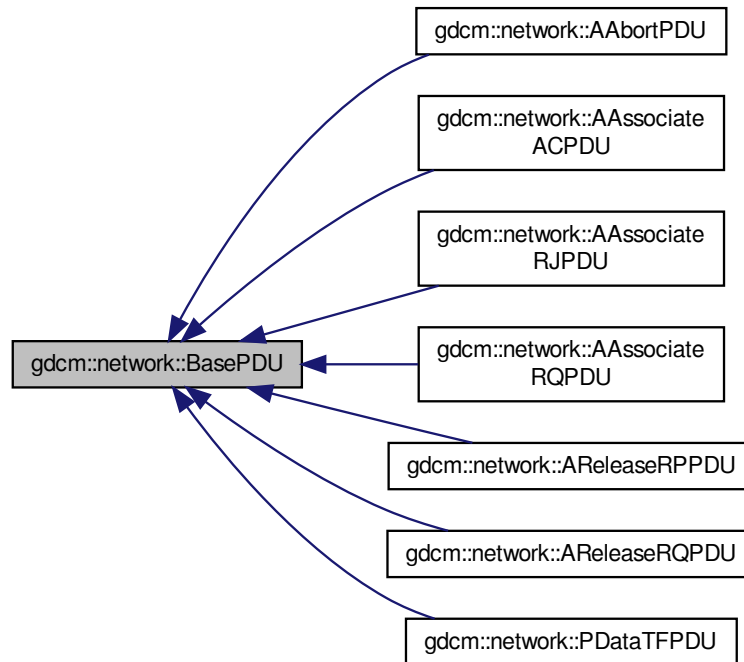
- [gdcmBaseCompositeMessage.h](#)

## 27.29 gdcm::network::BasePDU Class Reference

[BasePDU](#) base class for PDUs.

```
#include <gdcmBasePDU.h>
```

Inheritance diagram for `gdc::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

### 27.29.1 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

## 27.29.2 Constructor & Destructor Documentation

27.29.2.1 `virtual gdcm::network::BasePDU::~BasePDU ( ) [inline], [virtual]`

## 27.29.3 Member Function Documentation

27.29.3.1 `virtual bool gdcm::network::BasePDU::IsLastFragment ( ) const [pure virtual]`

Implemented in [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAAbortPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AReleaseRPPDU](#), and [gdcm::network::AReleaseRQPDU](#).

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

Implemented in [gdcm::network::AAssociateRQPDU](#), [gdcm::network::AAssociateACPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAAbortPDU](#), [gdcm::network::AReleaseRPPDU](#), [gdcm::network::AReleaseRQPDU](#), and [gdcm::network::AAssociateRJPDU](#).

27.29.3.3 `virtual std::istream& gdcm::network::BasePDU::Read ( std::istream & is ) [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](#).

27.29.3.4 `virtual size_t gdcm::network::BasePDU::Size ( ) const [pure virtual]`

Implemented in [gdcm::network::AAssociateACPDU](#), [gdcm::network::AAssociateRQPDU](#), [gdcm::network::PDataTFPDU](#), [gdcm::network::AAAbortPDU](#), [gdcm::network::AAssociateRJPDU](#), [gdcm::network::AReleaseRPPDU](#), and [gdcm::network::AReleaseRQPDU](#).

27.29.3.5 `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](#).

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

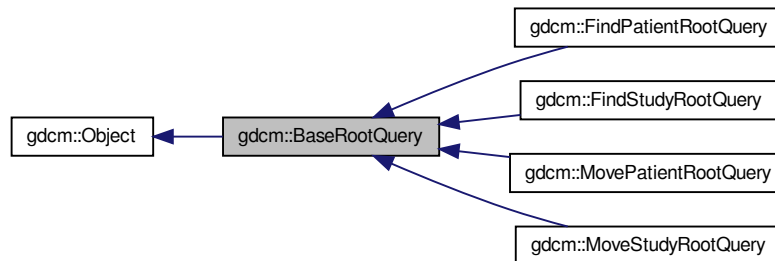
- [gdcmBasePDU.h](#)

## 27.30 gdcm::BaseRootQuery Class Reference

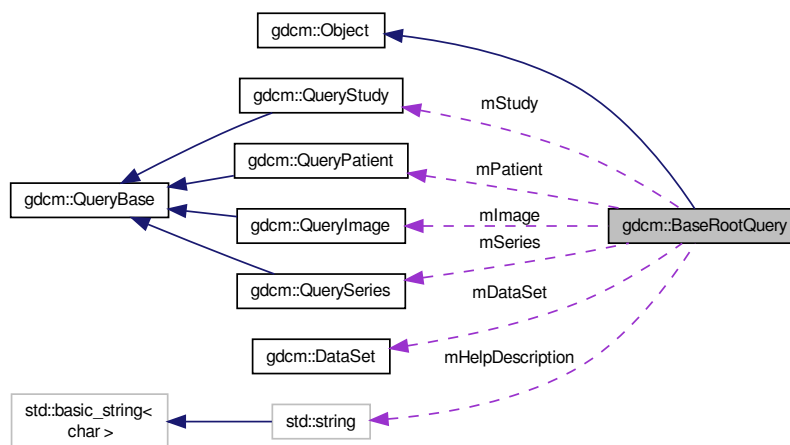
[BaseRootQuery](#) contains: a baseclass which will produce a dataset for c-find and c-move with patient/study root.

```
#include <gdcmBaseRootQuery.h>
```

Inheritance diagram for `gdc::BaseRootQuery`:



Collaboration diagram for `gdc::BaseRootQuery`:



## Public Member Functions

- virtual `~BaseRootQuery ()`
- 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 ()`
- `EQueryLevel` `GetQueryLevelFromQueryRoot (ERootType roottype)`
- virtual `std::vector< Tag > GetTagListByLevel (const EQueryLevel &inQueryLevel)=0`
- virtual void `InitializeDataSet (const EQueryLevel &inQueryLevel)=0`
- 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)
- virtual bool [ValidateQuery](#) (bool inStrict=true) const =0
- const std::ostream & [WriteHelpFile](#) (std::ostream &os)
- bool [WriteQuery](#) (const std::string &inFileName)

### 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](#) ()
- void [SetSearchParameter](#) (const [Tag](#) &inTag, const [DictEntry](#) &inDictEntry, const std::string &inValue)

### Protected Attributes

- [DataSet](#) mDataSet
- std::string mHelpDescription
- [QueryImage](#) mImage
- [QueryPatient](#) mPatient
- [ERootType](#) mRootType
- [QuerySeries](#) mSeries
- [QueryStudy](#) mStudy

### Friends

- class [QueryFactory](#)

## 27.30.1 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.

## 27.30.2 Constructor & Destructor Documentation

27.30.2.1 `gdcm::BaseRootQuery::BaseRootQuery ( )` [protected]

27.30.2.2 `virtual gdcm::BaseRootQuery::~BaseRootQuery ( )` [virtual]

### 27.30.3 Member Function Documentation

27.30.3.1 `void gdcmm::BaseRootQuery::AddQueryDataSet ( const DataSet & ds )`

27.30.3.2 `static QueryBase* gdcmm::BaseRootQuery::Construct ( ERootType inRootType, EQueryLevel qllevel )`  
[static]

27.30.3.3 `virtual UIDs::TSName gdcmm::BaseRootQuery::GetAbstractSyntaxUID ( ) const` [pure virtual]

Implemented in [gdcmm::FindStudyRootQuery](#), [gdcmm::MovePatientRootQuery](#), [gdcmm::MoveStudyRootQuery](#), and [gdcmm::FindPatientRootQuery](#).

27.30.3.4 `DataSet const& gdcmm::BaseRootQuery::GetQueryDataSet ( ) const`

Set/Get the internal representation of the query as a [DataSet](#).

27.30.3.5 `DataSet& gdcmm::BaseRootQuery::GetQueryDataSet ( )`

27.30.3.6 `EQueryLevel gdcmm::BaseRootQuery::GetQueryLevelFromQueryRoot ( ERootType roottype )`

27.30.3.7 `static int gdcmm::BaseRootQuery::GetQueryLevelFromString ( const char * str )` [static]

27.30.3.8 `static const char* gdcmm::BaseRootQuery::GetQueryLevelString ( EQueryLevel ql )` [static]

27.30.3.9 `virtual std::vector<Tag> gdcmm::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 [gdcmm::FindPatientRootQuery](#), [gdcmm::FindStudyRootQuery](#), [gdcmm::MovePatientRootQuery](#), and [gdcmm::MoveStudyRootQuery](#).

27.30.3.10 `virtual void gdcmm::BaseRootQuery::InitializeDataSet ( const EQueryLevel & inQueryLevel )` [pure virtual]

this function sets tag 8,52 to the appropriate value based on query level also fills in the right unique tags, as per the standard's requirements should allow for connection with dcmTk

Implemented in [gdcmm::FindPatientRootQuery](#), [gdcmm::FindStudyRootQuery](#), [gdcmm::MovePatientRootQuery](#), and [gdcmm::MoveStudyRootQuery](#).

27.30.3.11 `void gdcmm::BaseRootQuery::Print ( std::ostream & os ) const` [virtual]

Reimplemented from [gdcmm::Object](#).

27.30.3.12 `void gdcmm::BaseRootQuery::SetSearchParameter ( const Tag & inTag, const DictEntry & inDictEntry, const std::string & inValue )` [protected]

27.30.3.13 `void gdcmm::BaseRootQuery::SetSearchParameter ( const Tag & inTag, const std::string & inValue )`



27.30.3.14 `void gdcm::BaseRootQuery::SetSearchParameter ( const std::string & inKeyword, const std::string & inValue )`

27.30.3.15 `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.

Implemented in [gdcm::FindStudyRootQuery](#), [gdcm::MovePatientRootQuery](#), [gdcm::MoveStudyRootQuery](#), and [gdcm::FindPatientRootQuery](#).

27.30.3.16 `const std::ostream& gdcm::BaseRootQuery::WriteHelpFile ( std::ostream & os )`

27.30.3.17 `bool gdcm::BaseRootQuery::WriteQuery ( const std::string & inFileName )`

## 27.30.4 Friends And Related Function Documentation

27.30.4.1 `friend class QueryFactory` [friend]

## 27.30.5 Member Data Documentation

27.30.5.1 `DataSet gdcm::BaseRootQuery::mDataSet` [protected]

27.30.5.2 `std::string gdcm::BaseRootQuery::mHelpDescription` [protected]

27.30.5.3 `QueryImage gdcm::BaseRootQuery::mImage` [protected]

27.30.5.4 `QueryPatient gdcm::BaseRootQuery::mPatient` [protected]

27.30.5.5 `ERootType gdcm::BaseRootQuery::mRootType` [protected]

27.30.5.6 `QuerySeries gdcm::BaseRootQuery::mSeries` [protected]

27.30.5.7 `QueryStudy gdcm::BaseRootQuery::mStudy` [protected]

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

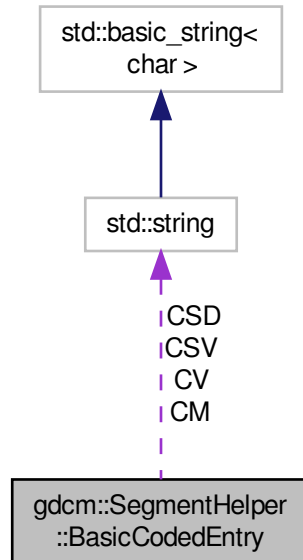
- [gdcmBaseRootQuery.h](#)

## 27.31 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](#)

### 27.31.1 Detailed Description

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

See also

PS 3.3 section 8.8.

### 27.31.2 Constructor & Destructor Documentation

**27.31.2.1** `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( ) [inline]`

Constructor.

**27.31.2.2** `gdcm::SegmentHelper::BasicCodedEntry::BasicCodedEntry ( const char * a_CV, const char * a_CSD, const char * a_CM ) [inline]`

constructor which defines type 1 attributes.

**27.31.2.3** `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.

### 27.31.3 Member Function Documentation

**27.31.3.1** `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.
--------------------------------	--------------------------------

### 27.31.4 Member Data Documentation

**27.31.4.1** `std::string gdcm::SegmentHelper::BasicCodedEntry::CM`

Coding Scheme [Version](#) attribute.

**27.31.4.2** `std::string gdcm::SegmentHelper::BasicCodedEntry::CSD`

Code [Value](#) attribute.

**27.31.4.3** `std::string gdcm::SegmentHelper::BasicCodedEntry::CSV`

Coding Scheme Designator attribute.

#### 27.31.4.4 `std::string gdcM::SegmentHelper::BasicCodedEntry::CV`

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

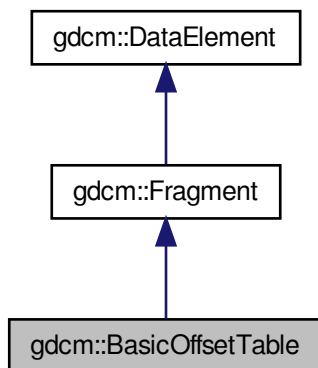
- [gdcMSegmentHelper.h](#)

## 27.32 `gdcM::BasicOffsetTable` Class Reference

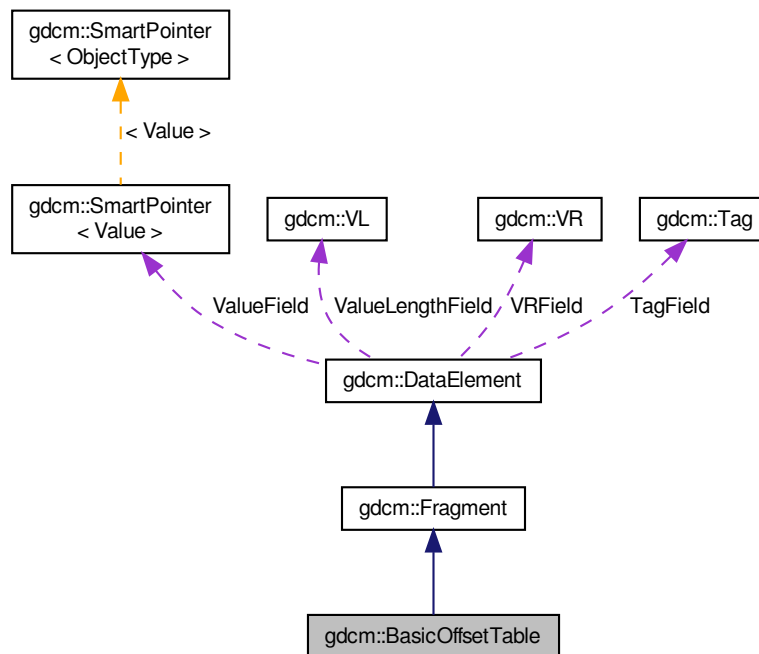
Class to represent a [BasicOffsetTable](#).

```
#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

### 27.32.1 Detailed Description

Class to represent a [BasicOffsetTable](#).

### 27.32.2 Constructor & Destructor Documentation

#### 27.32.2.1 gdcm::BasicOffsetTable::BasicOffsetTable ( ) [inline]

### 27.32.3 Member Function Documentation

27.32.3.1 `template<typename TSwap > std::istream& gdcmm::BasicOffsetTable::Read ( std::istream & is ) [inline]`

References `gdcmmDebugMacro`.

### 27.32.4 Friends And Related Function Documentation

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

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

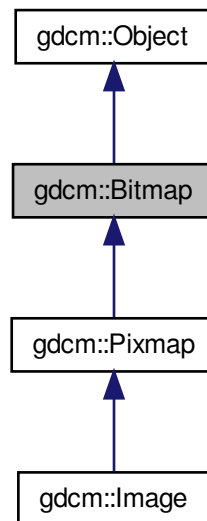
- [gdcmmBasicOffsetTable.h](#)

## 27.33 gdcmm::Bitmap Class Reference

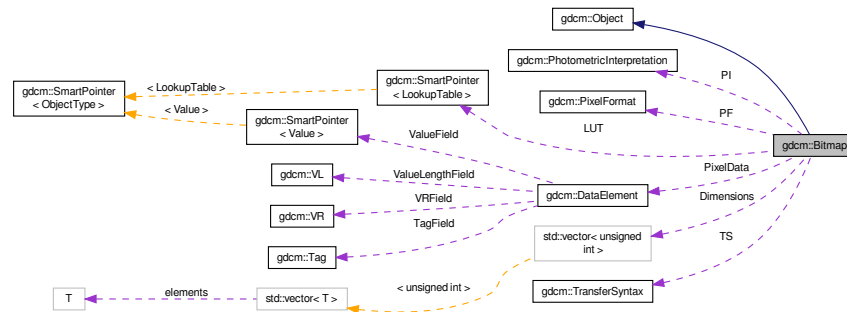
**Bitmap** class A bitmap based image. Used as parent for both `IconImage` and the main Pixel Data `Image` It does not contains any World Space information (IPP, IOP)

```
#include <gdcmmBitmap.h>
```

Inheritance diagram for `gdcmm::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.*

## 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 [TryRLECodec](#) (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](#)

### 27.33.1 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](#).

### 27.33.2 Member Typedef Documentation

27.33.2.1 `typedef SmartPointer<LookupTable> gdcm::Bitmap::LUTPtr` `[protected]`

### 27.33.3 Constructor & Destructor Documentation

27.33.3.1 `gdcm::Bitmap::Bitmap ( )`

27.33.3.2 `gdcm::Bitmap::~~Bitmap ( )`

### 27.33.4 Member Function Documentation

27.33.4.1 `virtual bool gdcm::Bitmap::AreOverlaysInPixelData ( ) const` `[inline],[virtual]`

Reimplemented in [gdcm::Pixmap](#).

27.33.4.2 `void gdcm::Bitmap::Clear ( )`

27.33.4.3 `bool gdcm::Bitmap::ComputeLossyFlag ( )` `[protected]`

27.33.4.4 `bool gdcm::Bitmap::GetBuffer ( char * buffer ) const`

Acces the raw data.

Examples:

[ConvertToQImage.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.33.4.5 `bool gdcm::Bitmap::GetBuffer2 ( std::ostream & os ) const` `[protected]`

27.33.4.6 `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](#), [GenFakelImage.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

27.33.4.7 `unsigned int gdcm::Bitmap::GetColumns ( ) const [inline]`

27.33.4.8 `const DataElement& gdcm::Bitmap::GetDataElement ( ) const [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.33.4.9 `DataElement& gdcm::Bitmap::GetDataElement ( ) [inline]`

27.33.4.10 `unsigned int gdcm::Bitmap::GetDimension ( unsigned int idx ) const`

27.33.4.11 `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](#).

27.33.4.12 `const LookupTable& gdcm::Bitmap::GetLUT ( ) const [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.33.4.13 `LookupTable& gdcm::Bitmap::GetLUT ( ) [inline]`

27.33.4.14 `bool gdcm::Bitmap::GetNeedByteSwap ( ) const [inline]`

27.33.4.15 `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](#).

27.33.4.16 `const PhotometricInterpretation& gdcm::Bitmap::GetPhotometricInterpretation ( ) const`

return the photometric interpretation

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), and [HelloVizWorld.cxx](#).

27.33.4.17 `const PixelFormat& gdcm::Bitmap::GetPixelFormat ( ) const` `[inline]`

Get/Set [PixelFormat](#).

Examples:

[ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakelImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), and [threadgdcm.cxx](#).

27.33.4.18 `PixelFormat& gdcm::Bitmap::GetPixelFormat ( )` `[inline]`

27.33.4.19 `unsigned int gdcm::Bitmap::GetPlanarConfiguration ( ) const`

return the planar configuration

27.33.4.20 `unsigned int gdcm::Bitmap::GetRows ( ) const` `[inline]`

27.33.4.21 `const TransferSyntax& gdcm::Bitmap::GetTransferSyntax ( ) const` `[inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.33.4.22 `bool gdcm::Bitmap::IsEmpty ( ) const` `[inline]`

27.33.4.23 `bool gdcm::Bitmap::IsLossy ( ) const`

Return whether or not the image was compressed using a lossy compressor or not.

27.33.4.24 `bool gdcm::Bitmap::IsTransferSyntaxCompatible ( TransferSyntax const & ts ) const`

27.33.4.25 `void gdcm::Bitmap::Print ( std::ostream & ) const` `[virtual]`

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::Image](#), and [gdcm::Pixmap](#).

Examples:

[ExtractIconFromFile.cxx](#).

27.33.4.26 `void gdcm::Bitmap::SetColumns ( unsigned int col )` `[inline]`

27.33.4.27 `void gdcm::Bitmap::SetDataElement ( DataElement const & de )` `[inline]`

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), and [iU22tomultisc.cxx](#).

27.33.4.28 void `gdcm::Bitmap::SetDimension` ( unsigned int *idx*, unsigned int *dim* )

Examples:

[csa2img.cxx](#), [GenFakelImage.cxx](#), and [iU22tomultisc.cxx](#).

27.33.4.29 void `gdcm::Bitmap::SetDimensions` ( const unsigned int *dims*[3] )

Examples:

[CreateARGBImage.cxx](#), and [CreateCMYKImage.cxx](#).

27.33.4.30 void `gdcm::Bitmap::SetLossyFlag` ( bool *f* ) [inline]

Specifically set that the image was compressed using a lossy compression mechanism.

27.33.4.31 void `gdcm::Bitmap::SetLUT` ( `LookupTable` const & *lut* ) [inline]

Set/Get LUT.

27.33.4.32 void `gdcm::Bitmap::SetNeedByteSwap` ( bool *b* ) [inline]

27.33.4.33 void `gdcm::Bitmap::SetNumberOfDimensions` ( unsigned int *dim* )

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

27.33.4.34 void `gdcm::Bitmap::SetPhotometricInterpretation` ( `PhotometricInterpretation` const & *pi* )

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), and [iU22tomultisc.cxx](#).

27.33.4.35 void `gdcm::Bitmap::SetPixelFormat` ( `PixelFormat` const & *pf* ) [inline]

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

References `gdcm::PixelFormat::Validate()`.

27.33.4.36 void `gdcm::Bitmap::SetPlanarConfiguration` ( unsigned int *pc* )

Warning

you need to call `SetPixelFormat` first (before `SetPlanarConfiguration`) for consistency checking

27.33.4.37 void gdcm::Bitmap::SetRows ( unsigned int *rows* ) [inline]

27.33.4.38 void gdcm::Bitmap::SetTransferSyntax ( TransferSyntax const & *ts* ) [inline]

Transfer syntax.

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [MergeTwoFiles.cxx](#).

27.33.4.39 bool gdcm::Bitmap::TryJPEG2000Codec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.40 bool gdcm::Bitmap::TryJPEG2000Codec2 ( std::ostream & *os* ) const [protected]

27.33.4.41 bool gdcm::Bitmap::TryJPEGCodec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.42 bool gdcm::Bitmap::TryJPEGCodec2 ( std::ostream & *os* ) const [protected]

27.33.4.43 bool gdcm::Bitmap::TryJPEGLSCodec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.44 bool gdcm::Bitmap::TryKAKADUCoder ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.45 bool gdcm::Bitmap::TryPVRGCodec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.46 bool gdcm::Bitmap::TryRAWCodec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

27.33.4.47 bool gdcm::Bitmap::TryRLECodec ( char \* *buffer*, bool & *lossyflag* ) const [protected]

## 27.33.5 Friends And Related Function Documentation

27.33.5.1 friend class ImageChangeTransferSyntax [friend]

27.33.5.2 friend class PixmapReader [friend]

## 27.33.6 Member Data Documentation

27.33.6.1 std::vector<unsigned int> gdcm::Bitmap::Dimensions [protected]

27.33.6.2 bool gdcm::Bitmap::LossyFlag [protected]

27.33.6.3 LUTPtr gdcm::Bitmap::LUT [protected]

27.33.6.4 bool gdcm::Bitmap::NeedByteSwap [protected]

27.33.6.5 unsigned int gdcm::Bitmap::NumberOfDimensions [protected]

27.33.6.6 PixelFormat gdcm::Bitmap::PF [protected]

27.33.6.7 PhotometricInterpretation gdcm::Bitmap::PI [protected]

27.33.6.8 DataElement gdcm::Bitmap::PixelData [protected]

27.33.6.9 `unsigned int gdcm::Bitmap::PlanarConfiguration` `[protected]`

27.33.6.10 `TransferSyntax gdcm::Bitmap::TS` `[protected]`

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

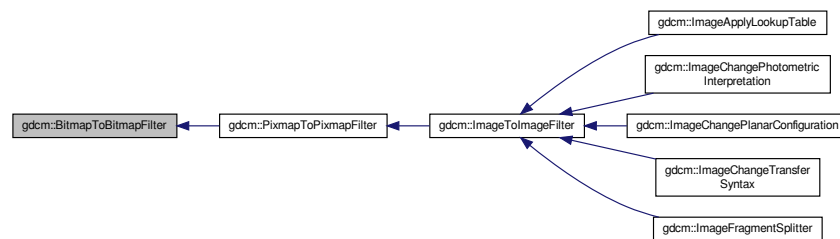
- [gdcmBitmap.h](#)

## 27.34 `gdcm::BitmapToBitmapFilter` Class Reference

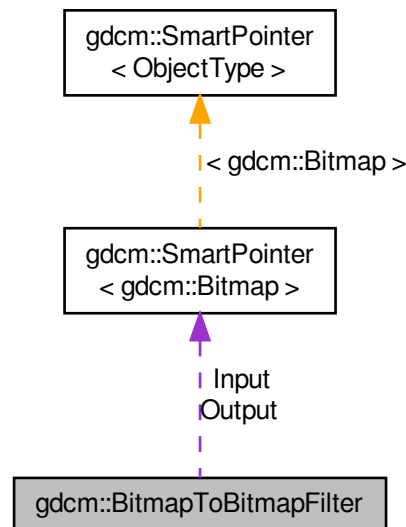
[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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](#)

#### 27.34.1 Detailed Description

[BitmapToBitmapFilter](#) class Super class for all filter taking an image and producing an output image.

#### 27.34.2 Constructor & Destructor Documentation

##### 27.34.2.1 gdcm::BitmapToBitmapFilter::BitmapToBitmapFilter ( )

27.34.2.2 `gdcm::BitmapToBitmapFilter::~~BitmapToBitmapFilter ( )` `[inline]`

### 27.34.3 Member Function Documentation

27.34.3.1 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutput ( ) const` `[inline]`

Get Output image.

27.34.3.2 `const Bitmap& gdcm::BitmapToBitmapFilter::GetOutputAsBitmap ( ) const`

27.34.3.3 `void gdcm::BitmapToBitmapFilter::SetInput ( const Bitmap & image )`

Set input image.

Examples:

[CompressImage.cxx](#).

### 27.34.4 Member Data Documentation

27.34.4.1 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Input` `[protected]`

27.34.4.2 `SmartPointer<Bitmap> gdcm::BitmapToBitmapFilter::Output` `[protected]`

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

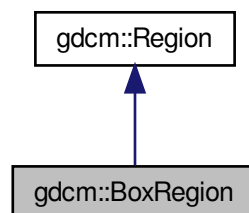
- [gdcmBitmapToBitmapFilter.h](#)

## 27.35 gdcm::BoxRegion Class Reference

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)

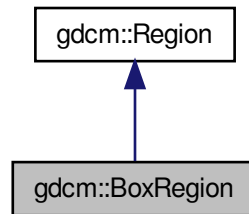
```
#include <gdcmBoxRegion.h>
```

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](#).*

### 27.35.1 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)

### 27.35.2 Constructor & Destructor Documentation

27.35.2.1 `gdcm::BoxRegion::BoxRegion ( )`

27.35.2.2 `gdcm::BoxRegion::~~BoxRegion ( )`

27.35.2.3 `gdcm::BoxRegion::BoxRegion ( const BoxRegion & )`

copy/cstor and al.

### 27.35.3 Member Function Documentation

27.35.3.1 `size_t gdcm::BoxRegion::Area ( ) const` [virtual]

compute the area

Implements [gdcm::Region](#).

27.35.3.2 `static BoxRegion gdcm::BoxRegion::BoundingBox ( BoxRegion const & b1, BoxRegion const & b2 )`  
[static]

Helper class to compute the bounding box of two [BoxRegion](#).

27.35.3.3 `Region* gdcm::BoxRegion::Clone ( ) const` [virtual]

Implements [gdcm::Region](#).

27.35.3.4 `BoxRegion gdcm::BoxRegion::ComputeBoundingBox ( )` [virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implements [gdcm::Region](#).

27.35.3.5 `bool gdcm::BoxRegion::Empty ( ) const` [virtual]

return whether this domain is empty:

Implements [gdcm::Region](#).

27.35.3.6 `unsigned int gdcm::BoxRegion::GetXMax ( ) const`

27.35.3.7 `unsigned int gdcm::BoxRegion::GetXMin ( ) const`

Get domain.

27.35.3.8 unsigned int gdcm::BoxRegion::GetYMax ( ) const

27.35.3.9 unsigned int gdcm::BoxRegion::GetYMin ( ) const

27.35.3.10 unsigned int gdcm::BoxRegion::GetZMax ( ) const

27.35.3.11 unsigned int gdcm::BoxRegion::GetZMin ( ) const

27.35.3.12 bool gdcm::BoxRegion::IsValid ( ) const [virtual]

return whether this is valid domain

Implements [gdcm::Region](#).

27.35.3.13 void gdcm::BoxRegion::operator= ( const BoxRegion & )

27.35.3.14 void gdcm::BoxRegion::Print ( std::ostream & os = std::cout ) const [virtual]

Print.

Reimplemented from [gdcm::Region](#).

27.35.3.15 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](#)

## 27.36 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](#) ()

### 27.36.1 Detailed Description

[ByteBuffer](#).

Detailed description here

## Note

looks like a `std::streambuf` or `std::filebuf` class with the get and peek pointer

## 27.36.2 Constructor & Destructor Documentation

27.36.2.1 `gdcmm::ByteBuffer::ByteBuffer ( )` `[inline]`

## 27.36.3 Member Function Documentation

27.36.3.1 `char* gdcmm::ByteBuffer::Get ( int len )` `[inline]`

27.36.3.2 `const char* gdcmm::ByteBuffer::GetStart ( ) const` `[inline]`

27.36.3.3 `void gdcmm::ByteBuffer::ShiftEnd ( int len )` `[inline]`

27.36.3.4 `void gdcmm::ByteBuffer::UpdatePosition ( )` `[inline]`

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

- [gdcmmByteBuffer.h](#)

## 27.37 gdcmm::ByteSwap< T > Class Template Reference

[ByteSwap.](#)

```
#include <gdcmmByteSwap.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](#) ()

### 27.37.1 Detailed Description

```
template<class T>class gdcmm::ByteSwap< T >
```

[ByteSwap.](#)

Perform machine dependent byte swapping (Little Endian, Big Endian, Bad Little Endian, Bad Big Endian). TODO: `bswap_32` / `bswap_64` ...

Examples:

[TestByteSwap.cxx.](#)

### 27.37.2 Member Function Documentation

27.37.2.1 `template<class T> static void gdcm::ByteSwap<T>::Swap ( T & p ) [static]`

27.37.2.2 `template<class T> static void gdcm::ByteSwap<T>::SwapFromSwapCodeIntoSystem ( T & p, SwapCode const & sc ) [static]`

Examples:

[TestByteSwap.cxx](#).

27.37.2.3 `template<class T> static void gdcm::ByteSwap<T>::SwapRange ( T * p, unsigned int num ) [static]`

27.37.2.4 `template<class T> static void gdcm::ByteSwap<T>::SwapRangeFromSwapCodeIntoSystem ( T * p, SwapCode const & sc, std::streamoff num ) [static]`

Examples:

[TestByteSwap.cxx](#).

27.37.2.5 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsBigEndian ( ) [static]`

Query the machine Endian-ness.

27.37.2.6 `template<class T> static bool gdcm::ByteSwap<T>::SystemIsLittleEndian ( ) [static]`

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

- [gdcmByteSwap.h](#)

## 27.38 gdcm::ByteSwapFilter Class Reference

[ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??

```
#include <gdcmByteSwapFilter.h>
```

### Public Member Functions

- [ByteSwapFilter](#) ([DataSet](#) &ds)
- [~ByteSwapFilter](#) ()
- bool [ByteSwap](#) ()
- void [SetByteSwapTag](#) (bool b)

### 27.38.1 Detailed Description

[ByteSwapFilter](#) In place byte-swapping of a dataset FIXME: FL status ??

## 27.38.2 Constructor & Destructor Documentation

27.38.2.1 `gdcm::ByteSwapFilter::ByteSwapFilter ( DataSet & ds )` `[inline]`

27.38.2.2 `gdcm::ByteSwapFilter::~~ByteSwapFilter ( )`

## 27.38.3 Member Function Documentation

27.38.3.1 `bool gdcm::ByteSwapFilter::ByteSwap ( )`

27.38.3.2 `void gdcm::ByteSwapFilter::SetByteSwapTag ( bool b )` `[inline]`

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

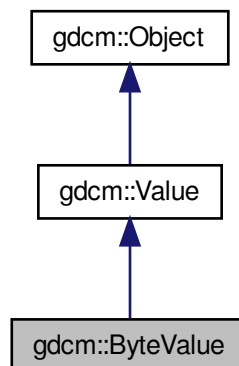
- [gdcmByteSwapFilter.h](#)

## 27.39 gdcm::ByteValue Class Reference

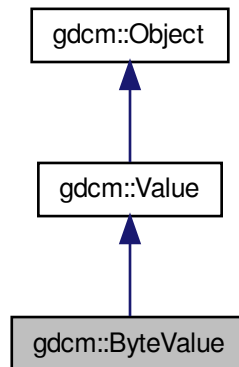
Class to represent binary value (array of bytes)

```
#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 [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)

### 27.39.1 Detailed Description

Class to represent binary value (array of bytes)

Note

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumplImageHeaderInfo.cxx](#), [DumpPhilipsECHO.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](#).

### 27.39.2 Constructor & Destructor Documentation

27.39.2.1 `gdcm::ByteValue::ByteValue ( const char * array = 0, VL const & vl = 0 ) [inline]`

References [gdcmDebugMacro](#).

27.39.2.2 `gdcm::ByteValue::ByteValue ( std::vector< char > & v ) [inline]`

Warning

casting to `uint32_t`

27.39.2.3 `gdcm::ByteValue::~ByteValue ( ) [inline]`

### 27.39.3 Member Function Documentation

27.39.3.1 `void gdcm::ByteValue::Clear ( ) [inline],[virtual]`

Implements [gdcm::Value](#).

27.39.3.2 `VL gdcm::ByteValue::ComputeLength ( ) const [inline]`

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



27.39.3.3 `void gdcm::ByteValue::Fill ( char c ) [inline]`

Examples:

[DuplicatePCDE.cxx](#).

27.39.3.4 `bool gdcm::ByteValue::GetBuffer ( char * buffer, unsigned long length ) const`

Examples:

[FixJAIBugJPEGLS.cxx](#).

27.39.3.5 `VL gdcm::ByteValue::GetLength ( ) const [inline], [virtual]`

Implements [gdcm::Value](#).

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.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\(\)](#).

27.39.3.6 `const char* gdcm::ByteValue::GetPointer ( ) const [inline]`

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.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\(\)](#).

27.39.3.7 `bool gdcm::ByteValue::IsEmpty ( ) const [inline]`

27.39.3.8 `bool gdcm::ByteValue::IsPrintable ( VL length ) const [inline]`

Checks whether a 'ByteValue' is printable or not (in order to avoid corrupting the terminal of invocation when printing) I don't think this function is working since it does not handle UNICODE or character set...

27.39.3.9 `gdcmm::ByteValue::operator const std::vector< char > & ( ) const` `[inline]`

27.39.3.10 `ByteValue& gdcmm::ByteValue::operator= ( const ByteValue & val )` `[inline]`

27.39.3.11 `bool gdcmm::ByteValue::operator== ( const ByteValue & val ) const` `[inline]`

27.39.3.12 `bool gdcmm::ByteValue::operator== ( const Value & val ) const` `[inline],[virtual]`

Implements [gdcmm::Value](#).

27.39.3.13 `void gdcmm::ByteValue::Print ( std::ostream & os ) const` `[inline],[protected],[virtual]`

Reimplemented from [gdcmm::Object](#).

27.39.3.14 `void gdcmm::ByteValue::PrintASCII ( std::ostream & os, VL maxlength ) const`

27.39.3.15 `void gdcmm::ByteValue::PrintASCIIXML ( std::ostream & os ) const`

27.39.3.16 `void gdcmm::ByteValue::PrintGroupLength ( std::ostream & os )` `[inline]`

27.39.3.17 `void gdcmm::ByteValue::PrintHex ( std::ostream & os, VL maxlength ) const`

27.39.3.18 `void gdcmm::ByteValue::PrintHexXML ( std::ostream & os ) const`

27.39.3.19 `void gdcmm::ByteValue::PrintPXML ( std::ostream & os ) const`

To Print Values in Native DICOM format

27.39.3.20 `template<typename TSwap, typename TType > std::istream& gdcmm::ByteValue::Read ( std::istream & is, bool readvalues = true )` `[inline]`

27.39.3.21 `template<typename TSwap > std::istream& gdcmm::ByteValue::Read ( std::istream & is )` `[inline]`

27.39.3.22 `void gdcmm::ByteValue::SetLength ( VL vl )` `[inline],[virtual]`

Implements [gdcmm::Value](#).

References `gdcmm::DebugMacro`, `gdcmm::VL::IsOdd()`, and `gdcmm::VL::IsUndefined()`.

27.39.3.23 `void gdcmm::ByteValue::SetLengthOnly ( VL vl )` `[inline],[protected],[virtual]`

Reimplemented from [gdcmm::Value](#).

27.39.3.24 `template<typename TSwap, typename TType > std::ostream const& gdcmm::ByteValue::Write ( std::ostream & os ) const` `[inline]`

Referenced by `gdcmm::Fragment::Write()`.

27.39.3.25 `template<typename TSwap > std::ostream const& gdcm::ByteValue::Write ( std::ostream & os ) const` `[inline]`

27.39.3.26 `bool gdcm::ByteValue::WriteBuffer ( std::ostream & os ) const` `[inline]`

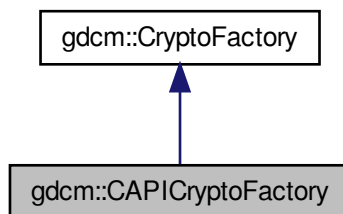
The documentation for this class was generated from the following file:

- [gdcmByteValue.h](#)

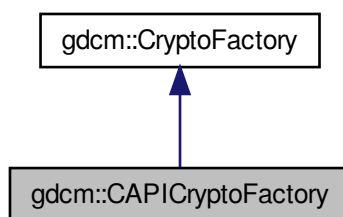
## 27.40 gdcm::CAPICryptoFactory Class Reference

```
#include <gdcmCAPICryptoFactory.h>
```

Inheritance diagram for gdcm::CAPICryptoFactory:



Collaboration diagram for gdcm::CAPICryptoFactory:



### Public Member Functions

- [CAPICryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Additional Inherited Members

### 27.40.1 Constructor & Destructor Documentation

27.40.1.1 `gdcm::CAPICryptoFactory::CAPICryptoFactory ( CryptoLib id )`

### 27.40.2 Member Function Documentation

27.40.2.1 `CryptographicMessageSyntax* gdcm::CAPICryptoFactory::CreateCMSProvider ( )` [virtual]

Implements [gdcm::CryptoFactory](#).

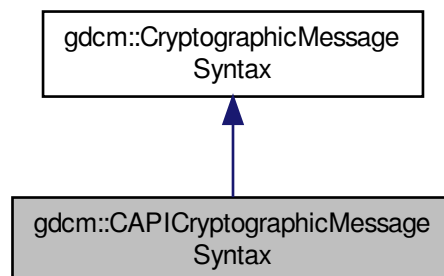
The documentation for this class was generated from the following file:

- [gdcmCAPICryptoFactory.h](#)

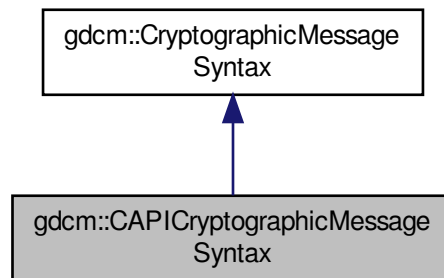
## 27.41 `gdcm::CAPICryptographicMessageSyntax` Class Reference

```
#include <gdcmCAPICryptographicMessageSyntax.h>
```

Inheritance diagram for `gdcm::CAPICryptographicMessageSyntax`:



Collaboration diagram for gdcmm::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

### 27.41.1 Constructor & Destructor Documentation

27.41.1.1 `gdcmm::CAPICryptographicMessageSyntax::CAPICryptographicMessageSyntax ( )`

27.41.1.2 `gdcmm::CAPICryptographicMessageSyntax::~~CAPICryptographicMessageSyntax ( )`

### 27.41.2 Member Function Documentation

27.41.2.1 `bool gdcmm::CAPICryptographicMessageSyntax::Decrypt ( char * output, size_t & outlen, const char * array, size_t len ) const [virtual]`

decrypt content from a CMS envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

27.41.2.2 `bool gdcM::CAPICryptographicMessageSyntax::Encrypt ( char * output, size_t & outlen, const char * array, size_t len ) const [virtual]`

create a CMS envelopedData structure

Implements [gdcM::CryptographicMessageSyntax](#).

27.41.2.3 `CipherTypes gdcM::CAPICryptographicMessageSyntax::GetCipherType ( ) const [virtual]`

Implements [gdcM::CryptographicMessageSyntax](#).

27.41.2.4 `bool gdcM::CAPICryptographicMessageSyntax::GetInitialized ( ) const [inline]`

27.41.2.5 `bool gdcM::CAPICryptographicMessageSyntax::ParseCertificateFile ( const char * filename ) [virtual]`

Implements [gdcM::CryptographicMessageSyntax](#).

27.41.2.6 `bool gdcM::CAPICryptographicMessageSyntax::ParseKeyFile ( const char * filename ) [virtual]`

Implements [gdcM::CryptographicMessageSyntax](#).

27.41.2.7 `void gdcM::CAPICryptographicMessageSyntax::SetCipherType ( CipherTypes type )`

27.41.2.8 `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:

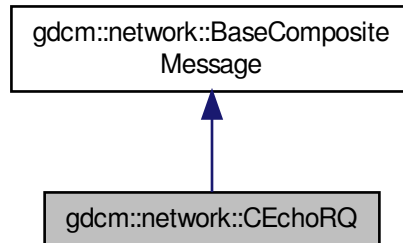
- [gdcMCAPICryptographicMessageSyntax.h](#)

## 27.42 gdcM::network::CEchoRQ Class Reference

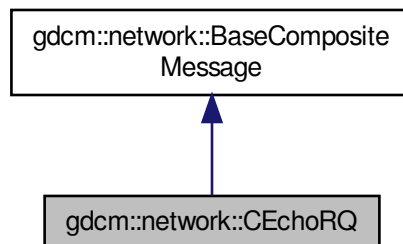
[CEchoRQ](#) this file defines the messages for the cecho action.

```
#include <gdcMCEchoMessages.h>
```

Inheritance diagram for gdcmm::network::CEchoRQ:



Collaboration diagram for gdcmm::network::CEchoRQ:



## Public Member Functions

- `std::vector`  
`< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const BaseRootQuery *inRootQuery)`

## Public Attributes

- `UIComp AffectedSOPClassUID`
- `uint16_t MessageID`

### 27.42.1 Detailed Description

`CEchoRQ` this file defines the messages for the cecho action.

## 27.42.2 Member Function Documentation

27.42.2.1 `std::vector<PresentationDataValue> gdcn::network::CEchoRQ::ConstructPDV ( const ULConnection & inConnection, const BaseRootQuery * inRootQuery ) [virtual]`

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

## 27.42.3 Member Data Documentation

27.42.3.1 `UIComp gdcn::network::CEchoRQ::AffectedSOPClassUID`

27.42.3.2 `uint16_t gdcn::network::CEchoRQ::MessageID`

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

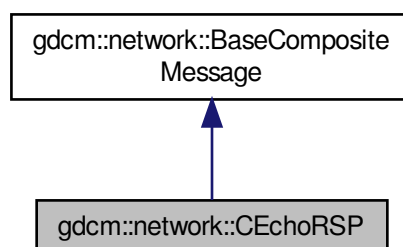
- [gdcnCEchoMessages.h](#)
- [gdcnDIMSE.h](#)

## 27.43 gdcn::network::CEchoRSP Class Reference

[CEchoRSP](#) this file defines the messages for the cecho action.

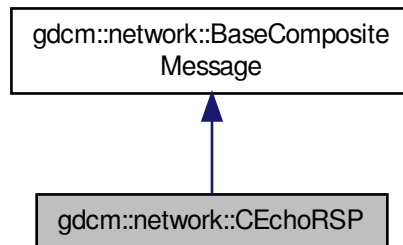
```
#include <gdcnCEchoMessages.h>
```

Inheritance diagram for `gdcn::network::CEchoRSP`:





Collaboration diagram for gdcmm::network::CEchoRSP:



### Public Member Functions

- `std::vector`  
`< PresentationDataValue > ConstructPDVByDataSet (const DataSet *inDataSet)`

#### 27.43.1 Detailed Description

`CEchoRSP` this file defines the messages for the cecho action.

#### 27.43.2 Member Function Documentation

27.43.2.1 `std::vector<PresentationDataValue> gdcmm::network::CEchoRSP::ConstructPDVByDataSet ( const DataSet *inDataSet )`

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

- `gdcmmCEchoMessages.h`

## 27.44 gdcmm::network::CFind Class Reference

```
#include <gdcmmDIMSE.h>
```

#### 27.44.1 Detailed Description

PS 3.4 - 2009 [Table B.2-1 C-STORE STATUS](#)

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

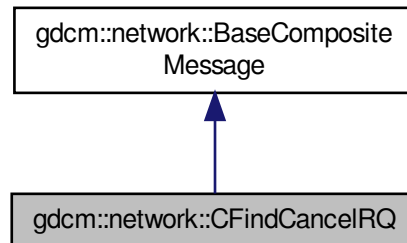
- `gdcmmDIMSE.h`

## 27.45 gdcm::network::CFindCancelRQ Class Reference

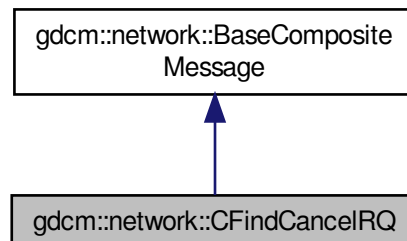
[CFindCancelRQ](#) this file defines the messages for the cfind action.

```
#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)

### 27.45.1 Detailed Description

[CFindCancelRQ](#) this file defines the messages for the cfind action.

## 27.45.2 Member Function Documentation

27.45.2.1 `std::vector<PresentationDataValue> gdcm::network::CFindCancelRQ::ConstructPDVByDataSet ( const DataSet * inDataSet )`

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

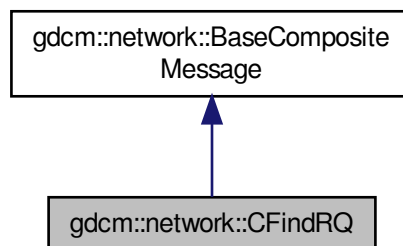
- [gdcmCFindMessages.h](#)

## 27.46 gdcm::network::CFindRQ Class Reference

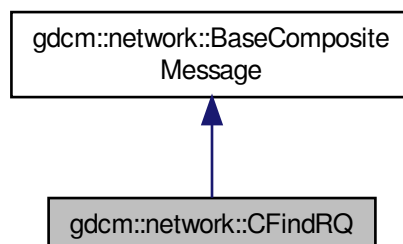
[CFindRQ](#) this file defines the messages for the cfind action.

```
#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)`

### 27.46.1 Detailed Description

[CFindRQ](#) this file defines the messages for the cfind action.

### 27.46.2 Member Function Documentation

27.46.2.1 `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:

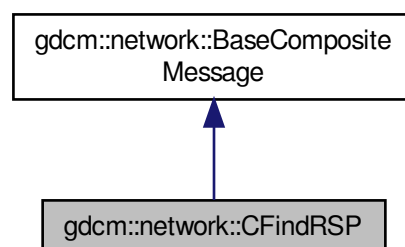
- [gdcmCFindMessages.h](#)

## 27.47 gdcm::network::CFindRSP Class Reference

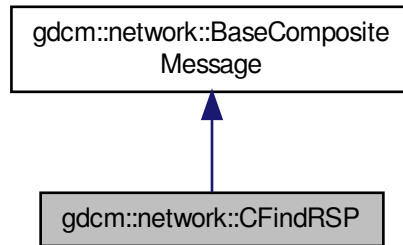
[CFindRSP](#) this file defines the messages for the cfind action.

```
#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)`

### 27.47.1 Detailed Description

`CFindRSP` this file defines the messages for the cfind action.

### 27.47.2 Member Function Documentation

**27.47.2.1** `std::vector<PresentationDataValue> gdcm::network::CFindRSP::ConstructPDVByDataSet ( const DataSet * inDataSet )`

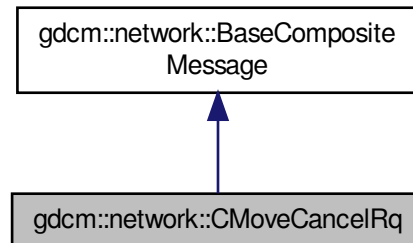
The documentation for this class was generated from the following file:

- `gdcmCFindMessages.h`

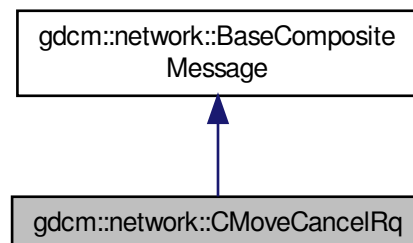
## 27.48 gdcm::network::CMoveCancelRq Class Reference

```
#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)`

### 27.48.1 Member Function Documentation

27.48.1.1 `std::vector<PresentationDataValue> gdcm::network::CMoveCancelRq::ConstructPDVByDataSet ( const DataSet *inDataSet )`

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

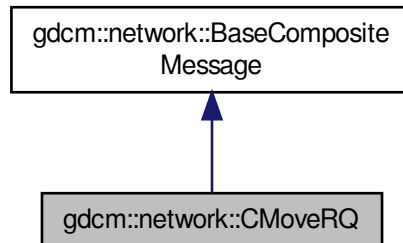
- [gdcmCMoveMessages.h](#)

## 27.49 gdcm::network::CMoveRQ Class Reference

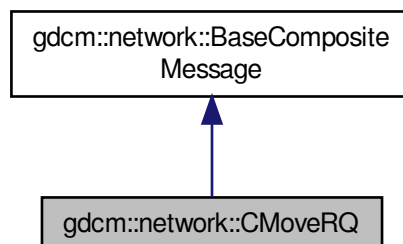
[CMoveRQ](#) this file defines the messages for the cmove action.

```
#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)

### 27.49.1 Detailed Description

[CMoveRQ](#) this file defines the messages for the cmove action.

## 27.49.2 Member Function Documentation

27.49.2.1 `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:

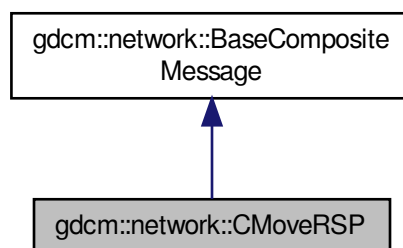
- [gdcmCMoveMessages.h](#)

## 27.50 gdcm::network::CMoveRSP Class Reference

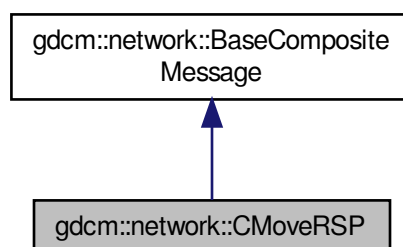
[CMoveRSP](#) this file defines the messages for the cmove action.

```
#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)`

### 27.50.1 Detailed Description

[CMoveRSP](#) this file defines the messages for the cmove action.

### 27.50.2 Member Function Documentation

27.50.2.1 `std::vector<PresentationDataValue> gdcmm::network::CMoveRSP::ConstructPDVByDataSet ( const DataSet * inDataSet )`

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

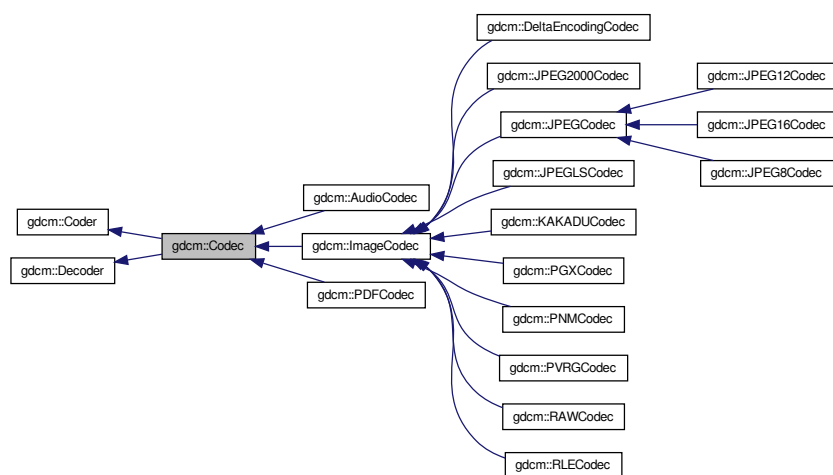
- [gdcmmCMoveMessages.h](#)

## 27.51 gdcmm::Codec Class Reference

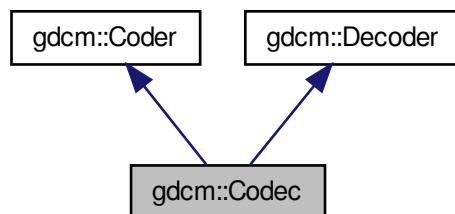
[Codec](#) class.

```
#include <gdcmmCodec.h>
```

Inheritance diagram for `gdcmm::Codec`:



Collaboration diagram for `gdcm::Codec`:



## Additional Inherited Members

### 27.51.1 Detailed Description

[Codec](#) class.

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

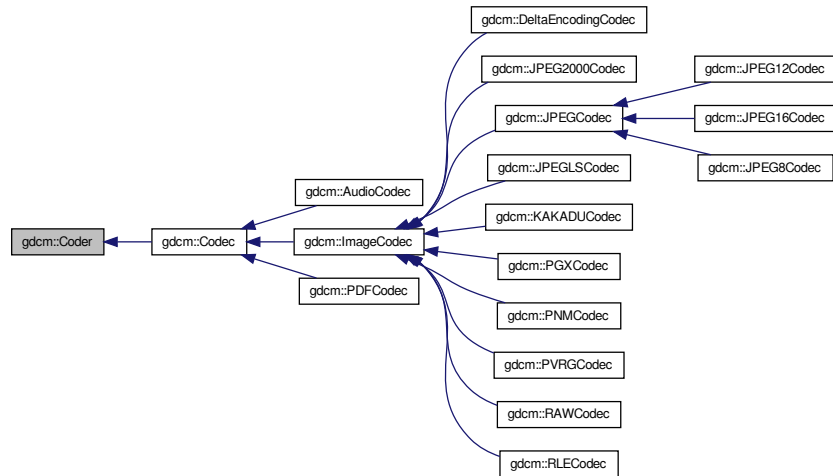
- [gdcmCodec.h](#)

## 27.52 `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)

## 27.52.1 Detailed Description

[Coder](#).

## 27.52.2 Constructor & Destructor Documentation

27.52.2.1 virtual gdcm::Coder::~~Coder ( ) [inline], [virtual]

## 27.52.3 Member Function Documentation

27.52.3.1 virtual bool gdcm::Coder::CanCode ( [TransferSyntax](#) const & ) const [pure virtual]

Return whether this coder support this transfer syntax (can code it)

Implemented in [gdcm::JPEGLSCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEGLSCodec](#), [gdcm::ImageCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), [gdcm::RAWCodec](#), [gdcm::AudioCodec](#), and [gdcm::PDFCodec](#).

27.52.3.2 `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](#).

27.52.3.3 `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](#)

## 27.53 gdcm::CodeString Class Reference

[CodeString](#) This is an implementation of DICOM VR: CS The ctor will properly Trim so that operator== is correct.

```
#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)

## 27.53.1 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.

## 27.53.2 Member Typedef Documentation

27.53.2.1 typedef InternalClass::const\_iterator gdcm::CodeString::const\_iterator

27.53.2.2 typedef InternalClass::const\_reference gdcm::CodeString::const\_reference

27.53.2.3 typedef InternalClass::const\_reverse\_iterator gdcm::CodeString::const\_reverse\_iterator

27.53.2.4 typedef InternalClass::difference\_type gdcm::CodeString::difference\_type

27.53.2.5 typedef InternalClass::iterator gdcm::CodeString::iterator

27.53.2.6 typedef InternalClass::pointer gdcm::CodeString::pointer

27.53.2.7 typedef InternalClass::reference gdcm::CodeString::reference

27.53.2.8 typedef InternalClass::reverse\_iterator gdcm::CodeString::reverse\_iterator

27.53.2.9 `typedef InternalClass::size_type gdcm::CodeString::size_type`

27.53.2.10 `typedef InternalClass::value_type gdcm::CodeString::value_type`

### 27.53.3 Constructor & Destructor Documentation

27.53.3.1 `gdcm::CodeString::CodeString ( ) [inline]`

[CodeString](#) constructors.

27.53.3.2 `gdcm::CodeString::CodeString ( const value_type * s ) [inline]`

27.53.3.3 `gdcm::CodeString::CodeString ( const value_type * s, size_type n ) [inline]`

27.53.3.4 `gdcm::CodeString::CodeString ( const InternalClass & s, size_type pos = 0, size_type n = InternalClass::npos ) [inline]`

### 27.53.4 Member Function Documentation

27.53.4.1 `std::string gdcm::CodeString::GetAsString ( ) const [inline]`

Return the full code string as std::string.

27.53.4.2 `bool gdcm::CodeString::IsValid ( ) const`

Check if [CodeString](#) obj is correct..

27.53.4.3 `size_type gdcm::CodeString::Size ( ) const [inline]`

Return the size of the string.

27.53.4.4 `std::string gdcm::CodeString::TrimInternal ( ) const [inline],[protected]`

### 27.53.5 Friends And Related Function Documentation

27.53.5.1 `bool operator!= ( const CodeString & ref, const CodeString & cs ) [friend]`

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

27.53.5.3 `bool operator== ( const CodeString & ref, const CodeString & cs ) [friend]`

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

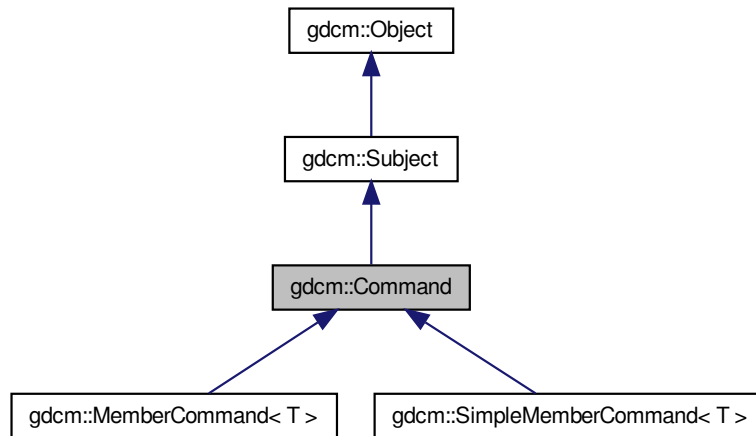
- [gdcmCodeString.h](#)

## 27.54 gdcm::Command Class Reference

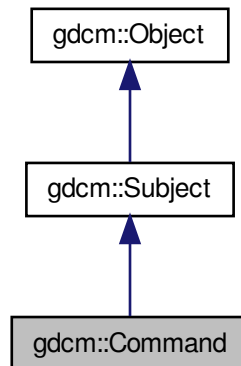
[Command](#) superclass for callback/observer methods.

```
#include <gdcMCommand.h>
```

Inheritance diagram for gdcM::Command:



Collaboration diagram for gdcM::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\(\)](#)

### 27.54.1 Detailed Description

[Command](#) superclass for callback/observer methods.

See also

[Subject](#)

### 27.54.2 Constructor & Destructor Documentation

27.54.2.1 `gdcM::Command::Command( )` [protected]

27.54.2.2 `gdcM::Command::~~Command( )` [protected]

### 27.54.3 Member Function Documentation

27.54.3.1 `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 >](#).

27.54.3.2 `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](#)

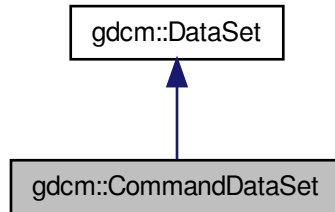
## 27.55 gdcM::CommandDataSet Class Reference

Class to represent a [Command DataSet](#).

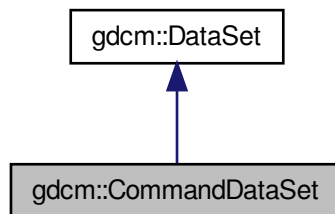
```
#include <gdcMCommandDataSet.h>
```



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

### 27.55.1 Detailed Description

Class to represent a [Command DataSet](#).

See also

[DataSet](#)

### 27.55.2 Constructor & Destructor Documentation

27.55.2.1 `gdcm::CommandDataSet::CommandDataSet ( )` `[inline]`

27.55.2.2 `gdcm::CommandDataSet::~~CommandDataSet ( )` `[inline]`

### 27.55.3 Member Function Documentation

27.55.3.1 `void gdcm::CommandDataSet::Insert ( const DataElement & de )` `[inline]`

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

27.55.3.2 `std::istream& gdcm::CommandDataSet::Read ( std::istream & is )`

Read.

27.55.3.3 `void gdcm::CommandDataSet::Replace ( const DataElement & de )` `[inline]`

References `gdcm::DataElement::GetTag()`.

27.55.3.4 `std::ostream& gdcm::CommandDataSet::Write ( std::ostream & os ) const`

Write.

### 27.55.4 Friends And Related Function Documentation

27.55.4.1 `std::ostream& operator<< ( std::ostream & _os, const CommandDataSet & _val )` `[friend]`

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

- [gdcmCommandDataSet.h](#)

## 27.56 gdcm::network::CompositeMessageFactory Class Reference

[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).

```
#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)
- static std::vector  
< [PresentationDataValue](#) > [ConstructCStoreRSP](#) (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)

#### 27.56.1 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).

#### 27.56.2 Member Function Documentation

- 27.56.2.1 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCEchoRQ ( const [ULConnection](#) & *inConnection* ) [static]
- 27.56.2.2 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCFindRQ ( const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) \* *inRootQuery* ) [static]
- 27.56.2.3 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCMoveRQ ( const [ULConnection](#) & *inConnection*, const [BaseRootQuery](#) \* *inRootQuery* ) [static]
- 27.56.2.4 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCStoreRQ ( const [ULConnection](#) & *inConnection*, const [File](#) & *file* ) [static]
- 27.56.2.5 static std::vector<[PresentationDataValue](#)> gdcm::network::CompositeMessageFactory::ConstructCStoreRSP ( const [DataSet](#) \* *inDataSet*, const [BasePDU](#) \* *inPC* ) [static]

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

- [gdcmCompositeMessageFactory.h](#)

## 27.57 gdcm::CompositeNetworkFunctions Class Reference

**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:

```
#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, bool inMove=false)
- static [BaseRootQuery](#) \* [ConstructQuery](#) ([ERootType](#) inRootType, [EQueryLevel](#) inQueryLevel, const [KeyValuePairArray↵Type](#) &keys, bool inMove=false)
- static bool [CStore](#) (const char \*remote, uint16\_t portno, const [Directory::FileNamesType](#) &filenames, const char \*aetitle=NULL, const char \*call=NULL)

### 27.57.1 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)

### 27.57.2 Member Typedef Documentation

#### 27.57.2.1 typedef std::vector< [KeyValuePairType](#) > [gdcm::CompositeNetworkFunctions::KeyValuePairArrayType](#)

27.57.2.2 `typedef std::pair<Tag, std::string> gdcmm::CompositeNetworkFunctions::KeyValuePairType`

### 27.57.3 Member Function Documentation

27.57.3.1 `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.

27.57.3.2 `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.

27.57.3.3 `static bool gdcmm::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.

**27.57.3.4** static **BaseRootQuery\*** **gdcM::CompositeNetworkFunctions::ConstructQuery** ( **ERootType** *inRootType*, **EQueryLevel** *inQueryLevel*, const **DataSet** & *queryds*, bool *inMove* = false ) [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)
------------------	--

**27.57.3.5** static **BaseRootQuery\*** **gdcM::CompositeNetworkFunctions::ConstructQuery** ( **ERootType** *inRootType*, **EQueryLevel** *inQueryLevel*, const **KeyValuePairArrayType** & *keys*, bool *inMove* = false ) [static]

## Deprecated

**27.57.3.6** 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](#)

## 27.58 gdcm::ConstCharWrapper Class Reference

Do not use me.

```
#include <gdcmConstCharWrapper.h>
```

### Public Member Functions

- [ConstCharWrapper](#) (const char \*i=0)
- [operator const char \\* \(\)](#) const

### 27.58.1 Detailed Description

Do not use me.

### 27.58.2 Constructor & Destructor Documentation

27.58.2.1 `gdcm::ConstCharWrapper::ConstCharWrapper ( const char * i = 0 ) [inline]`

### 27.58.3 Member Function Documentation

27.58.3.1 `gdcm::ConstCharWrapper::operator const char * ( ) const [inline]`

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

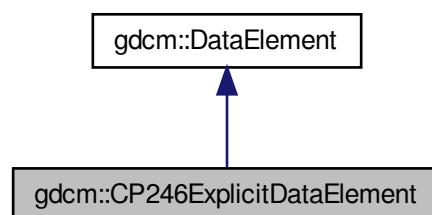
- [gdcmConstCharWrapper.h](#)

## 27.59 gdcm::CP246ExplicitDataElement Class Reference

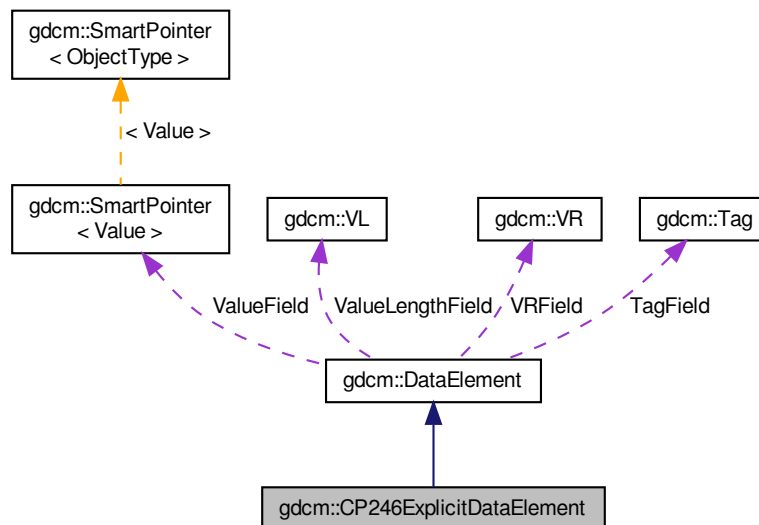
Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).

```
#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

### 27.59.1 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

### 27.59.2 Member Function Documentation

#### 27.59.2.1 VL `gdcm::CP246ExplicitDataElement::GetLength` ( ) const



27.59.2.2 `template<typename TSwap> std::istream& gdcm::CP246ExplicitDataElement::Read ( std::istream & is )`

27.59.2.3 `template<typename TSwap> std::istream& gdcm::CP246ExplicitDataElement::ReadPreValue ( std::istream & is )`

27.59.2.4 `template<typename TSwap> std::istream& gdcm::CP246ExplicitDataElement::ReadValue ( std::istream & is, bool readvalues = true )`

27.59.2.5 `template<typename TSwap> std::istream& gdcm::CP246ExplicitDataElement::ReadWithLength ( std::istream & is, VL & length )`

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

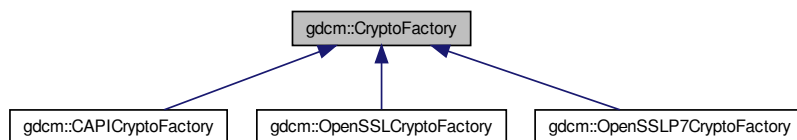
- [gdcmCP246ExplicitDataElement.h](#)

## 27.60 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 \(\)](#)

### 27.60.1 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)
- OPENSSL7 (portable) By default the factory will try: CAPI if on windows OPENSSL if possible OPENSSL7 when older OpenSSL is used.

Examples:

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

### 27.60.2 Member Enumeration Documentation

#### 27.60.2.1 enum gdcm::CryptoFactory::CryptoLib

Enumerator

***DEFAULT***  
***OPENSSL***  
***CAPI***  
***OPENSSL7***

### 27.60.3 Constructor & Destructor Documentation

27.60.3.1 `gdcm::CryptoFactory::CryptoFactory ( CryptoLib id )` `[inline]`, `[protected]`

27.60.3.2 `gdcm::CryptoFactory::CryptoFactory ( )` `[inline]`, `[protected]`

27.60.3.3 `gdcm::CryptoFactory::~~CryptoFactory ( )` `[inline]`, `[protected]`

### 27.60.4 Member Function Documentation

27.60.4.1 `virtual CryptographicMessageSyntax* gdcm::CryptoFactory::CreateCMSProvider ( )` `[pure virtual]`

Implemented in [gdcm::OpenSSLCryptoFactory](#), [gdcm::OpenSSL7CryptoFactory](#), and [gdcm::CAPICryptoFactory](#).

27.60.4.2 `static CryptographicMessageSyntax* gdcm::CryptoFactory::GetFactoryInstance ( CryptoLib id = DEFAULT )` `[static]`

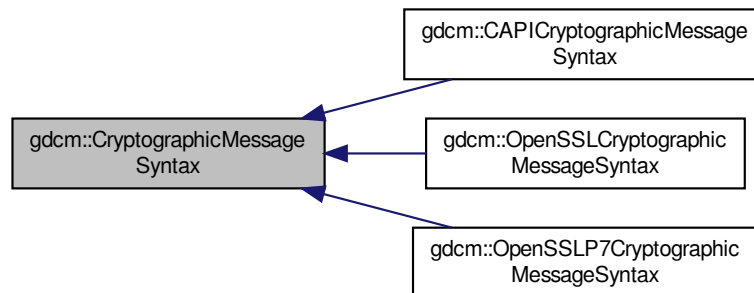
The documentation for this class was generated from the following file:

- [gdcmCryptoFactory.h](#)

## 27.61 gdcM::CryptographicMessageSyntax Class Reference

```
#include <gdcMCryptographicMessageSyntax.h>
```

Inheritance diagram for gdcM::CryptographicMessageSyntax:



### Public Types

- enum [CipherTypes](#) {  
[DES3\\_CIPHER](#),  
[AES128\\_CIPHER](#),  
[AES192\\_CIPHER](#),  
[AES256\\_CIPHER](#) }

### Public Member Functions

- [CryptographicMessageSyntax](#) ()
- 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

#### 27.61.1 Detailed Description

Examples:

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

## 27.61.2 Member Enumeration Documentation

### 27.61.2.1 enum `gdcmm::CryptographicMessageSyntax::CipherTypes`

Enumerator

***DES3\_CIPHER***  
***AES128\_CIPHER***  
***AES192\_CIPHER***  
***AES256\_CIPHER***

## 27.61.3 Constructor & Destructor Documentation

27.61.3.1 `gdcmm::CryptographicMessageSyntax::CryptographicMessageSyntax ( )` `[inline]`

27.61.3.2 `virtual gdcmm::CryptographicMessageSyntax::~~CryptographicMessageSyntax ( )` `[inline]`, `[virtual]`

## 27.61.4 Member Function Documentation

27.61.4.1 `virtual bool gdcmm::CryptographicMessageSyntax::Decrypt ( char * output, size_t & outlen, const char * array, size_t len ) const` `[pure virtual]`

decrypt content from a CMS envelopedData structure

Implemented in [gdcmm::OpenSSL7CryptographicMessageSyntax](#), [gdcmm::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

27.61.4.2 `virtual bool gdcmm::CryptographicMessageSyntax::Encrypt ( char * output, size_t & outlen, const char * array, size_t len ) const` `[pure virtual]`

create a CMS envelopedData structure

Implemented in [gdcmm::OpenSSL7CryptographicMessageSyntax](#), [gdcmm::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

27.61.4.3 `virtual CipherTypes gdcmm::CryptographicMessageSyntax::GetCipherType ( ) const` `[pure virtual]`

Implemented in [gdcmm::OpenSSL7CryptographicMessageSyntax](#), [gdcmm::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

27.61.4.4 `virtual bool gdcmm::CryptographicMessageSyntax::ParseCertificateFile ( const char * filename )` `[pure virtual]`

Implemented in [gdcmm::OpenSSL7CryptographicMessageSyntax](#), [gdcmm::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

27.61.4.5 `virtual bool gdcmm::CryptographicMessageSyntax::ParseKeyFile ( const char * filename )` `[pure virtual]`

Implemented in [gdcmm::OpenSSL7CryptographicMessageSyntax](#), [gdcmm::CAPICryptographicMessageSyntax](#), and [gdcmm::OpenSSLCryptographicMessageSyntax](#).

27.61.4.6 `virtual void gdcM::CryptographicMessageSyntax::SetCipherType ( CipherTypes type ) [pure virtual]`

Implemented in [gdcM::OpenSSLP7CryptographicMessageSyntax](#), and [gdcM::OpenSSLCryptographicMessageSyntax](#).

27.61.4.7 `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:

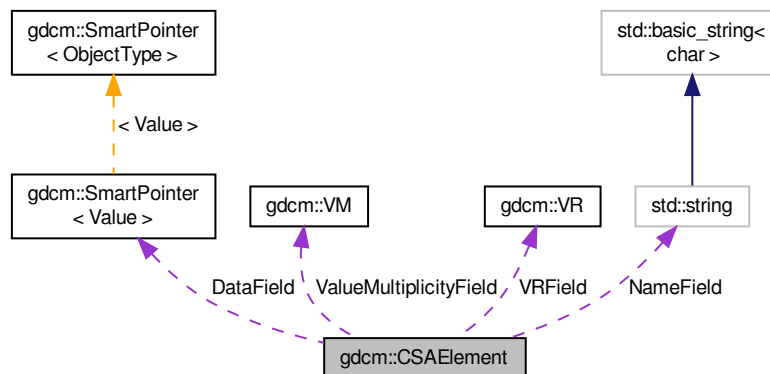
- [gdcMCryptographicMessageSyntax.h](#)

## 27.62 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)

### 27.62.1 Detailed Description

Class to represent a CSA [Element](#).

See also

[CSAHeader](#)

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

### 27.62.2 Member Typedef Documentation

27.62.2.1 `typedef SmartPointer<Value> gdcm::CSAElement::DataPtr` `[protected]`

### 27.62.3 Constructor & Destructor Documentation

27.62.3.1 `gdcm::CSAElement::CSAElement ( unsigned int kf = 0 )` `[inline]`

27.62.3.2 `gdcm::CSAElement::CSAElement ( const CSAElement &_val )` `[inline]`

### 27.62.4 Member Function Documentation

27.62.4.1 `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:

[MrProtocol.cxx](#).

27.62.4.2 `unsigned int gdcm::CSAElement::GetKey ( ) const` `[inline]`

Set/Get Key.

Referenced by operator<().

27.62.4.3 `const char* gdcm::CSAElement::GetName ( ) const` `[inline]`

Set/Get Name.

27.62.4.4 `unsigned int gdcm::CSAElement::GetNoOfItems ( ) const` `[inline]`

Set/Get NoOfItems.

27.62.4.5 `unsigned int gdcmm::CSAElement::GetSyngoDT ( ) const [inline]`

Set/Get SyngoDT.

27.62.4.6 `Value const& gdcmm::CSAElement::GetValue ( ) const [inline]`

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

Examples:

[csa2img.cxx](#).

27.62.4.7 `Value& gdcmm::CSAElement::GetValue ( ) [inline]`

27.62.4.8 `const VM& gdcmm::CSAElement::GetVM ( ) const [inline]`

Set/Get [VM](#).

27.62.4.9 `VR const& gdcmm::CSAElement::GetVR ( ) const [inline]`

Set/Get [VR](#).

27.62.4.10 `bool gdcmm::CSAElement::IsEmpty ( ) const [inline]`

Check if CSA [Element](#) is empty.

Examples:

[csa2img.cxx](#).

27.62.4.11 `bool gdcmm::CSAElement::operator< ( const CSAElement & de ) const [inline]`

References `GetKey()`.

27.62.4.12 `CSAElement& gdcmm::CSAElement::operator= ( const CSAElement & de ) [inline]`

References `DataField`, `KeyField`, `NameField`, `NoOfItemsField`, `SyngoDTField`, `ValueMultiplicityField`, and `VRField`.

27.62.4.13 `bool gdcmm::CSAElement::operator== ( const CSAElement & de ) const [inline]`

References `KeyField`, `NameField`, `SyngoDTField`, `ValueMultiplicityField`, and `VRField`.

27.62.4.14 `void gdcmm::CSAElement::SetByteValue ( const char * array, VL length ) [inline]`

Set.



- 27.62.4.15 void gdcm::CSAElement::SetKey ( unsigned int *key* ) [inline]
- 27.62.4.16 void gdcm::CSAElement::SetName ( const char \* *name* ) [inline]
- 27.62.4.17 void gdcm::CSAElement::SetNoOfItems ( unsigned int *items* ) [inline]
- 27.62.4.18 void gdcm::CSAElement::SetSyngoDT ( unsigned int *syngodt* ) [inline]
- 27.62.4.19 void gdcm::CSAElement::SetValue ( Value const & *v* ) [inline]
- 27.62.4.20 void gdcm::CSAElement::SetVM ( const VM & *vm* ) [inline]
- 27.62.4.21 void gdcm::CSAElement::SetVR ( VR const & *vr* ) [inline]

## 27.62.5 Friends And Related Function Documentation

- 27.62.5.1 std::ostream& operator<< ( std::ostream & *os*, const CSAElement & *val* ) [friend]

## 27.62.6 Member Data Documentation

- 27.62.6.1 DataPtr gdcm::CSAElement::DataField [protected]

Referenced by gdcm::operator<<(), and operator=().

- 27.62.6.2 unsigned int gdcm::CSAElement::KeyField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

- 27.62.6.3 std::string gdcm::CSAElement::NameField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

- 27.62.6.4 unsigned int gdcm::CSAElement::NoOfItemsField [protected]

Referenced by gdcm::operator<<(), and operator=().

- 27.62.6.5 unsigned int gdcm::CSAElement::SyngoDTField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

- 27.62.6.6 VM gdcm::CSAElement::ValueMultiplicityField [protected]

Referenced by gdcm::operator<<(), operator=(), and operator==().

- 27.62.6.7 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](#)

## 27.63 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 [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)`

### 27.63.1 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](#), and [MrProtocol.cxx](#).

### 27.63.2 Member Enumeration Documentation

#### 27.63.2.1 enum gdcm::CSAHeader::CSAHeaderType

Divers format of [CSAHeader](#) as found 'in the wild'.

#### Enumerator

**UNKNOWN**

**SV10**

**NOMAGIC**

**DATASET\_FORMAT**

**INTERFILE**

**ZEROED\_OUT**

### 27.63.3 Constructor & Destructor Documentation

27.63.3.1 `gdcM::CSAHeader::CSAHeader ( )` `[inline]`

27.63.3.2 `gdcM::CSAHeader::~~CSAHeader ( )` `[inline]`

### 27.63.4 Member Function Documentation

27.63.4.1 `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](#), and [MrProtocol.cxx](#).

27.63.4.2 `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");

27.63.4.3 `const CSAElement& gdcM::CSAHeader::GetCSAEEnd ( ) const` `[protected]`

27.63.4.4 `const CSAElement& gdcM::CSAHeader::GetCSAElementByName ( const char * name )`

Return the [CSAElement](#) corresponding to name 'name'

#### Warning

Case Sensitive

#### Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

27.63.4.5 `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](#), and [PublicDict.cxx](#).

27.63.4.6 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](#).

27.63.4.7 const DataSet& gdcm::CSAHeader::GetDataSet ( ) const [inline]

Return the DataSet output (use only if Format == DATASET\_FORMAT )

27.63.4.8 CSAHeaderType gdcm::CSAHeader::GetFormat ( ) const

return the format of the CSAHeader SV10 and NOMAGIC are equivalent.

27.63.4.9 const char\* gdcm::CSAHeader::GetInterfile ( ) const [inline]

Return the string output (use only if Format == Interfile)

27.63.4.10 bool gdcm::CSAHeader::LoadFromDataElement ( DataElement const & de )

Decode the CSAHeader from element 'de'.

Examples:

[csa2img.cxx](#), and [MrProtocol.cxx](#).

27.63.4.11 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<<().

27.63.4.12 template<typename TSwap > std::istream& gdcm::CSAHeader::Read ( std::istream & is )

27.63.4.13 template<typename TSwap > const std::ostream& gdcm::CSAHeader::Write ( std::ostream & os ) const

## 27.63.5 Friends And Related Function Documentation

27.63.5.1 std::ostream& operator<< ( std::ostream & \_os, const CSAHeader & d ) [friend]

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

- [gdcmCSAHeader.h](#)

## 27.64 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)

#### 27.64.1 Detailed Description

Class to represent a map of [CSAHeaderDictEntry](#).

Examples:

[MrProtocol.cxx](#).

#### 27.64.2 Member Typedef Documentation

27.64.2.1 typedef MapCSAHeaderDictEntry::const\_iterator gdcm::CSAHeaderDict::ConstIterator

27.64.2.2 typedef MapCSAHeaderDictEntry::iterator gdcm::CSAHeaderDict::Iterator

27.64.2.3 typedef std::set<CSAHeaderDictEntry> gdcm::CSAHeaderDict::MapCSAHeaderDictEntry

### 27.64.3 Constructor & Destructor Documentation

27.64.3.1 `gdcm::CSAHeaderDict::CSAHeaderDict ( )` `[inline]`

### 27.64.4 Member Function Documentation

27.64.4.1 `void gdcm::CSAHeaderDict::AddCSAHeaderDictEntry ( const CSAHeaderDictEntry & de )` `[inline]`

27.64.4.2 `ConstIterator gdcm::CSAHeaderDict::Begin ( ) const` `[inline]`

27.64.4.3 `ConstIterator gdcm::CSAHeaderDict::End ( ) const` `[inline]`

27.64.4.4 `const CSAHeaderDictEntry& gdcm::CSAHeaderDict::GetCSAHeaderDictEntry ( const char * name ) const` `[inline]`

Examples:

[MrProtocol.cxx](#).

27.64.4.5 `bool gdcm::CSAHeaderDict::IsEmpty ( ) const` `[inline]`

27.64.4.6 `void gdcm::CSAHeaderDict::LoadDefault ( )` `[protected]`

### 27.64.5 Friends And Related Function Documentation

27.64.5.1 `friend class Dicts` `[friend]`

27.64.5.2 `std::ostream& operator<< ( std::ostream & _os, const CSAHeaderDict & _val )` `[friend]`

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

- [gdcmCSAHeaderDict.h](#)

## 27.65 gdcm::CSAHeaderDictEntry Class Reference

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.

```
#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)

### 27.65.1 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](#).

### 27.65.2 Constructor & Destructor Documentation

- 27.65.2.1 `gdcm::CSAHeaderDictEntry::CSAHeaderDictEntry ( const char * name = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VM0, const char * desc = " " ) [inline]`

### 27.65.3 Member Function Documentation

- 27.65.3.1 `const char* gdcm::CSAHeaderDictEntry::GetDescription ( ) const [inline]`

Set/Get Description.

- 27.65.3.2 `const char* gdcm::CSAHeaderDictEntry::GetName ( ) const [inline]`

Set/Get Name.

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

- 27.65.3.3 `const VM& gdcm::CSAHeaderDictEntry::GetVM ( ) const [inline]`

Set/Get [VM](#).



27.65.3.4 `const VR& gdcm::CSAHeaderDictEntry::GetVR ( ) const` `[inline]`

Set/Get [VR](#).

27.65.3.5 `bool gdcm::CSAHeaderDictEntry::operator< ( const CSAHeaderDictEntry & entry ) const` `[inline]`

References `GetName()`.

27.65.3.6 `void gdcm::CSAHeaderDictEntry::SetDescription ( const char * desc )` `[inline]`

27.65.3.7 `void gdcm::CSAHeaderDictEntry::SetName ( const char * name )` `[inline]`

27.65.3.8 `void gdcm::CSAHeaderDictEntry::SetVM ( VM const & vm )` `[inline]`

27.65.3.9 `void gdcm::CSAHeaderDictEntry::SetVR ( const VR & vr )` `[inline]`

## 27.65.4 Friends And Related Function Documentation

27.65.4.1 `std::ostream& operator<< ( std::ostream & _os, const CSAHeaderDictEntry & _val )` `[friend]`

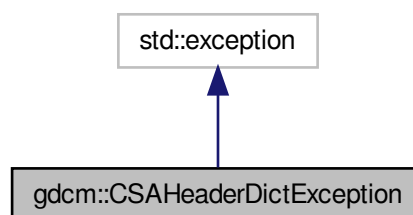
The documentation for this class was generated from the following file:

- [gdcmCSAHeaderDictEntry.h](#)

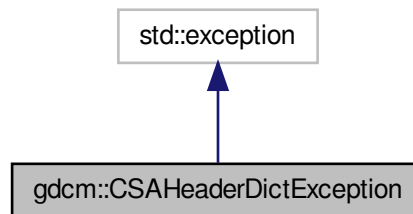
## 27.66 gdcm::CSAHeaderDictException Class Reference

```
#include <gdcmCSAHeaderDict.h>
```

Inheritance diagram for `gdcm::CSAHeaderDictException`:



Collaboration diagram for `gdcm::CSAHeaderDictException`:



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

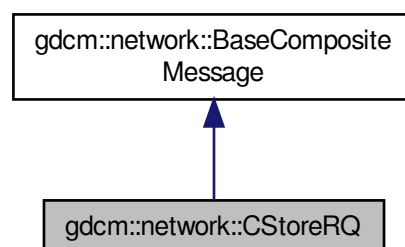
- [gdcmCSAHeaderDict.h](#)

## 27.67 `gdcm::network::CStoreRQ` Class Reference

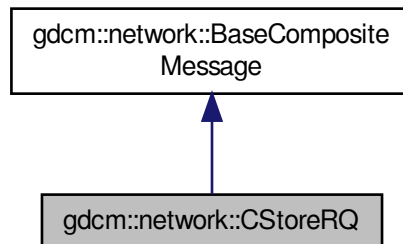
`CStoreRQ` this file defines the messages for the cecho action.

```
#include <gdcmCStoreMessages.h>
```

Inheritance diagram for `gdcm::network::CStoreRQ`:



Collaboration diagram for gdcmm::network::CStoreRQ:



## Public Member Functions

- `std::vector`  
`< PresentationDataValue > ConstructPDV (const ULConnection &inConnection, const File &file)`

### 27.67.1 Detailed Description

`CStoreRQ` this file defines the messages for the cecho action.

### 27.67.2 Member Function Documentation

**27.67.2.1** `std::vector<PresentationDataValue> gdcmm::network::CStoreRQ::ConstructPDV ( const ULConnection & inConnection, const File & file )`

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

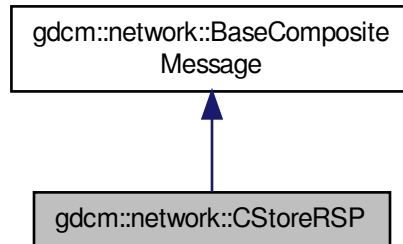
- `gdcmmCStoreMessages.h`

## 27.68 gdcmm::network::CStoreRSP Class Reference

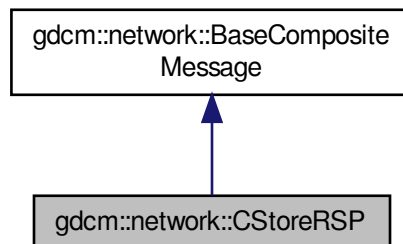
`CStoreRSP` this file defines the messages for the cecho action.

```
#include <gdcmmCStoreMessages.h>
```

Inheritance diagram for `gdcn::network::CStoreRSP`:



Collaboration diagram for `gdcn::network::CStoreRSP`:



## Public Member Functions

- `std::vector`  
`< PresentationDataValue > ConstructPDV (const DataSet *inDataSet, const BasePDU *inPC)`

### 27.68.1 Detailed Description

[CStoreRSP](#) this file defines the messages for the cecho action.

### 27.68.2 Member Function Documentation

27.68.2.1 `std::vector<PresentationDataValue> gdcn::network::CStoreRSP::ConstructPDV ( const DataSet * inDataSet, const BasePDU * inPC )`

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

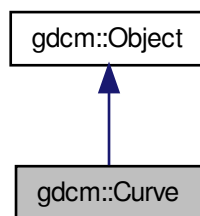
- [gdcmCStoreMessages.h](#)

## 27.69 gdcm::Curve Class Reference

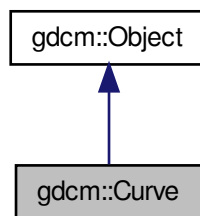
[Curve](#) class to handle element 50xx,3000 [Curve](#) Data WARNING: This is deprecated and lastly defined in PS 3.3 - 2004.

```
#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

#### 27.69.1 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/integriss\_HV\_5000/xa\_integriss.dcm
- TOSHIBA-CurveData[1-3].dcm

#### 27.69.2 Constructor & Destructor Documentation

27.69.2.1 `gdcm::Curve::Curve ( )`

27.69.2.2 `gdcm::Curve::~~Curve ( )`

27.69.2.3 `gdcm::Curve::Curve ( Curve const & ov )`

#### 27.69.3 Member Function Documentation

- 27.69.3.1 void gdcm::Curve::Decode ( std::istream & *is*, std::ostream & *os* )
- 27.69.3.2 void gdcm::Curve::GetAsPoints ( float \* *array* ) const
- 27.69.3.3 std::vector<unsigned short> const& gdcm::Curve::GetCurveDataDescriptor ( ) const
- 27.69.3.4 unsigned short gdcm::Curve::GetDataValueRepresentation ( ) const
- 27.69.3.5 unsigned short gdcm::Curve::GetDimensions ( ) const
- 27.69.3.6 unsigned short gdcm::Curve::GetGroup ( ) const
- 27.69.3.7 static unsigned int gdcm::Curve::GetNumberOfCurves ( DataSet const & *ds* ) [static]
- 27.69.3.8 unsigned short gdcm::Curve::GetNumberOfPoints ( ) const
- 27.69.3.9 const char\* gdcm::Curve::GetTypeOfData ( ) const
- 27.69.3.10 const char\* gdcm::Curve::GetTypeOfDataDescription ( ) const
- 27.69.3.11 bool gdcm::Curve::IsEmpty ( ) const
- 27.69.3.12 void gdcm::Curve::Print ( std::ostream & ) const [virtual]

Reimplemented from [gdcm::Object](#).

- 27.69.3.13 void gdcm::Curve::SetCoordinateStartValue ( unsigned short *v* )
- 27.69.3.14 void gdcm::Curve::SetCoordinateStepValue ( unsigned short *v* )
- 27.69.3.15 void gdcm::Curve::SetCurve ( const char \* *array*, unsigned int *length* )
- 27.69.3.16 void gdcm::Curve::SetCurveDataDescriptor ( const uint16\_t \* *values*, size\_t *num* )
- 27.69.3.17 void gdcm::Curve::SetCurveDescription ( const char \* *curvedescription* )
- 27.69.3.18 void gdcm::Curve::SetDataValueRepresentation ( unsigned short *datavaluerepresentation* )
- 27.69.3.19 void gdcm::Curve::SetDimensions ( unsigned short *dimensions* )
- 27.69.3.20 void gdcm::Curve::SetGroup ( unsigned short *group* )
- 27.69.3.21 void gdcm::Curve::SetNumberOfPoints ( unsigned short *numberofpoints* )
- 27.69.3.22 void gdcm::Curve::SetTypeOfData ( const char \* *typeofdata* )
- 27.69.3.23 void gdcm::Curve::Update ( const DataElement & *de* )

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

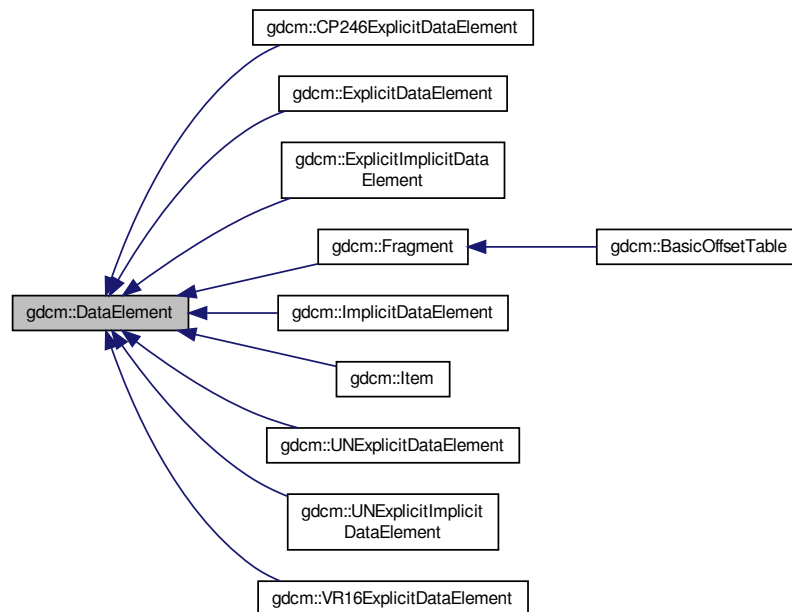
- [gdcmCurve.h](#)

## 27.70 gdcm::DataElement Class Reference

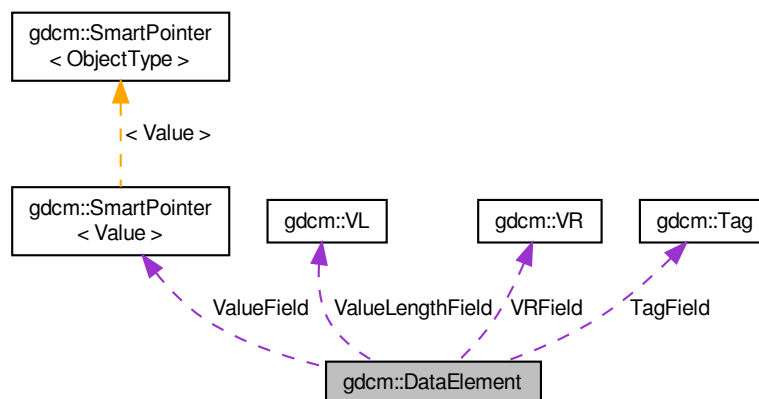
Class to represent a Data [Element](#) either Implicit or Explicit.

```
#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)

## 27.70.1 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:

[ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilips↵  
ECHO.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.↵  
cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FixBroken↵  
J2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.↵  
cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), and [Stream↵  
ImageReaderTest.cxx](#).

## 27.70.2 Member Typedef Documentation

27.70.2.1 `typedef SmartPointer<Value> gdcm::DataElement::ValuePtr` `[protected]`

## 27.70.3 Constructor & Destructor Documentation

27.70.3.1 `gdcm::DataElement::DataElement ( const Tag & t = Tag ( 0 ), const VL & vl = 0, const VR & vr = VR::INVALID )`  
`[inline]`

27.70.3.2 `gdcm::DataElement::DataElement ( const DataElement & _val )` `[inline]`

## 27.70.4 Member Function Documentation

27.70.4.1 `void gdcm::DataElement::Clear ( )` `[inline]`

Clear Data [Element](#) (make [Value](#) empty and invalidate [Tag](#) & [VR](#))

References `gdcm::VR::INVALID`.

Referenced by `gdcm::Item::Clear()`.

27.70.4.2 `void gdcm::DataElement::Empty ( )` `[inline]`

Make Data [Element](#) empty (no [Value](#))

27.70.4.3 `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](#), [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()`.

27.70.4.4 `template<typename TDE> VL gdcm::DataElement::GetLength ( ) const` `[inline]`

27.70.4.5 `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](#).

**27.70.4.6 SequenceOfFragments\*** `gdcm::DataElement::GetSequenceOfFragments ( )`

**27.70.4.7 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()`.

**27.70.4.8 Tag&** `gdcm::DataElement::GetTag ( )` `[inline]`

**27.70.4.9 Value const&** `gdcm::DataElement::GetValue ( ) const` `[inline]`

Set/Get [Value](#) (bytes array, SQ of items, SQ of fragments):

**Examples:**

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::Element< VR::OB, VM::VM1_n >::SetFromDataElement()`, and `gdcm::Element< TVR, VM::VM1_n >::SetFromDataElement()`.

**27.70.4.10 Value&** `gdcm::DataElement::GetValue ( )` `[inline]`

**27.70.4.11 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](#), [ExtractEncryptedContent.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDSExplicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

27.70.4.12 `const VL& gdcm::DataElement::GetVL ( ) const [inline]`

Get [VL](#).

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfItems::Read()`, and `gdcm::SequenceOfFragments::ReadValue()`.

27.70.4.13 `VL& gdcm::DataElement::GetVL ( ) [inline]`

27.70.4.14 `VR const& gdcm::DataElement::GetVR ( ) const [inline]`

Get [VR](#) do not set [VR::SQ](#) on bytevalue data element

Examples:

[DuplicatePCDE.cxx](#), and [GenFakeIdentifyFile.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::Element< TVR, VM::VM1_n >::GetAsDataElement()`, `gdcm::Attribute< Group, 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()`.

27.70.4.15 `bool gdcm::DataElement::IsEmpty ( ) const [inline]`

Check if Data [Element](#) is empty.

Examples:

[DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.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()`.

27.70.4.16 `bool gdcm::DataElement::IsUndefinedLength ( ) const [inline]`

return if [Value](#) Length if of undefined length

27.70.4.17 `bool gdcm::DataElement::operator< ( const DataElement & de ) const [inline]`

References `GetTag()`.

27.70.4.18 `DataElement& gdcm::DataElement::operator= ( const DataElement & de ) [inline]`

References `TagField`, `ValueField`, `ValueLengthField`, and `VRField`.

27.70.4.19 `bool gdcmm::DataElement::operator==( const DataElement & de ) const` `[inline]`

References TagField, ValueField, ValueLengthField, and VRField.

27.70.4.20 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::Read ( std::istream & is )`  
`[inline]`

27.70.4.21 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::ReadOrSkip ( std::istream & is,`  
`std::set< Tag > const & skiptags )` `[inline]`

27.70.4.22 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::ReadPreValue ( std::istream & is,`  
`std::set< Tag > const & skiptags )` `[inline]`

27.70.4.23 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::ReadValue ( std::istream & is,`  
`std::set< Tag > const & skiptags )` `[inline]`

27.70.4.24 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::ReadValueWithLength ( std::istream &`  
`is, VL & length, std::set< Tag > const & skiptags )` `[inline]`

27.70.4.25 `template<typename TDE , typename TSwap > std::istream& gdcmm::DataElement::ReadWithLength ( std::istream & is,`  
`VL & length )` `[inline]`

27.70.4.26 `void gdcmm::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:

[ChangeSequenceUltrasound.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_←\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), and [StreamImageReaderTest.←cxx](#).

Referenced by `gdcmm::Element< VR::OB, VM::VM1_n >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, T←VR, TVM >::GetAsDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement()`, `gdcmm←::Element< TVR, VM::VM1_n >::GetAsDataElement()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::←GetAsDataElement()`.

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

Set [Tag](#) Use with cautious (need to match Part 6)

#### Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenFakeIdentifyFile.cxx](#), and [Get←SubSequenceData.cxx](#).

27.70.4.28 void gdcmm::DataElement::SetValue ( Value const & vl ) [inline]

#### Warning

you need to set the ValueLengthField explicitly

#### Examples:

[DuplicatePCDE.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

References gdcmm::Value::GetLength().

27.70.4.29 void gdcmm::DataElement::SetValueFieldLength ( VL vl, bool readvalues ) [protected]

27.70.4.30 void gdcmm::DataElement::SetVL ( const VL & vl ) [inline]

Set [VL](#) Use with cautious (need to match Part 6), advanced user only

#### See also

[SetByteValue](#)

27.70.4.31 void gdcmm::DataElement::SetVLToUndefined ( )

#### Examples:

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

27.70.4.32 void gdcmm::DataElement::SetVR ( VR const & vr ) [inline]

Set [VR](#) Use with cautious (need to match Part 6), advanced user only

#### Precondition

vr is a [VR::VRALL](#) (not a dual one such as OB\_OW)

#### Examples:

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), and [StreamImageReaderTest.cxx](#).

References gdcmm::VR::IsVRFile().

Referenced by gdcmm::Element< VR::OB, VM::VM1\_n >::GetAsDataElement(), gdcmm::Attribute< Group, Element, T< VR, TVM >::GetAsDataElement(), gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::GetAsDataElement(), gdcmm::Element< TVR, VM::VM1\_n >::GetAsDataElement(), and gdcmm::Attribute< Group, Element, TVR, VM::VM1\_n >::GetAsDataElement().

27.70.4.33 `template<typename TDE , typename TSwap > const std::ostream& gdcm::DataElement::Write ( std::ostream & os )  
const [inline]`

## 27.70.5 Friends And Related Function Documentation

27.70.5.1 `std::ostream& operator<< ( std::ostream & _os, const DataElement & _val ) [friend]`

## 27.70.6 Member Data Documentation

27.70.6.1 `Tag gdcm::DataElement::TagField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.70.6.2 `ValuePtr gdcm::DataElement::ValueField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.70.6.3 `VL gdcm::DataElement::ValueLengthField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

27.70.6.4 `VR gdcm::DataElement::VRField [protected]`

Referenced by `gdcm::operator<<()`, `operator=()`, and `operator==()`.

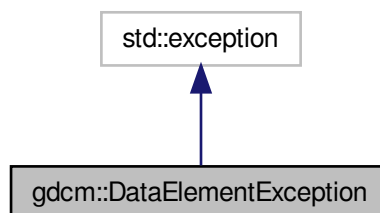
The documentation for this class was generated from the following file:

- [gdcmDataElement.h](#)

## 27.71 gdcm::DataElementException Class Reference

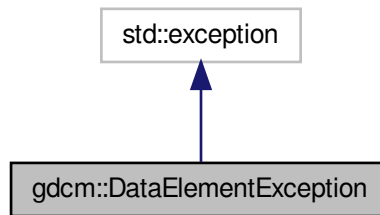
```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcm::DataElementException`:





Collaboration diagram for gdcm::DataElementException:



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

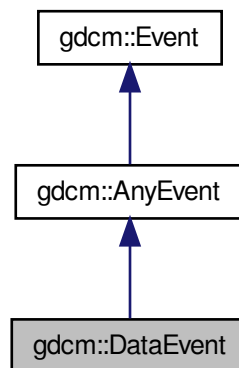
- [gdcmDataSet.h](#)

## 27.72 gdcm::DataEvent Class Reference

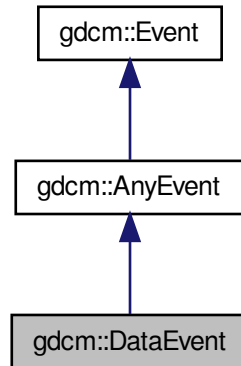
[DataEvent.](#)

```
#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`)

### 27.72.1 Detailed Description

[DataEvent](#).

### 27.72.2 Member Typedef Documentation

27.72.2.1 typedef `DataEvent` `gdcm::DataEvent::Self`

27.72.2.2 typedef `AnyEvent` `gdcm::DataEvent::Superclass`

### 27.72.3 Constructor & Destructor Documentation

27.72.3.1 `gdcm::DataEvent::DataEvent ( const char * bytes = 0, size_t len = 0 )` `[inline]`

27.72.3.2 `virtual gdcm::DataEvent::~~DataEvent ( )` `[inline],[virtual]`

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

### 27.72.4 Member Function Documentation

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

27.72.4.2 `const char* gdcm::DataEvent::GetData ( ) const` `[inline]`

27.72.4.3 `size_t gdcm::DataEvent::GetDataLength ( ) const` `[inline]`

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.72.4.5 `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](#).

27.72.4.6 `void gdcm::DataEvent::SetData ( const char * bytes, size_t len )` `[inline]`

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

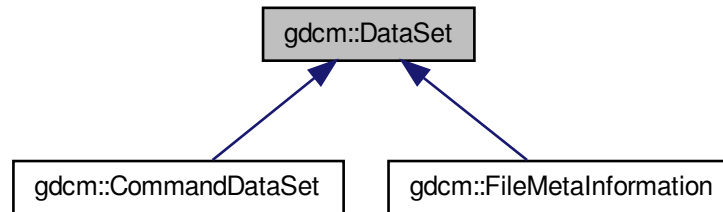
- [gdcmDataEvent.h](#)

## 27.73 gdcm::DataSet Class Reference

Class to represent a Data Set (which contains Data Elements) A Data Set represents an instance of a real world Information [Object](#).

```
#include <gdcmDataSet.h>
```

Inheritance diagram for `gdcm::DataSet`:



## Public Types

- typedef  
DataElementSet::const\_iterator [ConstIterator](#)
- typedef std::set< [DataElement](#) > [DataElementSet](#)
- typedef DataElementSet::iterator [Iterator](#)
- typedef DataElementSet::size\_type [SizeType](#)

## Public Member Functions

- [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)

### 27.73.1 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:

[ChangeSequenceUltrasound.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [gdcmrtpionplan.cxx](#), [gdcmrtpian.cxx](#), [GenAllVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [rle2img.cxx](#), [SortImage.cxx](#), [StreamImageReaderTest.cxx](#), and [VolumeSorter.cxx](#).

### 27.73.2 Member Typedef Documentation

27.73.2.1 `typedef DataElementSet::const_iterator gdcmm::DataSet::ConstIterator`

27.73.2.2 `typedef std::set<DataElement> gdcmm::DataSet::DataElementSet`

27.73.2.3 `typedef DataElementSet::iterator gdcmm::DataSet::Iterator`

27.73.2.4 `typedef DataElementSet::size_type gdcmm::DataSet::SizeType`

### 27.73.3 Member Function Documentation

27.73.3.1 `ConstIterator gdcmm::DataSet::Begin ( ) const` `[inline]`

#### Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

27.73.3.2 `Iterator gdcmm::DataSet::Begin ( )` `[inline]`

27.73.3.3 `void gdcmm::DataSet::Clear ( )` `[inline]`

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

27.73.3.4 **Tag** gdcm::DataSet::ComputeDataElement ( const PrivateTag & t ) const [protected]

27.73.3.5 **template**<typename TDE > unsigned int gdcm::DataSet::ComputeGroupLength ( Tag const & tag ) const [inline]

References gdcm::Tag::GetElement(), and gdcm::Tag::GetGroup().

27.73.3.6 **ConstIterator** gdcm::DataSet::End ( ) const [inline]

Examples:

[DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), and [DuplicatePCDE.cxx](#).

27.73.3.7 **Iterator** gdcm::DataSet::End ( ) [inline]

27.73.3.8 **bool** gdcm::DataSet::FindDataElement ( const PrivateTag & t ) const

Look up if private tag 't' is present in the dataset:

Examples:

[ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.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().

27.73.3.9 **bool** gdcm::DataSet::FindDataElement ( const Tag & t ) const [inline]

27.73.3.10 **const DataElement&** gdcm::DataSet::FindNextDataElement ( const Tag & t ) const [inline]

Examples:

[DuplicatePCDE.cxx](#).

27.73.3.11 **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:

[ChangeSequenceUltrasound.cxx](#), [csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugPEGLS.cxx](#), [gdcmrtonplan.cxx](#), [gdcmrtpplan.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 `gdcmm::Attribute< Group, Element, TVR, TVM >::Set()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::Set()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::Set()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataSet()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataSet()`, and `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataSet()`.

**27.73.3.12** `const DataElement& gdcmm::DataSet::GetDataElement ( const PrivateTag & t ) const`

Return the dataelement.

**27.73.3.13** `const DataElement& gdcmm::DataSet::GetDEEnd ( ) const` `[protected]`

**27.73.3.14** `const DataElementSet& gdcmm::DataSet::GetDES ( ) const` `[inline]`

Examples:

[ReadAndDumpDICOMDIR.cxx](#).

**27.73.3.15** `DataElementSet& gdcmm::DataSet::GetDES ( )` `[inline]`

**27.73.3.16** `template<typename TDE > VL gdcmm::DataSet::GetLength ( ) const` `[inline]`

**27.73.3.17** `MediaStorage gdcmm::DataSet::GetMediaStorage ( ) const`

**27.73.3.18** `std::string gdcmm::DataSet::GetPrivateCreator ( const Tag & t ) const`

Return the private creator of the private tag 't':

Examples:

[DuplicatePCDE.cxx](#).

**27.73.3.19** `void gdcmm::DataSet::Insert ( const DataElement & de )` `[inline]`

Insert a [DataElement](#) in the [DataSet](#).

Warning

: [Tag](#) need to be `>= 0x8` to be considered valid data element

Examples:

[CreateJIPIDataSet.cxx](#), [DuplicatePCDE.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), and [StreamImageReaderTest.cxx](#).

References `gdcmmErrorMacro`, `gdcmm::Tag::GetGroup()`, and `gdcmm::DataElement::GetTag()`.

**27.73.3.20** `void gdcmm::DataSet::InsertDataElement ( const DataElement & de )` `[inline]`, `[protected]`

References `gdcmmWarningMacro`, `gdcmm::Value::GetLength()`, `gdcmm::DataElement::GetValue()`, `gdcmm::DataElement::GetVL()`, and `gdcmm::DataElement::IsEmpty()`.



27.73.3.21 `bool gdcm::DataSet::IsEmpty ( ) const [inline]`

Returns if the dataset is empty.

Referenced by `gdcm::Item::Read()`.

27.73.3.22 `const DataElement& gdcm::DataSet::operator() ( uint16_t group, uint16_t element ) const [inline]`

27.73.3.23 `DataSet& gdcm::DataSet::operator= ( DataSet const & val ) [inline]`

27.73.3.24 `const DataElement& gdcm::DataSet::operator[] ( const Tag & t ) const [inline]`

27.73.3.25 `void gdcm::DataSet::Print ( std::ostream & os, std::string const & indent = " " ) const [inline]`

Referenced by `gdcm::operator<<()`.

27.73.3.26 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::Read ( std::istream & is )`

27.73.3.27 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadNested ( std::istream & is )`

27.73.3.28 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTags ( std::istream & is, const std::set< PrivateTag > & tags, bool readvalues = true )`

27.73.3.29 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedPrivateTagsWithLength ( std::istream & is, const std::set< PrivateTag > & tags, VL & length, bool readvalues = true )`

27.73.3.30 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTags ( std::istream & is, const std::set< Tag > & tags, bool readvalues = true )`

27.73.3.31 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadSelectedTagsWithLength ( std::istream & is, const std::set< Tag > & tags, VL & length, bool readvalues = true )`

27.73.3.32 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTag ( std::istream & is, const Tag & t, std::set< Tag > const & skiptags )`

27.73.3.33 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadUpToTagWithLength ( std::istream & is, const Tag & t, std::set< Tag > const & skiptags, VL & length )`

27.73.3.34 `template<typename TDE , typename TSwap > std::istream& gdcm::DataSet::ReadWithLength ( std::istream & is, VL & length )`

27.73.3.35 `SizeType gdcm::DataSet::Remove ( const Tag & tag ) [inline]`

Completely remove a dataelement from the dataset.

Examples:

[GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

27.73.3.36 `void gdcm::DataSet::Replace ( const DataElement & de ) [inline]`

Replace a dataelement with another one.

Examples:

[ChangeSequenceUltrasound.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeIdentifyFile.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**27.73.3.37** `void gdcm::DataSet::ReplaceEmpty ( const DataElement & de ) [inline]`

Only replace a DICOM attribute when it is missing or empty.

**27.73.3.38** `SizeType gdcm::DataSet::Size ( ) const [inline]`

Examples:

[DumpGEMSMovieGroup.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

**27.73.3.39** `template<typename TDE , typename TSwap > std::ostream const& gdcm::DataSet::Write ( std::ostream & os ) const`

## 27.73.4 Friends And Related Function Documentation

**27.73.4.1** `friend class CSAHeader [friend]`

**27.73.4.2** `std::ostream& operator<< ( std::ostream & _os, const DataSet & val ) [friend]`

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

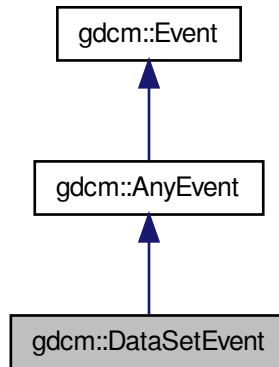
- [gdcmDataSet.h](#)

## 27.74 gdcm::DataSetEvent Class Reference

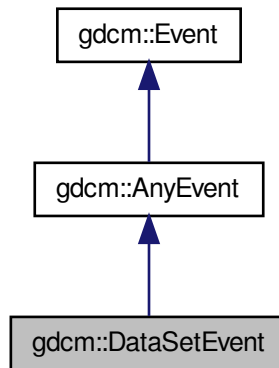
**DataSetEvent** Special type of event triggered during the [DataSet](#) store/move process.

```
#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

### 27.74.1 Detailed Description

[DataSetEvent](#) Special type of event triggered during the [DataSet](#) store/move process.

See also

### 27.74.2 Member Typedef Documentation

27.74.2.1 typedef [DataSetEvent](#) [gdcm::DataSetEvent::Self](#)

27.74.2.2 typedef [AnyEvent](#) [gdcm::DataSetEvent::Superclass](#)

### 27.74.3 Constructor & Destructor Documentation

27.74.3.1 [gdcm::DataSetEvent::DataSetEvent](#) ( [DataSet](#) const \* *ds* = NULL ) [inline]

27.74.3.2 virtual [gdcm::DataSetEvent::~~DataSetEvent](#) ( ) [inline],[virtual]

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

### 27.74.4 Member Function Documentation

27.74.4.1 virtual bool [gdcm::DataSetEvent::CheckEvent](#) ( const [::gdcm::Event](#) \* e ) const [inline],[virtual]

27.74.4.2 [DataSet](#) const& [gdcm::DataSetEvent::GetDataSet](#) ( ) const [inline]

27.74.4.3 virtual const char\* [gdcm::DataSetEvent::GetEventName](#) ( ) const [inline],[virtual]

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.74.4.4 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](#)

## 27.75 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)

### 27.75.1 Detailed Description

[DataSetHelper](#) (internal class, not intended for user level)

### 27.75.2 Member Function Documentation

27.75.2.1 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:

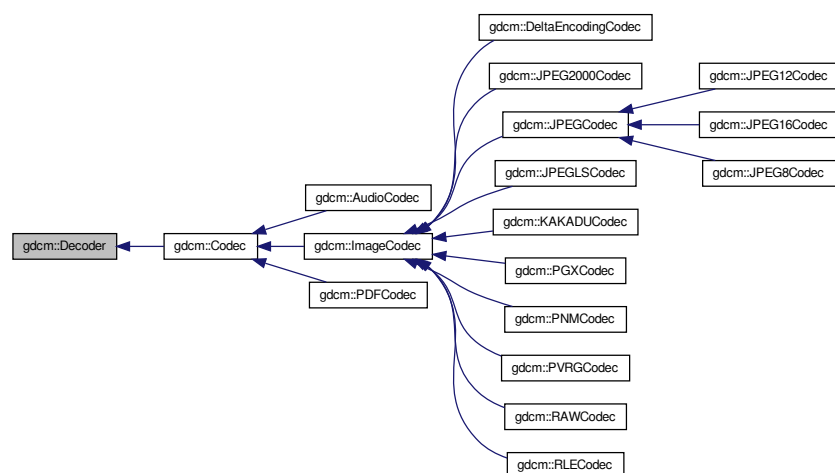
- [gdcmDataSetHelper.h](#)

## 27.76 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 &)

### 27.76.1 Detailed Description

[Decoder](#).

### 27.76.2 Constructor & Destructor Documentation

27.76.2.1 virtual [gdcmm::Decoder::~Decoder](#) ( ) [inline],[virtual]

### 27.76.3 Member Function Documentation

27.76.3.1 virtual bool [gdcmm::Decoder::CanDecode](#) ( [TransferSyntax](#) const & ) const [pure virtual]

Return whether this decoder support this transfer syntax (can decode it)

Implemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::PVRGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::ImageCodec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::PNMCodec](#), [gdcmm::RAWCodec](#), [gdcmm::AudioCodec](#), [gdcmm::PDFCodec](#), [gdcmm::PGXCodec](#), and [gdcmm::KAKADUCodec](#).

27.76.3.2 virtual bool [gdcmm::Decoder::Decode](#) ( [DataElement](#) const &, [DataElement](#) & ) [inline],[virtual]

Decode.

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::RLECodec](#), [gdcmm::JPEGLSCodec](#), [gdcmm::PVRGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::ImageCodec](#), [gdcmm::DeltaEncodingCodec](#), [gdcmm::KAKADUCodec](#), [gdcmm::RAWCodec](#), [gdcmm::AudioCodec](#), and [gdcmm::PDFCodec](#).

27.76.3.3 virtual bool [gdcmm::Decoder::DecodeByStreams](#) ( std::istream &, std::ostream & ) [inline],[protected],[virtual]

Reimplemented in [gdcmm::JPEGCodec](#), [gdcmm::JPEG2000Codec](#), [gdcmm::RLECodec](#), [gdcmm::ImageCodec](#), [gdcmm::RAWCodec](#), [gdcmm::JPEG12Codec](#), [gdcmm::JPEG16Codec](#), and [gdcmm::JPEG8Codec](#).

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

- [gdcmmDecoder.h](#)

## 27.77 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](#) ()

#### 27.77.1 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.

#### 27.77.2 Constructor & Destructor Documentation

27.77.2.1 `gdcm::DefinedTerms::DefinedTerms ( ) [inline]`

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

- [gdcmDefinedTerms.h](#)

## 27.78 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](#)

## 27.78.1 Detailed Description

FIXME I do not like the name '[Defs](#)'.

#### Note

bla

#### Examples:

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

## 27.78.2 Constructor & Destructor Documentation

27.78.2.1 [gdcm::Defs::Defs](#) ( )

27.78.2.2 [gdcm::Defs::~~Defs](#) ( )

## 27.78.3 Member Function Documentation

27.78.3.1 const IOD& [gdcm::Defs::GetIODFromFile](#) ( const [File](#) & *file* ) const

27.78.3.2 static const char\* [gdcm::Defs::GetIODNameFromMediaStorage](#) ( [MediaStorage](#) const & *ms* ) [static]

#### Examples:

[GenerateStandardSOPClasses.cxx](#).



27.78.3.3 `const IODs& gdcm::Defs::GetIODs ( ) const [inline]`

27.78.3.4 `IODs& gdcm::Defs::GetIODs ( ) [inline]`

27.78.3.5 `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

27.78.3.6 `Macros& gdcm::Defs::GetMacros ( ) [inline]`

27.78.3.7 `const Modules& gdcm::Defs::GetModules ( ) const [inline]`

27.78.3.8 `Modules& gdcm::Defs::GetModules ( ) [inline]`

27.78.3.9 `Type gdcm::Defs::GetTypeFromTag ( const File & file, const Tag & tag ) const`

27.78.3.10 `bool gdcm::Defs::IsEmpty ( ) const [inline]`

27.78.3.11 `void gdcm::Defs::LoadDefaults ( ) [protected]`

27.78.3.12 `void gdcm::Defs::LoadFromFile ( const char * filename ) [protected]`

27.78.3.13 `bool gdcm::Defs::Verify ( const File & file ) const`

27.78.3.14 `bool gdcm::Defs::Verify ( const DataSet & ds ) const`

## 27.78.4 Friends And Related Function Documentation

27.78.4.1 `friend class Global [friend]`

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

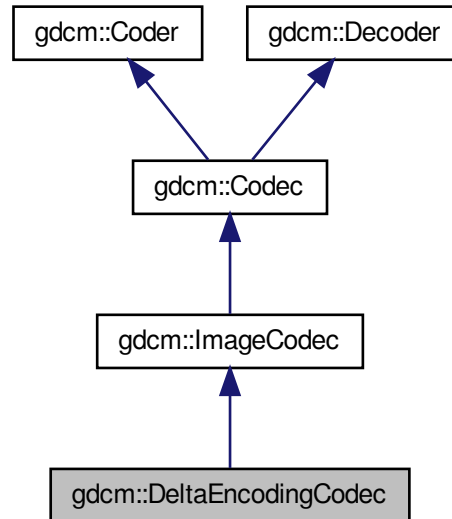
- [gdcmDefs.h](#)

## 27.79 gdcm::DeltaEncodingCodec Class Reference

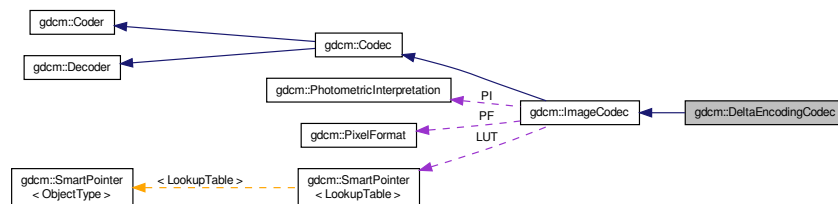
[DeltaEncodingCodec](#) compression used by some private vendor.

```
#include <gdcmDeltaEncodingCodec.h>
```

Inheritance diagram for `gdcm::DeltaEncodingCodec`:



Collaboration diagram for `gdcm::DeltaEncodingCodec`:



## Public Member Functions

- [DeltaEncodingCodec](#) ()
- [~DeltaEncodingCodec](#) ()
- bool [CanDecode](#) ([TransferSyntax](#) const &ts)
- bool [Decode](#) ([DataElement](#) const &is, [DataElement](#) &os)

*Decode.*

## Protected Member Functions

- bool [Decode](#) (std::istream &is, std::ostream &os)

## Additional Inherited Members

### 27.79.1 Detailed Description

[DeltaEncodingCodec](#) compression used by some private vendor.

### 27.79.2 Constructor & Destructor Documentation

27.79.2.1 `gdcm::DeltaEncodingCodec::DeltaEncodingCodec ( )`

27.79.2.2 `gdcm::DeltaEncodingCodec::~~DeltaEncodingCodec ( )`

### 27.79.3 Member Function Documentation

27.79.3.1 `bool gdcm::DeltaEncodingCodec::CanDecode ( TransferSyntax const & ts )`

27.79.3.2 `bool gdcm::DeltaEncodingCodec::Decode ( DataElement const &, DataElement & ) [virtual]`

Decode.

Reimplemented from [gdcm::Decoder](#).

27.79.3.3 `bool gdcm::DeltaEncodingCodec::Decode ( std::istream & is, std::ostream & os ) [protected]`

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

- [gdcmDeltaEncodingCodec.h](#)

## 27.80 gdcm::DICOMDIR Class Reference

[DICOMDIR](#) class.

```
#include <gdcmDICOMDIR.h>
```

### Public Member Functions

- [DICOMDIR](#) ()
- [DICOMDIR](#) (const [FileSet](#) &fs)

### 27.80.1 Detailed Description

[DICOMDIR](#) class.

Structured for handling [DICOMDIR](#)

### 27.80.2 Constructor & Destructor Documentation

27.80.2.1 `gdcm::DICOMDIR::DICOMDIR ( ) [inline]`

27.80.2.2 `gdcm::DICOMDIR::DICOMDIR ( const FileSet & fs ) [inline]`

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

- [gdcmDICOMDIR.h](#)

## 27.81 gdcm::DICOMDIRGenerator Class Reference

[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.

```
#include <gdcmDICOMDIRGenerator.h>
```

### Public Types

- typedef [Directory::FileNamesType](#) FileNamesType
- typedef [Directory::FilenameType](#) FilenameType

### Public Member Functions

- [DICOMDIRGenerator](#) ()
- [~DICOMDIRGenerator](#) ()
- bool [Generate](#) ()  
*Main function to generate the [DICOMDIR](#).*
- [File](#) & [GetFile](#) ()
- void [SetDescriptor](#) (const char \*d)
- void [SetFile](#) (const [File](#) &f)  
*Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.*
- void [SetFileNames](#) ([FileNamesType](#) const &fns)  
*Set the list of filenames from which the [DICOMDIR](#) should be generated from.*
- void [SetRootDirectory](#) ([FilenameType](#) const &root)  
*Set the root directory from which the filenames should be considered.*

### Protected Member Functions

- bool [AddImageDirectoryRecord](#) ()
- bool [AddPatientDirectoryRecord](#) ()
- bool [AddSeriesDirectoryRecord](#) ()
- bool [AddStudyDirectoryRecord](#) ()
- [Scanner](#) & [GetScanner](#) ()

#### 27.81.1 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 [gdcm::Scanner](#) does not allow us See PS 3.11 / [Table D.3-2 STD-GEN Additional DICOMDIR Keys](#)

## Examples:

[GenerateDICOMDIR.cs](#).

## 27.81.2 Member Typedef Documentation

27.81.2.1 `typedef Directory::FilenameType gdcm::DICOMDIRGenerator::FilenameType`

27.81.2.2 `typedef Directory::FileType gdcm::DICOMDIRGenerator::FileType`

## 27.81.3 Constructor &amp; Destructor Documentation

27.81.3.1 `gdcm::DICOMDIRGenerator::DICOMDIRGenerator ( )`

27.81.3.2 `gdcm::DICOMDIRGenerator::~~DICOMDIRGenerator ( )`

## 27.81.4 Member Function Documentation

27.81.4.1 `bool gdcm::DICOMDIRGenerator::AddImageDirectoryRecord ( )` [protected]

27.81.4.2 `bool gdcm::DICOMDIRGenerator::AddPatientDirectoryRecord ( )` [protected]

27.81.4.3 `bool gdcm::DICOMDIRGenerator::AddSeriesDirectoryRecord ( )` [protected]

27.81.4.4 `bool gdcm::DICOMDIRGenerator::AddStudyDirectoryRecord ( )` [protected]

27.81.4.5 `bool gdcm::DICOMDIRGenerator::Generate ( )`

Main function to generate the [DICOMDIR](#).

27.81.4.6 `File& gdcm::DICOMDIRGenerator::GetFile ( )`

27.81.4.7 `Scanner& gdcm::DICOMDIRGenerator::GetScanner ( )` [protected]

27.81.4.8 void gdcmm::DICOMDIRGenerator::SetDescriptor ( const char \* *d* )

Set the [File](#) Set ID.

Warning

this need to be a valid [VR::CS](#) value

27.81.4.9 void gdcmm::DICOMDIRGenerator::SetFile ( const File & *f* )

Set/Get file. The [DICOMDIR](#) file will be valid once a call to Generate has been done.

27.81.4.10 void gdcmm::DICOMDIRGenerator::SetFilenames ( FilenamesType const & *fns* )

Set the list of filenames from which the [DICOMDIR](#) should be generated from.

27.81.4.11 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](#)

## 27.82 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)

### 27.82.1 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](#).

### 27.82.2 Member Typedef Documentation

27.82.2.1 typedef MapDictEntry::const\_iterator gdcm::Dict::ConstIterator

27.82.2.2 typedef MapDictEntry::iterator gdcm::Dict::Iterator

27.82.2.3 typedef std::map<Tag, DictEntry> gdcm::Dict::MapDictEntry

### 27.82.3 Constructor & Destructor Documentation

27.82.3.1 gdcm::Dict::Dict ( ) [\[inline\]](#)

### 27.82.4 Member Function Documentation

27.82.4.1 void gdcm::Dict::AddDictEntry ( const Tag &tag, const DictEntry &de ) [\[inline\]](#)

27.82.4.2 ConstIterator gdcm::Dict::Begin ( ) const [\[inline\]](#)

#### Examples:

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

27.82.4.3 ConstIterator gdcm::Dict::End ( ) const [\[inline\]](#)

#### Examples:

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

27.82.4.4 `const DictEntry& gdcmm::Dict::GetDictEntry ( const Tag & tag ) const` `[inline]`

Examples:

[GenFakelIdentifyFile.cxx](#), and [PublicDict.cxx](#).

27.82.4.5 `const DictEntry& gdcmm::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](#).

27.82.4.6 `const DictEntry& gdcmm::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](#).

27.82.4.7 `const char* gdcmm::Dict::GetKeywordFromTag ( Tag const & tag ) const` `[inline]`

Function to return the Keyword from a [Tag](#).

27.82.4.8 `bool gdcmm::Dict::IsEmpty ( ) const` `[inline]`

Referenced by `gdcmm::Dicts::IsEmpty()`.

27.82.4.9 `void gdcmm::Dict::LoadDefault ( )` `[protected]`

## 27.82.5 Friends And Related Function Documentation

27.82.5.1 `friend class Dicts` `[friend]`

27.82.5.2 `std::ostream& operator<< ( std::ostream & _os, const Dict & _val )` `[friend]`

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

- [gdcmmDict.h](#)

## 27.83 gdcmm::DictConverter Class Reference

Class to convert a .dic file into something else:

```
#include <gdcmmDictConverter.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](#) ()

### 27.83.1 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

## 27.83.2 Member Enumeration Documentation

### 27.83.2.1 enum gdcm::DictConverter::OutputTypes

Enumerator

***DICT\_DEFAULT***

***DICT\_DEBUG***

***DICT\_XML***

## 27.83.3 Constructor & Destructor Documentation

### 27.83.3.1 gdcm::DictConverter::DictConverter ( )

### 27.83.3.2 gdcm::DictConverter::~~DictConverter ( )

## 27.83.4 Member Function Documentation

### 27.83.4.1 void gdcm::DictConverter::AddGroupLength ( ) [protected]

### 27.83.4.2 void gdcm::DictConverter::Convert ( )

### 27.83.4.3 bool gdcm::DictConverter::ConvertToCXX ( const char \* raw, std::string & cxx ) [protected]

### 27.83.4.4 bool gdcm::DictConverter::ConvertToXML ( const char \* raw, std::string & cxx ) [protected]

### 27.83.4.5 const std::string& gdcm::DictConverter::GetDictName ( ) const

### 27.83.4.6 const std::string& gdcm::DictConverter::GetInputFilename ( ) const

### 27.83.4.7 const std::string& gdcm::DictConverter::GetOutputFilename ( ) const

### 27.83.4.8 int gdcm::DictConverter::GetOutputType ( ) const [inline]

### 27.83.4.9 static bool gdcm::DictConverter::Readuint16 ( const char \* raw, uint16\_t & ov ) [static]

### 27.83.4.10 static bool gdcm::DictConverter::ReadVM ( const char \* raw, VM::VMType & type ) [static]

### 27.83.4.11 static bool gdcm::DictConverter::ReadVR ( const char \* raw, VR::VRType & type ) [static]

### 27.83.4.12 void gdcm::DictConverter::SetDictName ( const char \* name )

### 27.83.4.13 void gdcm::DictConverter::SetInputFileName ( const char \* filename )

### 27.83.4.14 void gdcm::DictConverter::SetOutputFileName ( const char \* filename )

### 27.83.4.15 void gdcm::DictConverter::SetOutputType ( int type ) [inline]

### 27.83.4.16 void gdcm::DictConverter::WriteFooter ( ) [protected]

27.83.4.17 void gdcm::DictConverter::WriteHeader ( ) [protected]

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

- [gdcmDictConverter.h](#)

## 27.84 gdcm::DictEntry Class Reference

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.

```
#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)

### 27.84.1 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](#), [GenFakelIdentifyFile.cxx](#), [PublicDict.cxx](#), and [TraverseModules.cxx](#).

### 27.84.2 Constructor & Destructor Documentation

27.84.2.1 `gdcm::DictEntry::DictEntry ( const char * name = " ", const char * keyword = " ", VR const & vr = VR::INVALID, VM const & vm = VM::VMO, bool ret = false ) [inline]`

### 27.84.3 Member Function Documentation

27.84.3.1 `const char* gdcm::DictEntry::GetKeyword ( ) const [inline]`

same as GetName but without spaces...

27.84.3.2 `const char* gdcm::DictEntry::GetName ( ) const [inline]`

Set/Get Name.

Referenced by `gdcm::PrivateDict::PrintXML()`.

27.84.3.3 `bool gdcm::DictEntry::GetRetired ( ) const [inline]`

Set/Get Retired flag.

#### Examples:

[GenAllVR.cxx](#).

27.84.3.4 `const VM& gdcm::DictEntry::GetVM ( ) const [inline]`

Set/Get [VM](#).

Referenced by `gdcm::PrivateDict::AddDictEntry()`, and `gdcm::PrivateDict::PrintXML()`.

27.84.3.5 `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()`.

27.84.3.6 `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)

27.84.3.7 `void gdcm::DictEntry::SetElementXX ( bool v )` `[inline]`

Set whether element is shared in multiple elements (Source [Image](#) IDs typically)

27.84.3.8 `void gdcm::DictEntry::SetGroupXX ( bool v )` `[inline]`

Set whether element is shared in multiple groups (Curve/Overlay typically)

27.84.3.9 `void gdcm::DictEntry::SetKeyword ( const char * keyword )` `[inline]`

27.84.3.10 `void gdcm::DictEntry::SetName ( const char * name )` `[inline]`

27.84.3.11 `void gdcm::DictEntry::SetRetired ( bool retired )` `[inline]`

27.84.3.12 `void gdcm::DictEntry::SetVM ( VM const & vm )` `[inline]`

27.84.3.13 `void gdcm::DictEntry::SetVR ( const VR & vr )` `[inline]`

Referenced by `gdcm::PrivateDict::AddDictEntry()`.

## 27.84.4 Friends And Related Function Documentation

27.84.4.1 `friend class Dict` `[friend]`

27.84.4.2 `std::ostream& operator<< ( std::ostream & _os, const DictEntry & _val )` `[friend]`

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

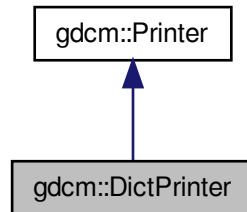
- [gdcmDictEntry.h](#)

## 27.85 gdcm::DictPrinter Class Reference

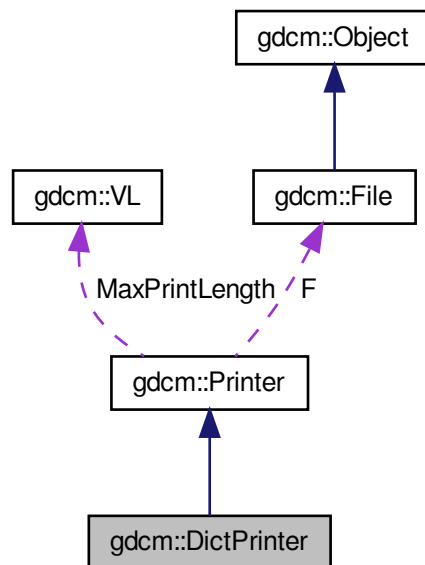
[DictPrinter](#) class.

```
#include <gdcmDictPrinter.h>
```

Inheritance diagram for `gdcm::DictPrinter`:



Collaboration diagram for `gdcm::DictPrinter`:



## 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

### 27.85.1 Detailed Description

[DictPrinter](#) class.

### 27.85.2 Constructor & Destructor Documentation

27.85.2.1 [gdcm::DictPrinter::DictPrinter](#) ( )

27.85.2.2 [gdcm::DictPrinter::~~DictPrinter](#) ( )

### 27.85.3 Member Function Documentation

27.85.3.1 void [gdcm::DictPrinter::Print](#) ( std::ostream & os )

27.85.3.2 void [gdcm::DictPrinter::PrintDataElement2](#) ( std::ostream & os, const [DataSet](#) & ds, const [DataElement](#) & ide )  
[protected]

27.85.3.3 void [gdcm::DictPrinter::PrintDataSet2](#) ( std::ostream & os, const [DataSet](#) & ds ) [protected]

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

- [gdcmDictPrinter.h](#)

## 27.86 gdcm::Dicts Class Reference

Class to manipulate the sum of knowledge (all the dict user load)

```
#include <gdcmDicts.h>
```

## Public Member Functions

- [Dicts](#) ()
- [~Dicts](#) ()
- const [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)

### 27.86.1 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](#).

### 27.86.2 Member Enumeration Documentation

#### 27.86.2.1 enum `gdcm::Dicts::ConstructorType` [protected]

##### Enumerator

***PHILIPS***

***GEMS***

***SIEMENS***

### 27.86.3 Constructor & Destructor Documentation

#### 27.86.3.1 `gdcm::Dicts::Dicts ( )`

#### 27.86.3.2 `gdcm::Dicts::~~Dicts ( )`

### 27.86.4 Member Function Documentation



27.86.4.1 `static const char* gdcmm::Dicts::GetConstructorString ( ConstructorType type )` `[static]`, `[protected]`

27.86.4.2 `const CSAHeaderDict& gdcmm::Dicts::GetCSAHeaderDict ( ) const`

Examples:

[MrProtocol.cxx](#).

27.86.4.3 `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](#).

27.86.4.4 `const DictEntry& gdcmm::Dicts::GetDictEntry ( const PrivateTag & tag ) const`

27.86.4.5 `const PrivateDict& gdcmm::Dicts::GetPrivateDict ( ) const`

27.86.4.6 `PrivateDict& gdcmm::Dicts::GetPrivateDict ( )`

27.86.4.7 `const Dict& gdcmm::Dicts::GetPublicDict ( ) const`

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

27.86.4.8 `bool gdcmm::Dicts::IsEmpty ( ) const` `[inline]`

References `gdcmm::Dict::IsEmpty()`.

27.86.4.9 `void gdcmm::Dicts::LoadDefaults ( )` `[protected]`

## 27.86.5 Friends And Related Function Documentation

27.86.5.1 `friend class Global` `[friend]`

27.86.5.2 `std::ostream& operator<< ( std::ostream & _os, const Dicts & d )` `[friend]`

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

- [gdcmmDicts.h](#)

## 27.87 gdcm::network::DIMSE Class Reference

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)

```
#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 }

### 27.87.1 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)

### 27.87.2 Member Enumeration Documentation

#### 27.87.2.1 enum gdcm::network::DIMSE::CommandTypes

Enumerator

***C\_STORE\_RQ***  
***C\_STORE\_RSP***  
***C\_GET\_RQ***  
***C\_GET\_RSP***  
***C\_FIND\_RQ***  
***C\_FIND\_RSP***

***C\_MOVE\_RQ***  
***C\_MOVE\_RSP***  
***C\_ECHO\_RQ***  
***C\_ECHO\_RSP***  
***N\_EVENT\_REPORT\_RQ***  
***N\_EVENT\_REPORT\_RSP***  
***N\_GET\_RQ***  
***N\_GET\_RSP***  
***N\_SET\_RQ***  
***N\_SET\_RSP***  
***N\_ACTION\_RQ***  
***N\_ACTION\_RSP***  
***N\_CREATE\_RQ***  
***N\_CREATE\_RSP***  
***N\_DELETE\_RQ***  
***N\_DELETE\_RSP***  
***C\_CANCEL\_RQ***

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

- [gdcmDIMSE.h](#)

## 27.88 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)`

### 27.88.1 Detailed Description

class to handle [DirectionCosines](#)

Examples:

[DiscriminateVolume.cxx](#).

### 27.88.2 Constructor & Destructor Documentation

27.88.2.1 `gdc::DirectionCosines::DirectionCosines ( )`

27.88.2.2 `gdc::DirectionCosines::DirectionCosines ( const double dircos[6] )`

27.88.2.3 `gdc::DirectionCosines::~~DirectionCosines ( )`

### 27.88.3 Member Function Documentation

27.88.3.1 `double gdc::DirectionCosines::ComputeDistAlongNormal ( const double ipp[3] ) const`

Compute the distance along the normal.

27.88.3.2 `void gdc::DirectionCosines::Cross ( double z[3] ) const`

Compute Cross product.

27.88.3.3 `double gdc::DirectionCosines::CrossDot ( DirectionCosines const & dc ) const`

Compute the Dot product of the two cross vector of both [DirectionCosines](#) object.

Examples:

[DiscriminateVolume.cxx](#).

27.88.3.4 `double gdc::DirectionCosines::Dot ( ) const`

Compute Dot.

27.88.3.5 `bool gdc::DirectionCosines::IsValid ( ) const`

Return whether or not this is a valid direction cosines.

27.88.3.6 void gdcm::DirectionCosines::Normalize ( )

Normalize in-place.

27.88.3.7 gdcm::DirectionCosines::operator const double \* ( ) const [inline]

Make the class behave like a const double \*.

27.88.3.8 void gdcm::DirectionCosines::Print ( std::ostream & ) const

Print.

27.88.3.9 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](#)

## 27.89 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)

### 27.89.1 Detailed Description

Class for manipulation directories.

#### Note

This implementation provide a cross platform implementation for manipulating diretores: 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 [gdcmm::Directory](#) does not make much sense.

#### Examples:

[DecompressImageMultiframe.cs](#), [DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

### 27.89.2 Member Typedef Documentation

27.89.2.1 `typedef std::vector<FilenameType> gdcmm::Directory::FilenamesType`

#### Examples:

[DiscriminateVolume.cxx](#).

27.89.2.2 `typedef std::string gdcmm::Directory::FilenameType`

### 27.89.3 Constructor & Destructor Documentation

27.89.3.1 `gdcmm::Directory::Directory ( ) [inline]`

27.89.3.2 `gdcmm::Directory::~~Directory ( ) [inline]`

### 27.89.4 Member Function Documentation

27.89.4.1 `unsigned int gdcmm::Directory::Explore ( FilenameType const & name, bool recursive ) [protected]`

Return number of file found when 'recursive'ly exploring directory *name*

**27.89.4.2** `FilenameType const& gdcmm::Directory::GetDirectories ( ) const` `[inline]`

Return the Directories traversed.

**27.89.4.3** `FilenameType const& gdcmm::Directory::GetFileNames ( ) const` `[inline]`

Set/Get the file names within the directory.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

**27.89.4.4** `FilenameType const& gdcmm::Directory::GetToplevel ( ) const` `[inline]`

Get the name of the toplevel directory.

**27.89.4.5** `unsigned int gdcmm::Directory::Load ( FilenameType const & name, bool recursive = false )` `[inline]`

construct a list of filenames and subdirectory beneath directory: name

Warning

: hidden file and hidden directory are not loaded.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [gdcmmorthoplanes.cxx](#), [GenerateRTSTRUCT.cxx](#), [ReadUTF8QtDir.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), [threadgdcmm.cxx](#), and [VolumeSorter.cxx](#).

**27.89.4.6** `void gdcmm::Directory::Print ( std::ostream & os = std::cout ) const`

Print.

Examples:

[SortImage.cxx](#).

Referenced by `gdcmm::operator<<()`.

**27.89.5 Friends And Related Function Documentation****27.89.5.1** `std::ostream& operator<< ( std::ostream & _os, const Directory & d )` `[friend]`

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

- [gdcmmDirectory.h](#)

## 27.90 gdcm::DirectoryHelper Class Reference

[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.

```
#include <gdcmDirectoryHelper.h>
```

### Static Public Member Functions

- static [Directory::FilenamesType GetCTImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType GetFileNamesFromSeriesUIDs](#) (const std::string &inDirectory, const std::string &inSeriesUID)
- static std::string [GetFrameOfReference](#) (const std::vector< [DataSet](#) > &inDS)
- static [Directory::FilenamesType GetMRIImageSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType GetRTStructSeriesUIDs](#) (const std::string &inDirectory)
- static [Directory::FilenamesType GetSeriesUIDsBySOPClassUID](#) (const std::string &inDirectory, const std::string &inSOPClassUID)
- static std::string [GetSOPClassUID](#) (const std::vector< [DataSet](#) > &inDS)
- static std::string [GetStringValueFromTag](#) (const [gdcm::Tag](#) &t, const [gdcm::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)

### 27.90.1 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.

### 27.90.2 Member Function Documentation

**27.90.2.1** static [Directory::FilenamesType](#) [gdcm::DirectoryHelper::GetCTImageSeriesUIDs](#) ( const std::string & *inDirectory* )  
[static]

**27.90.2.2** static [Directory::FilenamesType](#) [gdcm::DirectoryHelper::GetFileNamesFromSeriesUIDs](#) ( const std::string & *inDirectory*, const std::string & *inSeriesUID* ) [static]

Examples:

[GenerateRTSTRUCT.cxx](#).

**27.90.2.3** static std::string [gdcm::DirectoryHelper::GetFrameOfReference](#) ( const std::vector< [DataSet](#) > & *inDS* ) [static]

**27.90.2.4** static [Directory::FilenamesType](#) [gdcm::DirectoryHelper::GetMRIImageSeriesUIDs](#) ( const std::string & *inDirectory* )  
[static]



27.90.2.5 `static Directory::FileNamesType gdcm::DirectoryHelper::GetRTStructSeriesUIDs ( const std::string & inDirectory )`  
`[static]`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.90.2.6 `static Directory::FileNamesType gdcm::DirectoryHelper::GetSeriesUIDsBySOPClassUID ( const std::string & inDirectory, const std::string & inSOPClassUID )` `[static]`

27.90.2.7 `static std::string gdcm::DirectoryHelper::GetSOPClassUID ( const std::vector< DataSet > & inDS )` `[static]`

27.90.2.8 `static std::string gdcm::DirectoryHelper::GetStringValueFromTag ( const gdcm::Tag & t, const gdcm::DataSet & ds )`  
`[static]`

27.90.2.9 `static std::vector<DataSet> gdcm::DirectoryHelper::LoadImageFromFiles ( const std::string & inDirectory, const std::string & inSeriesUID )` `[static]`

27.90.2.10 `static std::string gdcm::DirectoryHelper::RetrieveSOPInstanceUIDFromIndex ( int inIndex, const std::vector< DataSet > & inDS )` `[static]`

27.90.2.11 `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](#)

## 27.91 gdcm::DummyValueGenerator Class Reference

Class for generating dummy value.

```
#include <gdcmDummyValueGenerator.h>
```

### Static Public Member Functions

- static const char \* [Generate](#) (const char \*input)

### 27.91.1 Detailed Description

Class for generating dummy value.

See also

[Anonymizer](#)

### 27.91.2 Member Function Documentation

27.91.2.1 `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:

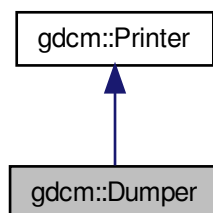
- [gdcmDummyValueGenerator.h](#)

## 27.92 gdcm::Dumper Class Reference

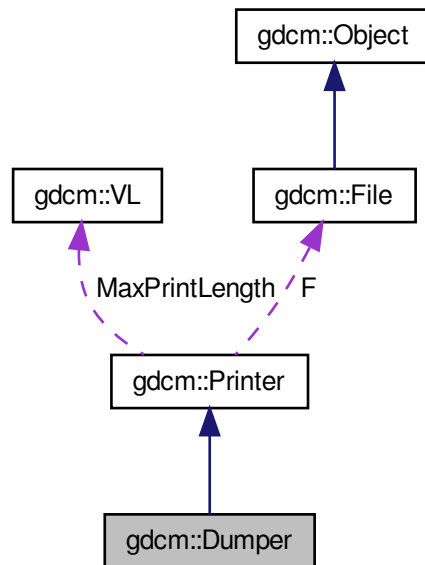
[Codec](#) class.

```
#include <gdcmDumper.h>
```

Inheritance diagram for `gdcm::Dumper`:



Collaboration diagram for gdcmm::Dumper:



### Public Member Functions

- [Dumper](#) ()
- [~Dumper](#) ()

### Additional Inherited Members

#### 27.92.1 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

#### 27.92.2 Constructor & Destructor Documentation

27.92.2.1 `gdcmm::Dumper::Dumper ( )` `[inline]`

27.92.2.2 `gdcmm::Dumper::~~Dumper ( )` `[inline]`

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

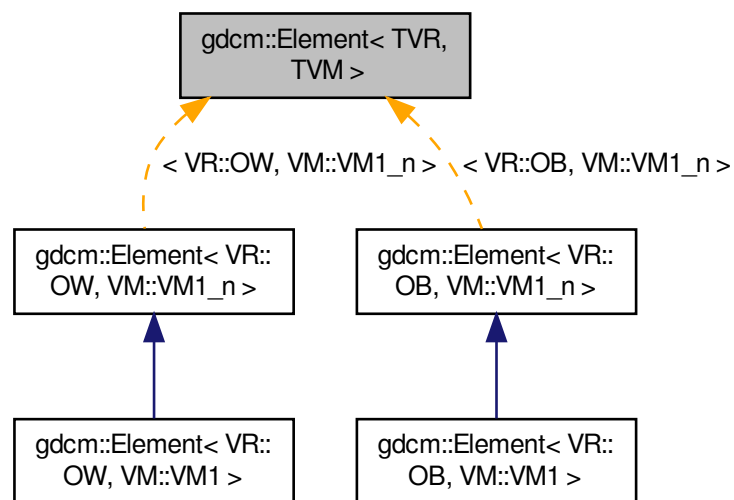
- [gdcmmDumper.h](#)

## 27.93 gdcmm::Element< TVR, TVM > Class Template Reference

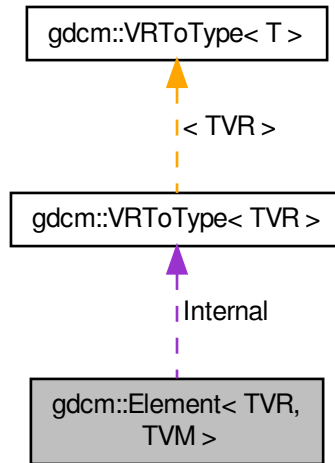
[Element](#) class.

```
#include <gdcmmElement.h>
```

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



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



## Public Types

- typedef `VRToType< TVR >::Type` `Type`

## Public Member Functions

- `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)
- const `VRToType< TVR >::Type * GetValues ()` const
- `VRToType< TVR >::Type operator[]` (unsigned int idx) const
- void `Print` (std::ostream &\_os) const
- void `Read` (std::istream &\_is)
- void `Set` (Value const &v)
- void `SetFromDataElement` (DataElement const &de)
- void `SetValue` (typename `VRToType< TVR >::Type` v, unsigned int idx=0)
- void `Write` (std::ostream &\_os) const

## Static Public Member Functions

- static `VM GetVM ()`
- static `VR GetVR ()`

## Public Attributes

- [VRToType](#)< TVR >::Type [Internal](#) [[VMToLength](#)< TVM >::Length]

## Protected Member Functions

- void [SetNoSwap](#) ([Value](#) const &v)

### 27.93.1 Detailed Description

template<int TVR, int TVM>class [gdcmm::Element](#)< TVR, TVM >

[Element](#) class.

Note

TODO

Examples:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GetSubSequenceData.cxx](#), and [iU22tomultisc.cxx](#).

### 27.93.2 Member Typedef Documentation

27.93.2.1 template<int TVR, int TVM> typedef [VRToType](#)<TVR>::Type [gdcmm::Element](#)< TVR, TVM >::Type

### 27.93.3 Member Function Documentation

27.93.3.1 template<int TVR, int TVM> [DataElement](#) [gdcmm::Element](#)< TVR, TVM >::GetAsDataElement ( ) const  
[inline]

27.93.3.2 template<int TVR, int TVM> unsigned long [gdcmm::Element](#)< TVR, TVM >::GetLength ( ) const [inline]

27.93.3.3 template<int TVR, int TVM> const [VRToType](#)<TVR>::Type& [gdcmm::Element](#)< TVR, TVM >::GetValue ( unsigned int *idx* = 0 ) const [inline]

27.93.3.4 template<int TVR, int TVM> [VRToType](#)<TVR>::Type& [gdcmm::Element](#)< TVR, TVM >::GetValue ( unsigned int *idx* = 0 ) [inline]

27.93.3.5 template<int TVR, int TVM> const [VRToType](#)<TVR>::Type\* [gdcmm::Element](#)< TVR, TVM >::GetValues ( ) const  
[inline]

27.93.3.6 template<int TVR, int TVM> static VM [gdcmm::Element](#)< TVR, TVM >::GetVM ( ) [inline],[static]

27.93.3.7 template<int TVR, int TVM> static VR [gdcmm::Element](#)< TVR, TVM >::GetVR ( ) [inline],[static]

27.93.3.8 template<int TVR, int TVM> [VRToType](#)<TVR>::Type [gdcmm::Element](#)< TVR, TVM >::operator[] ( unsigned int *idx* ) const [inline]

27.93.3.9 template<int TVR, int TVM> void [gdcmm::Element](#)< TVR, TVM >::Print ( std::ostream &\_os ) const [inline]

27.93.3.10 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Read ( std::istream & _is ) [inline]`

27.93.3.11 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Set ( Value const & v ) [inline]`

27.93.3.12 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetFromDataElement ( DataElement< TVR, TVM > const & de ) [inline]`

27.93.3.13 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetNoSwap ( Value const & v ) [inline], [protected]`

27.93.3.14 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::SetValue ( typename VRToType< TVR >::Type v, unsigned int idx = 0 ) [inline]`

27.93.3.15 `template<int TVR, int TVM> void gdcm::Element< TVR, TVM >::Write ( std::ostream & _os ) const [inline]`

## 27.93.4 Member Data Documentation

27.93.4.1 `template<int TVR, int TVM> VRToType<TVR>::Type gdcm::Element< TVR, TVM >::Internal[VMToLength< TVM >::Length]`

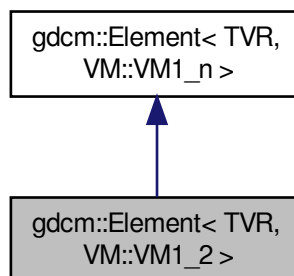
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

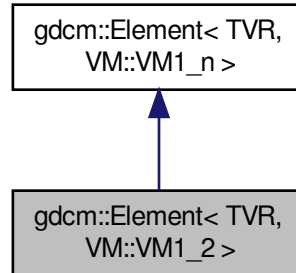
## 27.94 gdcm::Element< TVR, VM::VM1\_2 > Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM1_2 >`:



Collaboration diagram for `gdcm::Element< TVR, VM::VM1_2 >`:



## Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

## Public Member Functions

- void `SetLength` (int len)

## Additional Inherited Members

### 27.94.1 Member Typedef Documentation

27.94.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM1_2 >::Parent`

### 27.94.2 Member Function Documentation

27.94.2.1 `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:

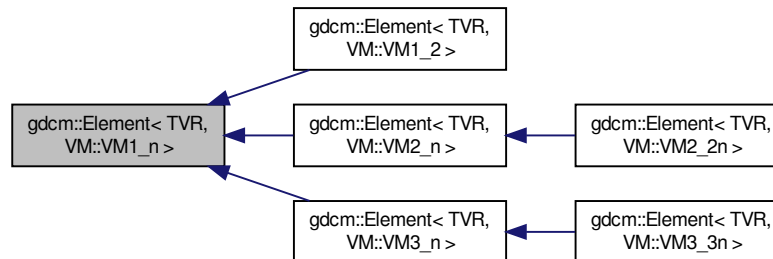
- `gdcmElement.h`

## 27.95 `gdcm::Element< TVR, VM::VM1_n >` Class Template Reference

```
#include <gdcmElement.h>
```



Inheritance diagram for gdcmm::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)

### 27.95.1 Member Typedef Documentation

27.95.1.1 `template<int TVR> typedef VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::Type`

### 27.95.2 Constructor & Destructor Documentation

27.95.2.1 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element ( ) [inline],[explicit]`

27.95.2.2 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::~~Element ( ) [inline]`

27.95.2.3 `template<int TVR> gdcm::Element< TVR, VM::VM1_n >::Element ( const Element< TVR, VM::VM1_n > & _val ) [inline]`

### 27.95.3 Member Function Documentation

27.95.3.1 `template<int TVR> DataElement gdcm::Element< TVR, VM::VM1_n >::GetAsDataElement ( ) const [inline]`

References `gdcm::DataElement::GetVR()`, `gdcm::DataElement::SetByteValue()`, `gdcm::DataElement::SetVR()`, `gdcm::VR::SQ`, `gdcm::VR::UI`, and `gdcm::VR::VRASCII`.

27.95.3.2 `template<int TVR> unsigned long gdcm::Element< TVR, VM::VM1_n >::GetLength ( ) const [inline]`

27.95.3.3 `template<int TVR> const VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue ( unsigned int idx = 0 ) const [inline]`

27.95.3.4 `template<int TVR> VRToType<TVR>::Type& gdcm::Element< TVR, VM::VM1_n >::GetValue ( unsigned int idx = 0 ) [inline]`

27.95.3.5 `template<int TVR> static VM gdcm::Element< TVR, VM::VM1_n >::GetVM ( ) [inline],[static]`

References `gdcm::VM::VM1_n`.

27.95.3.6 `template<int TVR> static VR gdcm::Element< TVR, VM::VM1_n >::GetVR ( ) [inline],[static]`

27.95.3.7 `template<int TVR> Element& gdcm::Element< TVR, VM::VM1_n >::operator= ( const Element< TVR, VM::VM1_n > &_val ) [inline]`

27.95.3.8 `template<int TVR> VRToType<TVR>::Type gdcm::Element< TVR, VM::VM1_n >::operator[] ( unsigned int idx ) const [inline]`

27.95.3.9 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Print ( std::ostream &_os ) const [inline]`

27.95.3.10 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Read ( std::istream &_is ) [inline]`

27.95.3.11 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Set ( Value const & v ) [inline]`

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

27.95.3.12 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetArray ( const Type * array, unsigned long len, bool save = false ) [inline]`

27.95.3.13 `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()`, `gdcm::DataElement::GetVR()`, `gdcm::VR::INVALID`, and `gdcm::VR::UN`.

27.95.3.14 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetLength ( unsigned long len ) [inline]`

27.95.3.15 `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::VR::VRBINARY`.

27.95.3.16 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::SetValue ( typename VRToType< TVR >::Type v, unsigned int idx = 0 ) [inline]`

27.95.3.17 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::Write ( std::ostream & _os ) const [inline]`

27.95.3.18 `template<int TVR> void gdcm::Element< TVR, VM::VM1_n >::WriteASCII ( std::ostream & os ) const [inline]`

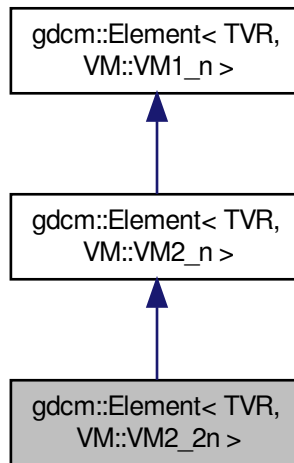
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

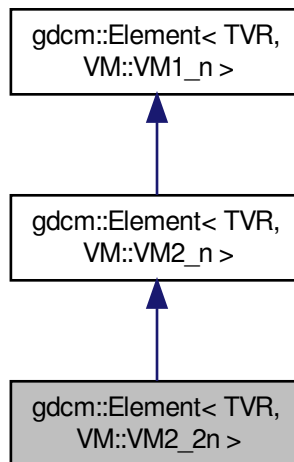
## 27.96 gdcm::Element< TVR, VM::VM2\_2n > Class Template Reference

```
#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

### 27.96.1 Member Typedef Documentation

27.96.1.1 `template<int TVR> typedef Element<TVR, VM::VM2_n> gdcm::Element< TVR, VM::VM2_2n >::Parent`

### 27.96.2 Member Function Documentation

27.96.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM2_2n >::SetLength ( int len ) [inline]`

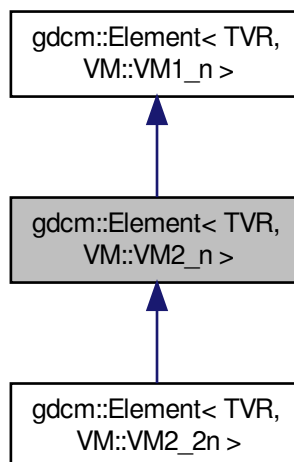
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

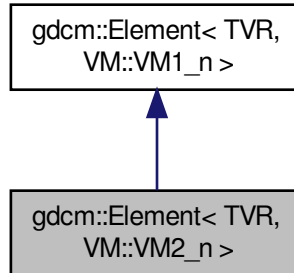
## 27.97 `gdcm::Element< TVR, VM::VM2_n >` Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for `gdcm::Element< TVR, VM::VM2_n >`:



Collaboration diagram for `gdcm::Element< TVR, VM::VM2_n >`:



## Public Types

- typedef `Element< TVR, VM::VM1_n >` `Parent`

## Public Member Functions

- void `SetLength` (int *len*)

## Additional Inherited Members

### 27.97.1 Member Typedef Documentation

27.97.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM2_n >::Parent`

### 27.97.2 Member Function Documentation

27.97.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM2_n >::SetLength ( int len ) [inline]`

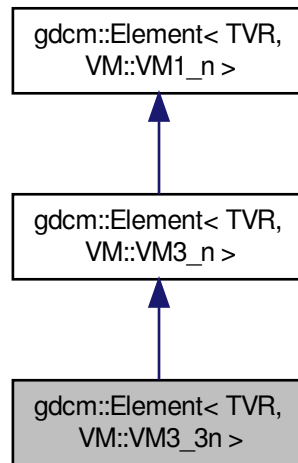
The documentation for this class was generated from the following file:

- `gdcmElement.h`

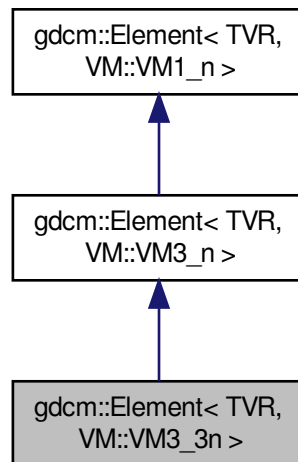
## 27.98 `gdcm::Element< TVR, VM::VM3_3n >` Class Template Reference

```
#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

### 27.98.1 Member Typedef Documentation

27.98.1.1 `template<int TVR> typedef Element<TVR, VM::VM3_n> gdcm::Element< TVR, VM::VM3_3n >::Parent`

### 27.98.2 Member Function Documentation

27.98.2.1 `template<int TVR> void gdcm::Element< TVR, VM::VM3_3n >::SetLength ( int len ) [inline]`

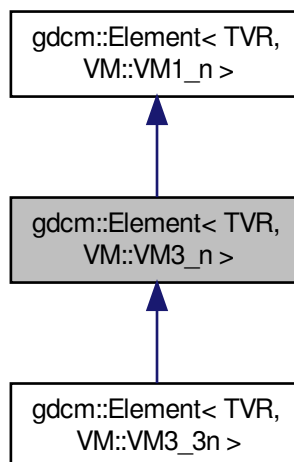
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

## 27.99 gdcm::Element< TVR, VM::VM3\_n > Class Template Reference

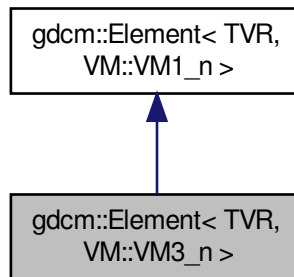
```
#include <gdcmElement.h>
```

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

#### 27.99.1 Member Typedef Documentation

27.99.1.1 `template<int TVR> typedef Element<TVR, VM::VM1_n> gdcm::Element< TVR, VM::VM3_n >::Parent`

#### 27.99.2 Member Function Documentation

27.99.2.1 `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](#)

## 27.100 `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)]

### 27.100.1 Member Function Documentation

27.100.1.1 unsigned long [gdcm::Element](#)< [VR::AS](#), [VM::VM5](#)>::GetLength ( ) const [\[inline\]](#)

27.100.1.2 void [gdcm::Element](#)< [VR::AS](#), [VM::VM5](#)>::Print ( std::ostream & \_os ) const [\[inline\]](#)

### 27.100.2 Member Data Documentation

27.100.2.1 char [gdcm::Element](#)< [VR::AS](#), [VM::VM5](#)>::Internal[[VMToLength](#)< [VM::VM5](#)>::Length \*sizeof([VRToType](#)< [VR::AS](#)>::Type)]

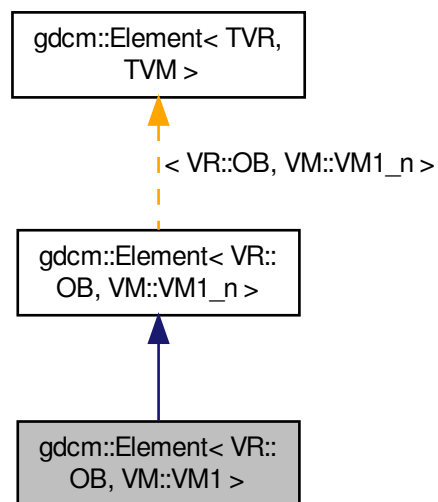
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

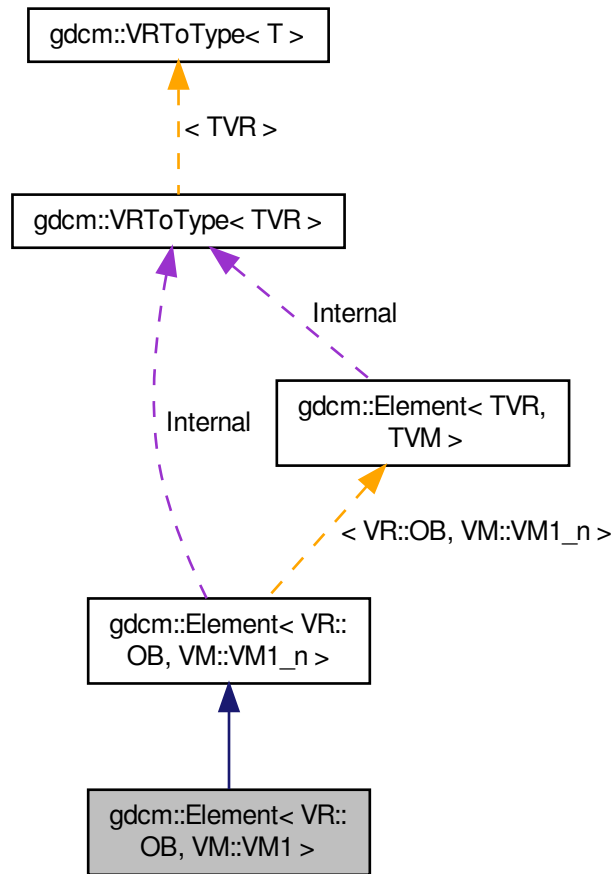
## 27.101 [gdcm::Element](#)< [VR::OB](#), [VM::VM1](#)> Class Template Reference

```
#include <gdcmElement.h>
```

Inheritance diagram for [gdcm::Element](#)< [VR::OB](#), [VM::VM1](#)>:



Collaboration diagram for gdcM::Element< VR::OB, VM::VM1 >:



### Additional Inherited Members

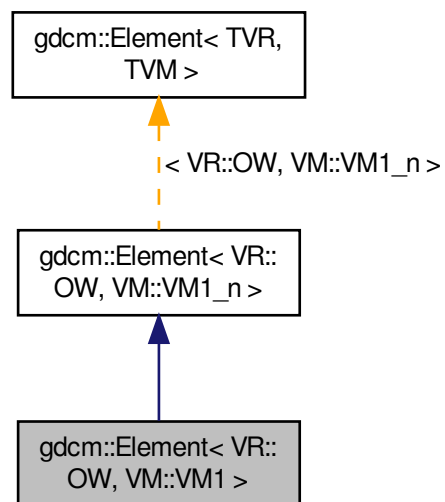
The documentation for this class was generated from the following file:

- [gdcMElement.h](#)

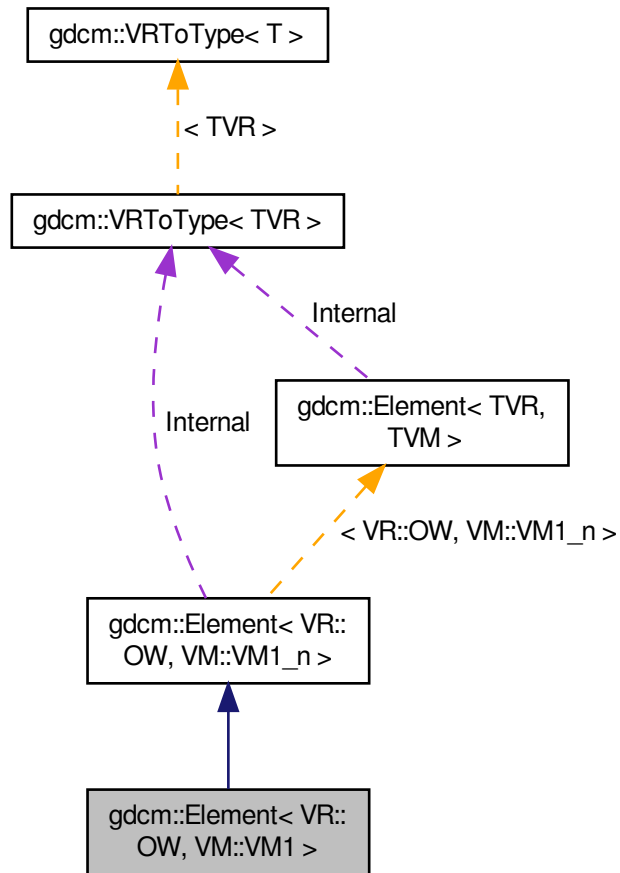
## 27.102 gdcM::Element< VR::OW, VM::VM1 > Class Template Reference

```
#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](#)

## 27.103 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>
```

### 27.103.1 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](#)

## 27.104 **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](#)

## 27.105 **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](#)

## 27.106 **gdcm::EncapsulatedDocument Class Reference**

[EncapsulatedDocument.](#)

```
#include <gdcmEncapsulatedDocument.h>
```

### Public Member Functions

- [EncapsulatedDocument \(\)](#)

### 27.106.1 Detailed Description

[EncapsulatedDocument.](#)

### 27.106.2 Constructor & Destructor Documentation

27.106.2.1 **gdcm::EncapsulatedDocument::EncapsulatedDocument ( )** `[inline]`

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

- [gdcmEncapsulatedDocument.h](#)

## 27.107 gdcm::EncodingImplementation< T > Class Template Reference

[EncodingImplementation](#).

```
#include <gdcmElement.h>
```

### 27.107.1 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](#)

## 27.108 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)

### 27.108.1 Member Function Documentation

27.108.1.1 template<typename T > static void gdcm::EncodingImplementation< VR::VRASCII >::Read ( T \* *data*, unsigned long *length*, std::istream & *is* ) [inline], [static]

27.108.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadComputeLength ( T * data, unsigned int & length, std::istream & _is ) [inline], [static]`

References `gdcm::backslash()`.

27.108.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::ReadNoSwap ( T * data, unsigned long length, std::istream & _is ) [inline], [static]`

27.108.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRASCII >::Write ( const T * data, unsigned long length, std::ostream & _os ) [inline], [static]`

27.108.1.5 `template<> void gdcm::EncodingImplementation< VR::VRASCII >::Write ( const float * data, unsigned long length, std::ostream & _os ) [inline]`

References `gdcm::to_string()`.

27.108.1.6 `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](#)

## 27.109 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)`

### 27.109.1 Member Function Documentation

27.109.1.1 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Read ( T * data, unsigned long length, std::istream & _is ) [inline], [static]`

References `gdcm::SwapperNoOp::SwapArray()`.



27.109.1.2 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadComputeLength ( T * data, unsigned int & length, std::istream & _is ) [inline], [static]`

27.109.1.3 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::ReadNoSwap ( T * data, unsigned long length, std::istream & _is ) [inline], [static]`

27.109.1.4 `template<typename T> static void gdcm::EncodingImplementation< VR::VRBINARY >::Write ( const T * data, unsigned long length, std::ostream & _os ) [inline], [static]`

References `gdcm::SwapperNoOp::Swap()`.

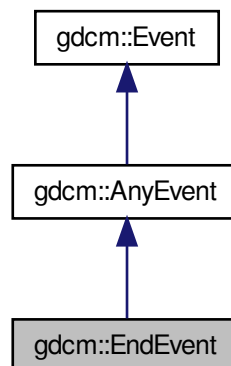
The documentation for this class was generated from the following file:

- [gdcmElement.h](#)

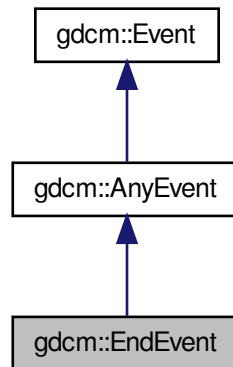
## 27.110 gdcm::EndEvent Class Reference

```
#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](#)

## 27.111 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](#) ()

#### 27.111.1 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](#).

## 27.111.2 Constructor & Destructor Documentation

### 27.111.2.1 gdcm::EnumeratedValues::EnumeratedValues ( ) [inline]

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

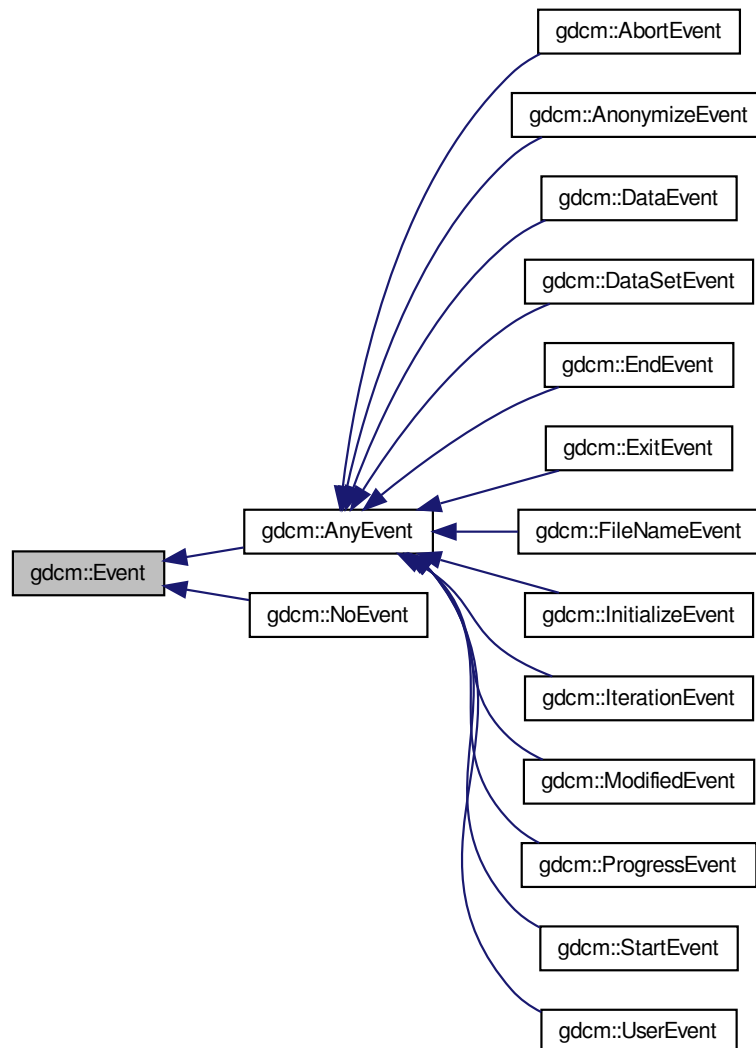
- [gdcmEnumeratedValues.h](#)

## 27.112 gdcm::Event Class Reference

superclass for callback/observer methods

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

Inheritance diagram for `gdcM::Event`:



## Public Member Functions

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

### 27.112.1 Detailed Description

superclass for callback/observer methods

See also

[Command Subject](#)

Examples:

[SimpleScanner.cxx](#).

### 27.112.2 Constructor & Destructor Documentation

27.112.2.1 `gdcmm::Event::Event ( )`

27.112.2.2 `gdcmm::Event::Event ( const Event & )`

27.112.2.3 `virtual gdcmm::Event::~~Event ( )` [virtual]

### 27.112.3 Member Function Documentation

27.112.3.1 `virtual bool gdcmm::Event::CheckEvent ( const Event * ) const` [pure virtual]

Check if given event matches or derives from this event.

27.112.3.2 `virtual const char* gdcmm::Event::GetEventName ( void ) const` [pure virtual]

Return the StringName associated with the event.

Implemented in [gdcmm::FileNameEvent](#), [gdcmm::ProgressEvent](#), [gdcmm::DataSetEvent](#), [gdcmm::AnonymizeEvent](#), and [gdcmm::DataEvent](#).

27.112.3.3 `virtual Event* gdcmm::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 [gdcmm::FileNameEvent](#), [gdcmm::ProgressEvent](#), [gdcmm::DataSetEvent](#), [gdcmm::AnonymizeEvent](#), and [gdcmm::DataEvent](#).

27.112.3.4 `virtual void gdcmm::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 `gdcmm::operator<<()`.

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

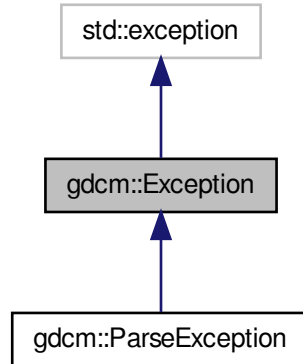
- [gdcmmEvent.h](#)

## 27.113 gdcM::Exception Class Reference

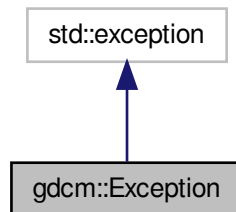
[Exception](#).

```
#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*

### 27.113.1 Detailed Description

[Exception](#).

Standard exception handling object.

#### Note

Its copy-constructor and assignment operator are generated by the compiler.

### 27.113.2 Constructor & Destructor Documentation

**27.113.2.1** `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.

**27.113.2.2** `virtual gdcm::Exception::~~Exception ( ) throw ) [inline],[virtual]`

### 27.113.3 Member Function Documentation

**27.113.3.1** `const char* gdcm::Exception::GetDescription ( ) const [inline]`

Return the Description.

Referenced by `gdcm::SequenceOfItems::Read()`.

**27.113.3.2** `const char* gdcm::Exception::what ( ) const throw ) [inline]`

*what implementation*

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

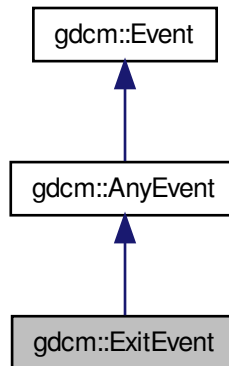
The documentation for this class was generated from the following file:

- [gdcmException.h](#)

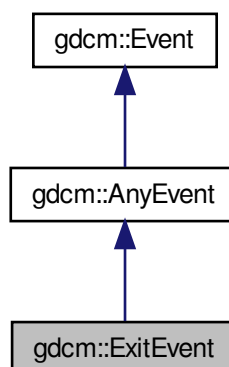
## 27.114 gdcm::ExitEvent Class Reference

```
#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](#)

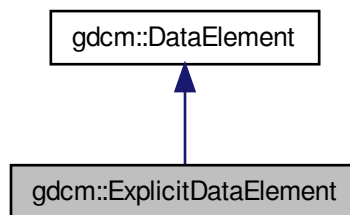


## 27.115 gdcM::ExplicitDataElement Class Reference

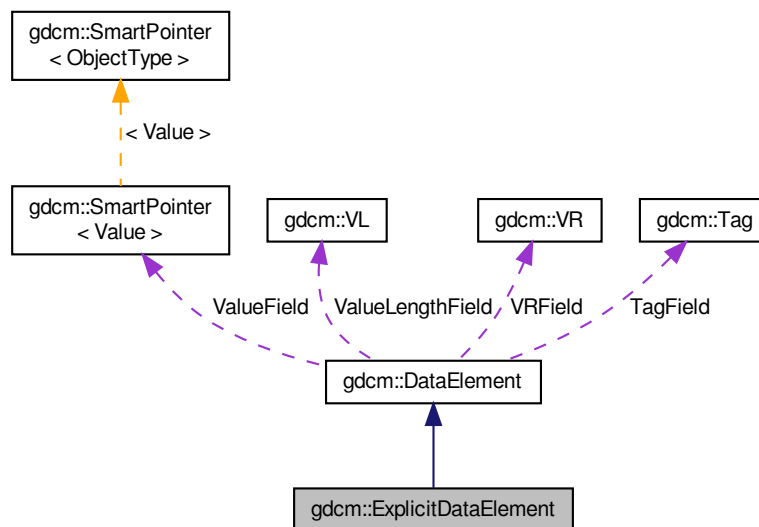
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#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

### 27.115.1 Detailed Description

Class to read/write a [DataElement](#) as Explicit Data [Element](#).

#### Note

bla

### 27.115.2 Member Function Documentation

27.115.2.1 `VL gdcmm::ExplicitDataElement::GetLength ( ) const`

27.115.2.2 `template<typename TSwap > std::istream& gdcmm::ExplicitDataElement::Read ( std::istream & is )`

27.115.2.3 `template<typename TSwap > std::istream& gdcmm::ExplicitDataElement::ReadPreValue ( std::istream & is )`

27.115.2.4 `template<typename TSwap > std::istream& gdcmm::ExplicitDataElement::ReadValue ( std::istream & is, bool readvalues = true )`

27.115.2.5 `template<typename TSwap > std::istream& gdcmm::ExplicitDataElement::ReadWithLength ( std::istream & is, VL & length )`

27.115.2.6 `template<typename TSwap > const std::ostream& gdcmm::ExplicitDataElement::Write ( std::ostream & os ) const`

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

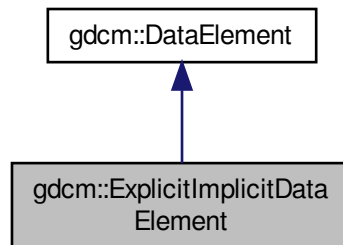
- [gdcmmExplicitDataElement.h](#)

## 27.116 gdcmm::ExplicitImplicitDataElement Class Reference

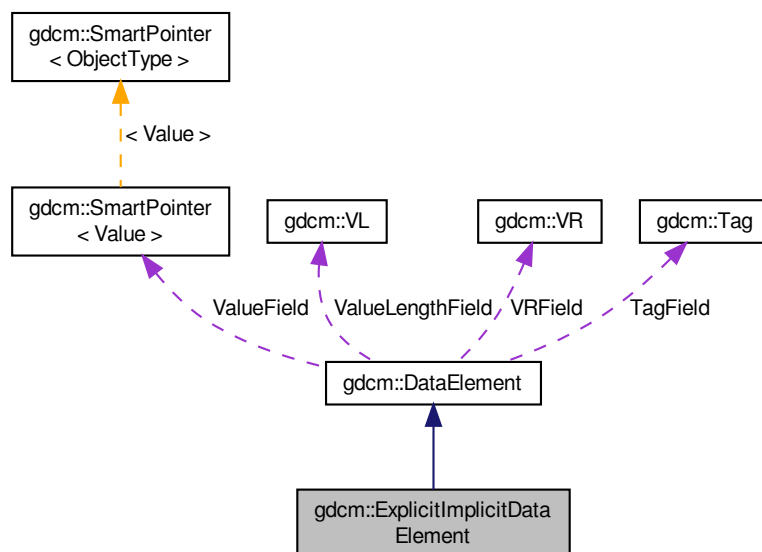
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#).

```
#include <gdcmmExplicitImplicitDataElement.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

### 27.116.1 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.

### 27.116.2 Member Function Documentation

27.116.2.1 `VL gdcmm::ExplicitImplicitDataElement::GetLength ( ) const`

27.116.2.2 `template<typename TSwap > std::istream& gdcmm::ExplicitImplicitDataElement::Read ( std::istream & is )`

27.116.2.3 `template<typename TSwap > std::istream& gdcmm::ExplicitImplicitDataElement::ReadPreValue ( std::istream & is )`

27.116.2.4 `template<typename TSwap > std::istream& gdcmm::ExplicitImplicitDataElement::ReadValue ( std::istream & is, bool readvalues = true )`

27.116.2.5 `template<typename TSwap > std::istream& gdcmm::ExplicitImplicitDataElement::ReadWithLength ( std::istream & is, VL & length ) [inline]`

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

- [gdcmmExplicitImplicitDataElement.h](#)

## 27.117 gdcmm::Fiducials Class Reference

[Fiducials](#).

```
#include <gdcmmFiducials.h>
```

### Public Member Functions

- [Fiducials](#) ()

### 27.117.1 Detailed Description

[Fiducials](#).

### 27.117.2 Constructor & Destructor Documentation

#### 27.117.2.1 gdcm::Fiducials::Fiducials ( ) [inline]

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

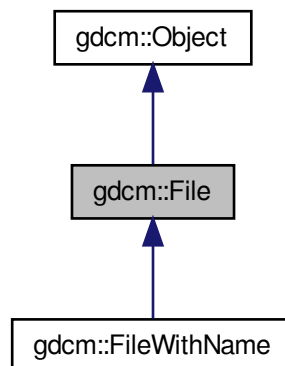
- [gdcmFiducials.h](#)

## 27.118 gdcm::File Class Reference

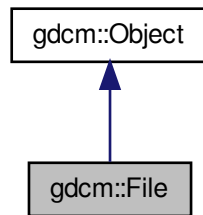
a DICOM [File](#) See PS 3.10 [File](#): A [File](#) is an ordered string of zero or more bytes, where the first byte is at the beginning of the file and the last byte at the end of the [File](#). Files are identified by a unique [File](#) ID and may be written, read and/or deleted.

```
#include <gdcmFile.h>
```

Inheritance diagram for gdcm::File:



Collaboration diagram for `gdcm::File`:



## Public Member Functions

- [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

### 27.118.1 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:

[ChangeSequenceUltrasound.cxx](#), [CreateJPIPDataSet.cxx](#), [DiffFile.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractImageRegion.cs](#), [ExtractImageRegionWithLUT.cs](#), [Extracting\\_All\\_Resolution.cxx](#), [ExtractOneFrame.cs](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FixBrokenJ2K.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MpegVideoInfo.cs](#), [NewSequence.cs](#), [PatchFile.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadGEMSSDO.cxx](#), [SimplePrintPatientName.cs](#), and [StreamImageReaderTest.cxx](#).

## 27.118.2 Constructor & Destructor Documentation

27.118.2.1 `gdcm::File ( ) [inline]`

## 27.118.3 Member Function Documentation

27.118.3.1 `const DataSet& gdcm::File::GetDataSet ( ) const [inline]`

Get Data Set.

Examples:

[ChangeSequenceUltrasound.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.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](#), and [StreamImageReaderTest.cxx](#).

27.118.3.2 `DataSet& gdcm::File::GetDataSet ( ) [inline]`

Get Data Set.

27.118.3.3 `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](#), [MergeTwoFiles.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

Referenced by `gdcm::operator<<()`.

27.118.3.4 **FileMetaInformation& gdcmm::File::GetHeader ( )** `[inline]`

Get [File](#) Meta Information.

27.118.3.5 **std::istream& gdcmm::File::Read ( std::istream & *is* )**

Read.

27.118.3.6 **void gdcmm::File::SetDataSet ( const DataSet & *ds* )** `[inline]`

Set Data Set.

27.118.3.7 **void gdcmm::File::SetHeader ( const FileMetaInformation & *fmi* )** `[inline]`

Set [File](#) Meta Information.

27.118.3.8 **std::ostream const& gdcmm::File::Write ( std::ostream & *os* ) const**

Write.

## 27.118.4 Friends And Related Function Documentation

27.118.4.1 **std::ostream& operator<< ( std::ostream & *os*, const File & *val* )** `[friend]`

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

- [gdcmmFile.h](#)

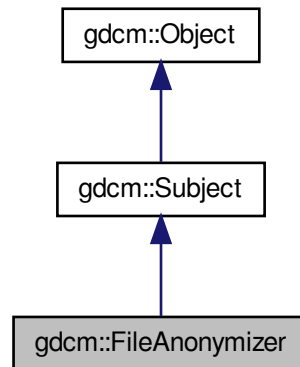
## 27.119 gdcmm::FileAnonymizer Class Reference

[FileAnonymizer](#).

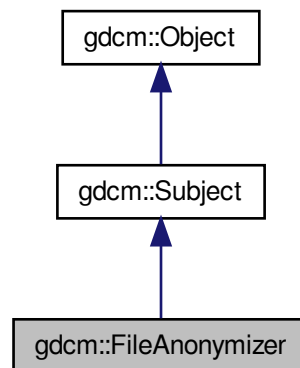
```
#include <gdcmmFileAnonymizer.h>
```



Inheritance diagram for gdcm::FileAnonymizer:



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

### 27.119.1 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 [gdcm::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.

#### Examples:

[FileAnonymize.cs](#), and [FileChangeTS.cs](#).

### 27.119.2 Constructor & Destructor Documentation

27.119.2.1 [gdcm::FileAnonymizer::FileAnonymizer](#) ( )

27.119.2.2 [gdcm::FileAnonymizer::~~FileAnonymizer](#) ( )

### 27.119.3 Member Function Documentation

27.119.3.1 void [gdcm::FileAnonymizer::Empty](#) ( [Tag](#) const & t )

Make [Tag](#) t empty  
Warning: does not handle SQ element

27.119.3.2 void [gdcm::FileAnonymizer::Remove](#) ( [Tag](#) const & t )

remove a tag (even a SQ can be removed)

27.119.3.3 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](#) !

27.119.3.4 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

27.119.3.5 void gdcm::FileAnonymizer::SetInputFileName ( const char \* *filename\_native* )

Set input filename.

27.119.3.6 void gdcm::FileAnonymizer::SetOutputFileName ( const char \* *filename\_native* )

Set output filename.

27.119.3.7 bool gdcm::FileAnonymizer::Write ( )

Write the output file.

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

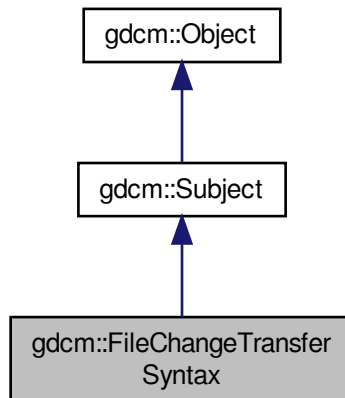
- [gdcmFileAnonymizer.h](#)

## 27.120 gdcm::FileChangeTransferSyntax Class Reference

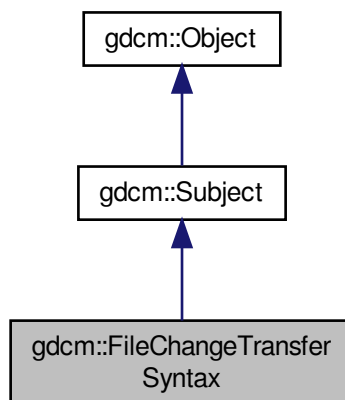
[FileChangeTransferSyntax](#).

```
#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

### 27.120.1 Detailed Description

[FileChangeTransferSyntax](#).

This class is a file-based replacement of the in-memory [ImageChangeTransferSyntax](#)

This class provide a file-based compression mecanism. It will take in an uncompressed DICOM image file (Pixel Data element). Then then produced as output a compressed DICOM file (Transfer Syntax will be updated).

Currently it supports the following transfer syntax:

- [JPEGLosslessProcess14\\_1](#)

### 27.120.2 Constructor & Destructor Documentation

27.120.2.1 `gdcm::FileChangeTransferSyntax::FileChangeTransferSyntax ( )`

27.120.2.2 `gdcm::FileChangeTransferSyntax::~~FileChangeTransferSyntax ( )`

### 27.120.3 Member Function Documentation

27.120.3.1 `bool gdcm::FileChangeTransferSyntax::Change ( )`

Change the transfer syntax.

27.120.3.2 `ImageCodec* gdcm::FileChangeTransferSyntax::GetCodec ( )`

Retrieve the actual codec (valid after calling SetTransferSyntax) Only advanced users should call this function.

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

for wrapped language: instantiate a reference counted object

27.120.3.4 void `gdcmm::FileChangeTransferSyntax::SetInputFileName ( const char * filename_native )`

Set input filename (raw DICOM)

27.120.3.5 void `gdcmm::FileChangeTransferSyntax::SetOutputFileName ( const char * filename_native )`

Set output filename (target compressed DICOM)

27.120.3.6 void `gdcmm::FileChangeTransferSyntax::SetTransferSyntax ( TransferSyntax const & ts )`

Specify the Target Transfer Syntax.

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

- [gdcmmFileChangeTransferSyntax.h](#)

## 27.121 gdcmm::FileDerivation Class Reference

[FileDerivation](#) class See PS 3.16 - 2008 For the list of Code [Value](#) that can be used for in Derivation Code Sequence.

```
#include <gdcmmFileDerivation.h>
```

### Public Member Functions

- [FileDerivation](#) ()
- [~FileDerivation](#) ()
- bool [AddReference](#) (const char \*referencedsopclassuid, const char \*referencedsopinstanceuid)
- bool [Derive](#) ()  
*Change.*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- 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](#) ()

### 27.121.1 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:

[GenFakelImage.cxx](#).

### 27.121.2 Constructor & Destructor Documentation

27.121.2.1 `gdcmm::FileDerivation::FileDerivation ( )`

27.121.2.2 `gdcmm::FileDerivation::~~FileDerivation ( )`

### 27.121.3 Member Function Documentation

27.121.3.1 `bool gdcmm::FileDerivation::AddDerivationDescription ( )` [protected]

27.121.3.2 `bool gdcmm::FileDerivation::AddPurposeOfReferenceCodeSequence ( DataSet & ds )` [protected]

27.121.3.3 `bool gdcmm::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:

[GenFakelImage.cxx](#).

27.121.3.4 `bool gdcmm::FileDerivation::AddSourceImageSequence ( )` [protected]

27.121.3.5 `bool gdcmm::FileDerivation::Derive ( )`

Change.

Examples:

[GenFakelImage.cxx](#).

27.121.3.6 **File& gdcmm::FileDerivation::GetFile ( )** [inline]

Examples:

[GenFakelImage.cxx](#).

27.121.3.7 **const File& gdcmm::FileDerivation::GetFile ( ) const** [inline]

27.121.3.8 **void gdcmm::FileDerivation::SetDerivationCodeSequenceCodeValue ( unsigned int *codevalue* )**

Specify the Derivation Code Sequence Code [Value](#). Eg 113040.

Examples:

[GenFakelImage.cxx](#).

27.121.3.9 **void gdcmm::FileDerivation::SetDerivationDescription ( const char \* *dd* )**

Specify the Derivation Description. Eg "lossy conversion".

27.121.3.10 **void gdcmm::FileDerivation::SetFile ( const File & *f* )** [inline]

Set/Get [File](#).

Examples:

[GenFakelImage.cxx](#).

27.121.3.11 **void gdcmm::FileDerivation::SetPurposeOfReferenceCodeSequenceCodeValue ( unsigned int *codevalue* )**

Specify the Purpose Of Reference Code [Value](#). Eg. 121320.

Examples:

[GenFakelImage.cxx](#).

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

- [gdcmmFileDerivation.h](#)

## 27.122 gdcmm::FileExplicitFilter Class Reference

[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.

```
#include <gdcmmFileExplicitFilter.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)

### 27.122.1 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](#).

### 27.122.2 Constructor & Destructor Documentation

27.122.2.1 `gdcm::FileExplicitFilter::FileExplicitFilter ( )` `[inline]`

27.122.2.2 `gdcm::FileExplicitFilter::~~FileExplicitFilter ( )` `[inline]`

### 27.122.3 Member Function Documentation

27.122.3.1 `bool gdcmm::FileExplicitFilter::Change ( )`

Set FMI Transfer Syntax.

Change

Examples:

[GenAllVR.cxx](#), and [LargeVRDSEExplicit.cxx](#).

27.122.3.2 `bool gdcmm::FileExplicitFilter::ChangeFMI ( )` [protected]

27.122.3.3 `File& gdcmm::FileExplicitFilter::GetFile ( )` [inline]

27.122.3.4 `bool gdcmm::FileExplicitFilter::ProcessDataSet ( DataSet & ds, Dicts const & dicts )` [protected]

27.122.3.5 `void gdcmm::FileExplicitFilter::SetChangePrivateTags ( bool b )` [inline]

Decide whether or not to [VR](#)ify private tags.

27.122.3.6 `void gdcmm::FileExplicitFilter::SetFile ( const File & f )` [inline]

Set/Get [File](#).

Examples:

[GenAllVR.cxx](#), and [LargeVRDSEExplicit.cxx](#).

27.122.3.7 `void gdcmm::FileExplicitFilter::SetRecomputeItemLength ( bool b )`

By default set Sequence & [Item](#) length to Undefined to avoid recomputing length:

27.122.3.8 `void gdcmm::FileExplicitFilter::SetRecomputeSequenceLength ( bool b )`

27.122.3.9 `void gdcmm::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:

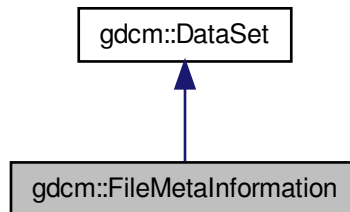
- [gdcmmFileExplicitFilter.h](#)

## 27.123 gdcmm::FileMetaInformation Class Reference

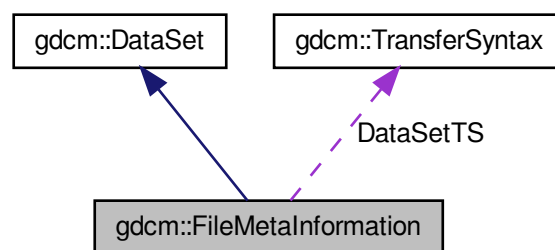
Class to represent a [File](#) Meta Information.

```
#include <gdcmmFileMetaInformation.h>
```

Inheritance diagram for gdcm::FileMetaInformation:



Collaboration diagram for gdcm::FileMetaInformation:



## Public Member Functions

- [FileMetaInformation](#) ()
- [FileMetaInformation](#) ([FileMetaInformation](#) const &fmi)
- 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

### 27.123.1 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:

[ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReformatFile.cs](#), and [StandardizeFiles.cs](#).

### 27.123.2 Constructor & Destructor Documentation

27.123.2.1 `gdcm::FileMetaInformation::FileMetaInformation ( )` `[inline]`

27.123.2.2 `gdcm::FileMetaInformation::FileMetaInformation ( FileMetaInformation const & fmi )` `[inline]`

References [DataSetMS](#), [DataSetTS](#), and [MetaInformationTS](#).

### 27.123.3 Member Function Documentation

27.123.3.1 `static void gdcm::FileMetaInformation::AppendImplementationClassUID ( const char * imp )` `[static]`

27.123.3.2 `void gdcm::FileMetaInformation::ComputeDataSetMediaStorageSOPClass ( )` `[protected]`

27.123.3.3 `void gdcm::FileMetaInformation::ComputeDataSetTransferSyntax ( )` `[protected]`

27.123.3.4 `void gdcm::FileMetaInformation::Default ( )` `[protected]`

27.123.3.5 `void gdcm::FileMetaInformation::FillFromDataSet ( DataSet const & ds )`

Construct a [FileMetaInformation](#) from an already existing [DataSet](#):

27.123.3.6 `const TransferSyntax& gdcm::FileMetaInformation::GetDataSetTransferSyntax ( ) const` `[inline]`

Examples:

[GetJPEGSamplePrecision.cxx](#), and [MergeTwoFiles.cxx](#).

27.123.3.7 `static const char* gdcm::FileMetaInformation::GetFileMetaInformationVersion ( ) [static],[protected]`

27.123.3.8 `VL gdcm::FileMetaInformation::GetFullLength ( ) const [inline]`

References `gdcm::VL::GetLength()`.

27.123.3.9 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationClassUID ( ) [static],[protected]`

27.123.3.10 `static const char* gdcm::FileMetaInformation::GetGDCMImplementationVersionName ( ) [static],[protected]`

27.123.3.11 `static const char* gdcm::FileMetaInformation::GetGDCMSourceApplicationEntityTitle ( ) [static],[protected]`

27.123.3.12 `static const char* gdcm::FileMetaInformation::GetImplementationClassUID ( ) [static]`

27.123.3.13 `static const char* gdcm::FileMetaInformation::GetImplementationVersionName ( ) [static]`

27.123.3.14 `MediaStorage gdcm::FileMetaInformation::GetMediaStorage ( ) const`

27.123.3.15 `std::string gdcm::FileMetaInformation::GetMediaStorageAsString ( ) const`

27.123.3.16 `TransferSyntax::NegociatedType gdcm::FileMetaInformation::GetMetaInformationTS ( ) const [inline]`

27.123.3.17 `const Preamble& gdcm::FileMetaInformation::GetPreamble ( ) const [inline]`

Get [Preamble](#).

Referenced by `gdcm::operator<<()`.

27.123.3.18 `Preamble& gdcm::FileMetaInformation::GetPreamble ( ) [inline]`

27.123.3.19 `static const char* gdcm::FileMetaInformation::GetSourceApplicationEntityTitle ( ) [static]`

27.123.3.20 `void gdcm::FileMetaInformation::Insert ( const DataElement & de ) [inline]`

References `gdcmErrorMacro`, `gdcm::Tag::GetGroup()`, and `gdcm::DataElement::GetTag()`.

27.123.3.21 `bool gdcm::FileMetaInformation::IsValid ( ) const [inline]`

27.123.3.22 `std::istream& gdcm::FileMetaInformation::Read ( std::istream & is )`

Read.

27.123.3.23 `std::istream& gdcm::FileMetaInformation::ReadCompat ( std::istream & is )`

27.123.3.24 `template<typename TSwap > std::istream& gdcm::FileMetaInformation::ReadCompatInternal ( std::istream & is ) [protected]`

27.123.3.25 void gdcM::FileMetaInformation::Replace ( const DataElement & *de* ) [inline]

Examples:

[LargeVRDSExplicit.cxx](#).

References gdcM::DataElement::GetTag().

27.123.3.26 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](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.123.3.27 static void gdcM::FileMetaInformation::SetImplementationClassUID ( const char \* *imp* ) [static]

Override the GDCM default values:

27.123.3.28 static void gdcM::FileMetaInformation::SetImplementationVersionName ( const char \* *version* ) [static]

27.123.3.29 void gdcM::FileMetaInformation::SetPreamble ( const Preamble & *p* ) [inline]

27.123.3.30 static void gdcM::FileMetaInformation::SetSourceApplicationEntityTitle ( const char \* *title* ) [static]

Examples:

[FixJAIBugJPEGLS.cxx](#).

27.123.3.31 std::ostream& gdcM::FileMetaInformation::Write ( std::ostream & *os* ) const

Write.

## 27.123.4 Friends And Related Function Documentation

27.123.4.1 std::ostream& operator<< ( std::ostream & *\_os*, const FileMetaInformation & *\_val* ) [friend]

## 27.123.5 Member Data Documentation

27.123.5.1 MediaStorage::MSType gdcM::FileMetaInformation::DataSetMS [protected]

Referenced by FileMetaInformation().

27.123.5.2 TransferSyntax gdcM::FileMetaInformation::DataSetTS [protected]

Referenced by FileMetaInformation().

### 27.123.5.3 TransferSyntax::NegociatedType gdcM::FileMetaInformation::MetaInformationTS [protected]

Referenced by FileMetaInformation().

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

- [gdcMFileMetaInformation.h](#)

## 27.124 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 foward slash (UNIX style) to windows style slash.*

### Static Public Member Functions

- static const char \* [Join](#) (const char \*path, const char \*filename)

### 27.124.1 Detailed Description

Class to manipulate file name's.

#### Note

OS independant representation of a filename (to query path, name and extension from a filename)

#### Examples:

[ClinicalTrialIdentificationWorkflow.cs.](#)



## 27.124.2 Constructor & Destructor Documentation

27.124.2.1 `gdcmm::Filename::Filename ( const char * filename = " " ) [inline]`

## 27.124.3 Member Function Documentation

27.124.3.1 `bool gdcmm::Filename::EndWith ( const char ending[ ] ) const`

Does the filename ends with a particular string ?

27.124.3.2 `const char* gdcmm::Filename::GetExtension ( )`

return only the extension part of a filename

27.124.3.3 `const char* gdcmm::Filename::GetFileName ( ) const [inline]`

Return the full filename.

27.124.3.4 `const char* gdcmm::Filename::GetName ( )`

return only the name part of a filename

27.124.3.5 `const char* gdcmm::Filename::GetPath ( )`

Return only the path component of a filename.

27.124.3.6 `bool gdcmm::Filename::IsEmpty ( ) const [inline]`

return whether the filename is empty

27.124.3.7 `bool gdcmm::Filename::IsIdentical ( Filename const & fn ) const`

27.124.3.8 `static const char* gdcmm::Filename::Join ( const char * path, const char * filename ) [static]`

Join two paths NOT THREAD SAFE

27.124.3.9 `gdcmm::Filename::operator const char * ( ) const [inline]`

Simple operator to allow `Filename myfilename( "...")`; `const char * s = myfilename`;

27.124.3.10 `const char* gdcmm::Filename::ToUnixSlashes ( )`

Convert backslash (windows style) to UNIX style slash.

27.124.3.11 `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:

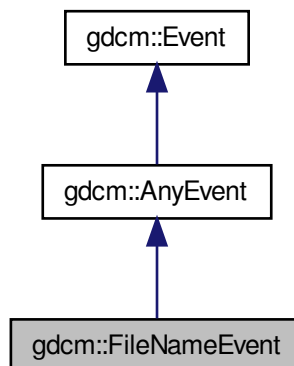
- [gdcmFilename.h](#)

## 27.125 gdcm::FileNameEvent Class Reference

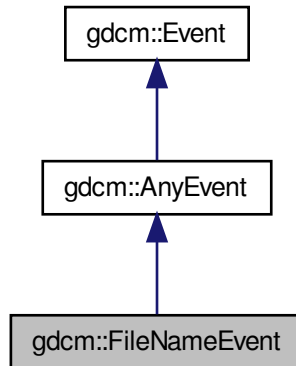
[FileNameEvent](#) Special type of event triggered during processing of [FileSet](#).

```
#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`)

### 27.125.1 Detailed Description

[FileNameEvent](#) Special type of event triggered during processing of [FileSet](#).

See also

[AnyEvent](#)

Examples:

[SimpleScanner.cxx](#).

## 27.125.2 Member Typedef Documentation

27.125.2.1 `typedef FileNameEvent gdcm::FileNameEvent::Self`

27.125.2.2 `typedef AnyEvent gdcm::FileNameEvent::Superclass`

## 27.125.3 Constructor & Destructor Documentation

27.125.3.1 `gdcm::FileNameEvent::FileNameEvent ( const char * s = " " ) [inline]`

27.125.3.2 `virtual gdcm::FileNameEvent::~~FileNameEvent ( ) [inline],[virtual]`

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

## 27.125.4 Member Function Documentation

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

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

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.125.4.3 `const char* gdcm::FileNameEvent::GetFileName ( ) const [inline]`

Examples:

[SimpleScanner.cxx](#).

27.125.4.4 `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](#).

27.125.4.5 `void gdcm::FileNameEvent::SetFileName ( const char * f ) [inline]`

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

- [gdcmFileNameEvent.h](#)

## 27.126 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.*

### 27.126.1 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](#).

### 27.126.2 Member Typedef Documentation

27.126.2.1 typedef std::vector<[FilenameType](#)> gdcm::FilenameGenerator::FileNamesType

27.126.2.2 typedef std::string gdcm::FilenameGenerator::FilenameType

27.126.2.3 typedef FileNamesType::size\_type gdcm::FilenameGenerator::SizeType

### 27.126.3 Constructor & Destructor Documentation

27.126.3.1 `gdcm::FilenameGenerator::FilenameGenerator ( )` `[inline]`

27.126.3.2 `gdcm::FilenameGenerator::~~FilenameGenerator ( )` `[inline]`

## 27.126.4 Member Function Documentation

27.126.4.1 `bool gdcm::FilenameGenerator::Generate ( )`

Generate (return success)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

27.126.4.2 `const char* gdcm::FilenameGenerator::GetFilename ( SizeType n ) const`

Get a particular filename (call after Generate)

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

27.126.4.3 `FilenameType const& gdcm::FilenameGenerator::GetFilenames ( ) const` `[inline]`

Return all filenames.

27.126.4.4 `SizeType gdcm::FilenameGenerator::GetNumberOfFilenames ( ) const`

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

27.126.4.5 `const char* gdcm::FilenameGenerator::GetPattern ( ) const` `[inline]`

27.126.4.6 `const char* gdcm::FilenameGenerator::GetPrefix ( ) const` `[inline]`

27.126.4.7 `void gdcm::FilenameGenerator::SetNumberOfFilenames ( SizeType nfiles )`

Set/Get the number of filenames to generate.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

27.126.4.8 `void gdcm::FilenameGenerator::SetPattern ( const char * pattern )` `[inline]`

Set/Get pattern.

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).

27.126.4.9 void gdcm::FilenameGenerator::SetPrefix ( const char \* *prefix* ) [inline]

Set/Get prefix.

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

- [gdcmFilenameGenerator.h](#)

## 27.127 gdcm::FileSet Class Reference

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.

```
#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)

### 27.127.1 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.

### 27.127.2 Member Typedef Documentation

27.127.2.1 typedef std::vector<[FileType](#)> gdcm::FileSet::FilesType

27.127.2.2 typedef std::string gdcm::FileSet::FileType

### 27.127.3 Constructor & Destructor Documentation

27.127.3.1 gdcm::FileSet::FileSet ( ) [inline]

### 27.127.4 Member Function Documentation

27.127.4.1 `void gdcM::FileSet::AddFile ( File const & ) [inline]`

**Deprecated** . Does nothing

27.127.4.2 `bool gdcM::FileSet::AddFile ( const char * filename )`

Add a file 'filename' to the list of files. Return true on success, false in case filename could not be found on system.

27.127.4.3 `FileType const& gdcM::FileSet::GetFiles ( ) const [inline]`

27.127.4.4 `void gdcM::FileSet::SetFiles ( FileType const & files )`

### 27.127.5 Friends And Related Function Documentation

27.127.5.1 `std::ostream& operator<< ( std::ostream & _os, const FileSet & d ) [friend]`

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

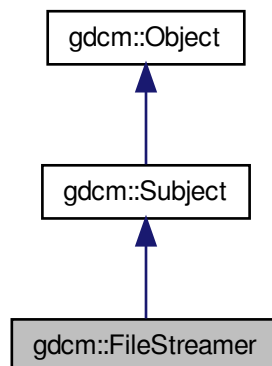
- [gdcMFileSet.h](#)

## 27.128 gdcM::FileStreamer Class Reference

**FileStreamer** This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

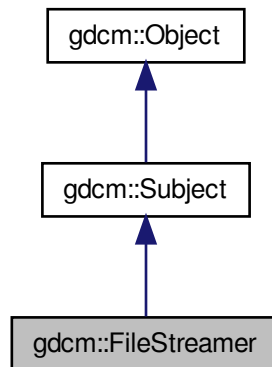
```
#include <gdcMFileStreamer.h>
```

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

### 27.128.1 Detailed Description

**FileStreamer** This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.

This class support two mode of operation:

1. Creating a single [DataElement](#) by appending chunk after chunk of data.
2. Creating a set of [DataElement](#) within the same group, using a private creator for start. New [DataElement](#) are added any time the user defined maximum size for data element is reached.

#### Warning

any existing [DataElement](#) is removed, pick carefully which [DataElement](#) to add.

#### Examples:

[FileChangeTS.cs](#), and [FileStreaming.cs](#).

### 27.128.2 Constructor & Destructor Documentation

27.128.2.1 `gdcm::FileStreamer::FileStreamer ( )`

27.128.2.2 `gdcm::FileStreamer::~~FileStreamer ( )`

### 27.128.3 Member Function Documentation

27.128.3.1 `bool gdcm::FileStreamer::AppendToDataElement ( const Tag & t, const char * array, size_t len )`

Append to previously started [Tag](#) t.

27.128.3.2 `bool gdcm::FileStreamer::AppendToGroupDataElement ( const PrivateTag & pt, const char * array, size_t len )`

Append to previously started private creator.

27.128.3.3 `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

27.128.3.4 `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.

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

for wrapped language: instantiate a reference counted object

27.128.3.6 `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

27.128.3.7 `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

27.128.3.8 `void gdcm::FileStreamer::SetOutputFileName ( const char * filename_native )`

Set output filename (target file)

27.128.3.9 `void gdcm::FileStreamer::SetTemplateFileName ( const char * filename_native )`

Set input DICOM template filename.

27.128.3.10 `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.

27.128.3.11 `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 <sup>32</sup> ). 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.
----------------	--

27.128.3.12 `bool gdcm::FileStreamer::StopDataElement ( const Tag & t )`

Stop appending to tag t. This will compute the proper attribute length.

27.128.3.13 `bool gdcm::FileStreamer::StopGroupDataElement ( const PrivateTag & pt )`

Stop appending to private creator.

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

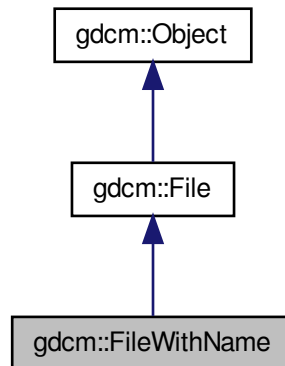
- [gdcmFileStreamer.h](#)

## 27.129 gdcm::FileWithName Class Reference

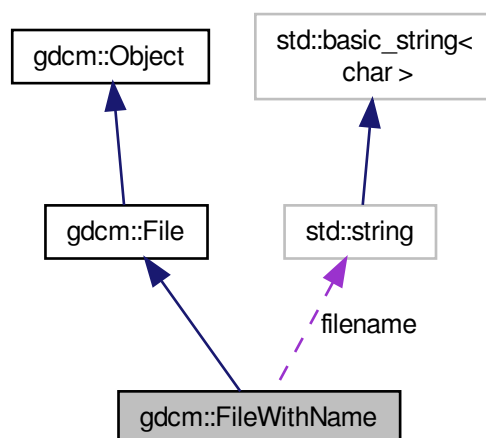
[FileWithName.](#)

```
#include <gdcmSerieHelper.h>
```

Inheritance diagram for gdcm::FileWithName:



Collaboration diagram for gdcm::FileWithName:



## Public Member Functions

- [FileWithName](#) ([File](#) &[f](#))

## Public Attributes

- `std::string` [filename](#)

## Additional Inherited Members

### 27.129.1 Detailed Description

[FileWithName](#).

Backward only class do not use in newer code

### 27.129.2 Constructor & Destructor Documentation

27.129.2.1 `gdcm::FileWithName::FileWithName ( File & f )` [[inline](#)]

### 27.129.3 Member Data Documentation

27.129.3.1 `std::string` `gdcm::FileWithName::filename`

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

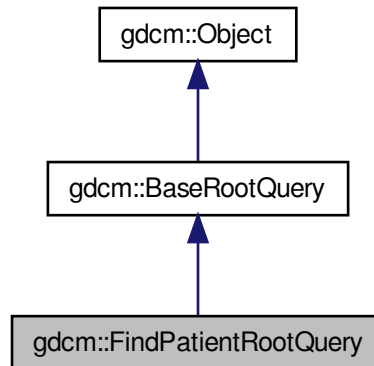
- [gdcmSerieHelper.h](#)

## 27.130 gdcm::FindPatientRootQuery Class Reference

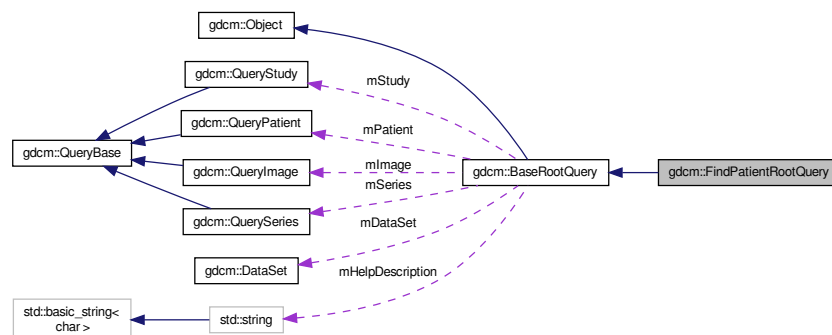
FindPatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

```
#include <gdcmFindPatientRootQuery.h>
```

Inheritance diagram for `gdcM::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

### 27.130.1 Detailed Description

PatientRootQuery contains: the class which will produce a dataset for c-find with patient root.

### 27.130.2 Constructor & Destructor Documentation

27.130.2.1 gdcmm::FindPatientRootQuery::FindPatientRootQuery ( )

### 27.130.3 Member Function Documentation

27.130.3.1 **UIDs::TSName** gdcmm::FindPatientRootQuery::GetAbstractSyntaxUID ( ) const [virtual]

Implements [gdcmm::BaseRootQuery](#).

27.130.3.2 **std::vector<Tag>** gdcmm::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 [gdcmm::BaseRootQuery](#).

27.130.3.3 **void** gdcmm::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 dcm4che

Implements [gdcmm::BaseRootQuery](#).

27.130.3.4 **bool** gdcmm::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 [gdcmm::BaseRootQuery](#).

### 27.130.4 Friends And Related Function Documentation

27.130.4.1 **friend class** QueryFactory [friend]

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

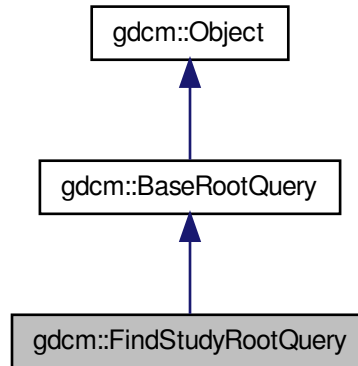
- [gdcmmFindPatientRootQuery.h](#)

## 27.131 gdcm::FindStudyRootQuery Class Reference

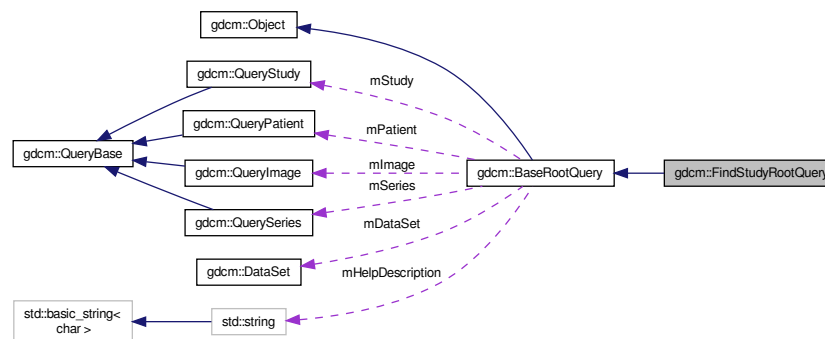
[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

```
#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

### 27.131.1 Detailed Description

[FindStudyRootQuery](#) contains: the class which will produce a dataset for C-FIND with study root.

### 27.131.2 Constructor & Destructor Documentation

27.131.2.1 `gdcmm::FindStudyRootQuery::FindStudyRootQuery ( )`

### 27.131.3 Member Function Documentation

27.131.3.1 `UIDs::TSName gdcmm::FindStudyRootQuery::GetAbstractSyntaxUID ( ) const` `[virtual]`

Implements [gdcmm::BaseRootQuery](#).

27.131.3.2 `std::vector<Tag> gdcmm::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 [gdcmm::BaseRootQuery](#).

27.131.3.3 `void gdcmm::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 dcmTk

Implements [gdcmm::BaseRootQuery](#).

27.131.3.4 `bool gdcmm::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 [gdcmm::BaseRootQuery](#).

### 27.131.4 Friends And Related Function Documentation

27.131.4.1 `friend class QueryFactory` `[friend]`

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

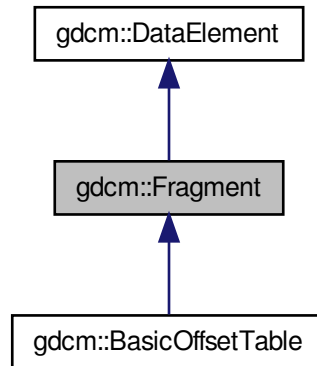
- [gdcmmFindStudyRootQuery.h](#)

## 27.132 gdcM::Fragment Class Reference

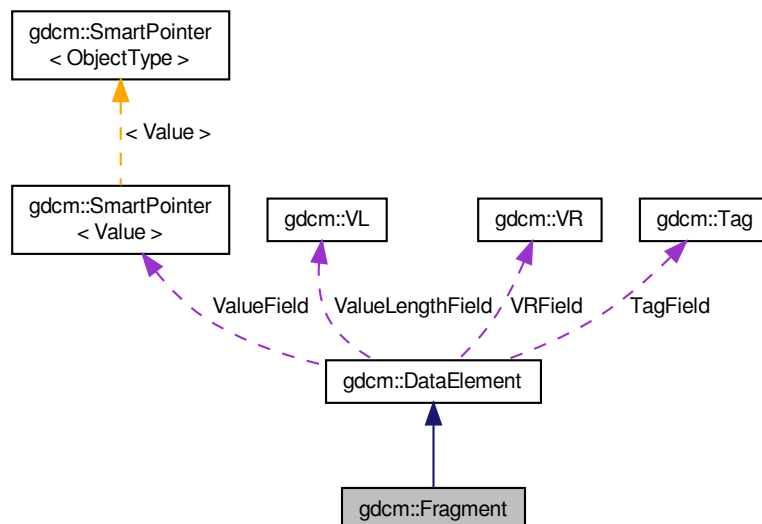
Class to represent a [Fragment](#).

```
#include <gdcMFragment.h>
```

Inheritance diagram for gdcM::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

### 27.132.1 Detailed Description

Class to represent a [Fragment](#).

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGLS.cxx](#).

### 27.132.2 Constructor & Destructor Documentation

27.132.2.1 `gdcm::Fragment::Fragment ( )` `[inline]`

### 27.132.3 Member Function Documentation

27.132.3.1 `VL gdcm::Fragment::ComputeLength ( )` const

27.132.3.2 `VL gdcm::Fragment::GetLength ( )` const

27.132.3.3 `template<typename TSwap > std::istream& gdcm::Fragment::Read ( std::istream & is )` `[inline]`

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

27.132.3.4 `template<typename TSwap > std::istream& gdcm::Fragment::ReadBacktrack ( std::istream & is )` `[inline]`

References `gdcmErrorMacro`, `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

Referenced by `gdcm::SequenceOfFragments::ReadValue()`.

27.132.3.5 `template<typename TSwap > std::istream& gdcm::Fragment::ReadPreValue ( std::istream & is ) [inline]`

27.132.3.6 `template<typename TSwap > std::istream& gdcm::Fragment::ReadValue ( std::istream & is ) [inline]`

References `gdcmWarningMacro`, and `gdcm::ParseException::SetLastElement()`.

27.132.3.7 `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()`.

## 27.132.4 Friends And Related Function Documentation

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

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

- [gdcmFragment.h](#)

## 27.133 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)`

### 27.133.1 Detailed Description

[Global](#).

#### Note

[Global](#) should be included in any translation unit that will use [Dict](#) or that implements the singleton pattern. It makes sure that the [Dict](#) singleton is created before and destroyed after all other singletons in GDCM.

#### Examples:

[BasicAnonymizer.cs](#), [ClinicalTrialIdentificationWorkflow.cs](#), [GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

### 27.133.2 Constructor & Destructor Documentation

27.133.2.1 `gdcmm::Global::Global ( )`

27.133.2.2 `gdcmm::Global::~~Global ( )`

### 27.133.3 Member Function Documentation

27.133.3.1 `bool gdcmm::Global::Append ( const char * path )`

Append path at the end of the path list

#### Warning

not thread safe !

27.133.3.2 `Defs const& gdcmm::Global::GetDefs ( ) const`

retrieve the default/internal (Part 3) You need to explicitly call LoadResourcesFiles before

#### Examples:

[GenerateStandardSOPClasses.cxx](#).

27.133.3.3 `Dicts const& gdcmm::Global::GetDicts ( ) const`

retrieve the default/internal dicts (Part 6) This dict is filled up at load time

#### Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), and [ReadAndPrintAttributes.cxx](#).

27.133.3.4 **Dicts& gdcmm::Global::GetDicts ( )**

27.133.3.5 **static Global& gdcmm::Global::GetInstance ( )** [static]

return the singleton instance

Examples:

[GenAllVR.cxx](#), [GenerateStandardSOPClasses.cxx](#), [GenFakeIdentifyFile.cxx](#), [MrProtocol.cxx](#), [PublicDict.cxx](#), [ReadAndPrintAttributes.cxx](#), and [TraverseModules.cxx](#).

27.133.3.6 **bool gdcmm::Global::LoadResourcesFiles ( )**

Load all internal XML files, resource path need to have been set before calling this member function (see [Append/↔](#) Prepend members func)

Warning

not thread safe !

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.133.3.7 **const char\* gdcmm::Global::Locate ( const char \* *resfile* ) const** [protected]

Locate a resource file.

27.133.3.8 **bool gdcmm::Global::Prepend ( const char \* *path* )**

Prepend path at the beginning of the path list

Warning

not thread safe !

## 27.133.4 Friends And Related Function Documentation

27.133.4.1 **std::ostream& operator<< ( std::ostream & *\_os*, const Global & *g* )** [friend]

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

- [gdcmmGlobal.h](#)

## 27.134 gdcmm::GroupDict Class Reference

Class to represent the mapping from group number to its abbreviation and name.

```
#include <gdcmmGroupDict.h>
```

## Public Types

- typedef std::vector< std::string > [GroupStringVector](#)

## Public Member Functions

- [GroupDict](#) ()
- [~GroupDict](#) ()
- 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)

### 27.134.1 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 ?

### 27.134.2 Member Typedef Documentation

27.134.2.1 typedef std::vector<std::string> gdcm::GroupDict::GroupStringVector

### 27.134.3 Constructor & Destructor Documentation

27.134.3.1 gdcm::GroupDict::GroupDict ( ) [\[inline\]](#)

27.134.3.2 gdcm::GroupDict::~~GroupDict ( ) [\[inline\]](#)

### 27.134.4 Member Function Documentation

27.134.4.1 void gdcm::GroupDict::Add ( std::string const & *abbreviation*, std::string const & *name* ) [\[protected\]](#)

27.134.4.2 std::string const& gdcm::GroupDict::GetAbbreviation ( uint16\_t *num* ) const

Referenced by gdcm::operator<<().

27.134.4.3 `std::string const& gdcm::GroupDict::GetName ( uint16_t num ) const`

Referenced by `gdcm::operator<<()`.

27.134.4.4 `void gdcm::GroupDict::Insert ( uint16_t num, std::string const & abbreviation, std::string const & name )`  
`[protected]`

27.134.4.5 `size_t gdcm::GroupDict::Size ( ) const` `[inline]`

Referenced by `gdcm::operator<<()`.

## 27.134.5 Friends And Related Function Documentation

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

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

- [gdcmGroupDict.h](#)

## 27.135 gdcm::IconImageFilter Class Reference

[IconImageFilter](#) This filter will extract icons from a [gdcm::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.

```
#include <gdcmIconImageFilter.h>
```

### Public Member Functions

- [IconImageFilter](#) ()
- [~IconImageFilter](#) ()
- bool [Extract](#) ()  
*Extract all Icon found in File.*
- [File](#) & [GetFile](#) ()
- const [File](#) & [GetFile](#) () const
- [IconImage](#) & [GetIconImage](#) (unsigned int i) const
- unsigned int [GetNumberOfIconImages](#) () const  
*Retrieve extract IconImage (need to call Extract first)*
- void [SetFile](#) (const [File](#) &f)  
*Set/Get File.*

### Protected Member Functions

- void [ExtractIconImages](#) ()
- void [ExtractVeprolIconImages](#) ()



### 27.135.1 Detailed Description

[IconImageFilter](#) This filter will extract icons from a [gdcm::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](#).

### 27.135.2 Constructor & Destructor Documentation

27.135.2.1 `gdcm::IconImageFilter::IconImageFilter ( )`

27.135.2.2 `gdcm::IconImageFilter::~~IconImageFilter ( )`

### 27.135.3 Member Function Documentation

27.135.3.1 `bool gdcm::IconImageFilter::Extract ( )`

Extract all Icon found in [File](#).

#### Examples:

[ExtractIconFromFile.cxx](#).

27.135.3.2 `void gdcm::IconImageFilter::ExtractIconImages ( )` [protected]

27.135.3.3 `void gdcm::IconImageFilter::ExtractVeprolconImages ( )` [protected]

27.135.3.4 **File& gdcm::IconImageFilter::GetFile ( )** [inline]

27.135.3.5 **const File& gdcm::IconImageFilter::GetFile ( ) const** [inline]

27.135.3.6 **IconImage& gdcm::IconImageFilter::GetIconImage ( unsigned int *i* ) const**

Examples:

[ExtractIconFromFile.cxx](#).

27.135.3.7 **unsigned int gdcm::IconImageFilter::GetNumberOfIconImages ( ) const**

Retrieve extract IconImage (need to call Extract first)

Examples:

[ExtractIconFromFile.cxx](#).

27.135.3.8 **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](#)

## 27.136 gdcm::IconImageGenerator Class Reference

[IconImageGenerator](#) This filter will generate a valid Icon from the Pixel Data element (an instance of [gdcm::Pixmap](#)). To generate a valid Icon, one is only allowed the following Photometric Interpretation:

```
#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.*

### 27.136.1 Detailed Description

[IconImageGenerator](#) This filter will generate a valid Icon from the Pixel Data element (an instance of [gdcm::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](#).

### 27.136.2 Constructor & Destructor Documentation

27.136.2.1 `gdcm::IconImageGenerator::IconImageGenerator ( )`

27.136.2.2 `gdcm::IconImageGenerator::~~IconImageGenerator ( )`

### 27.136.3 Member Function Documentation

27.136.3.1 `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](#).

27.136.3.2 `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

27.136.3.3 `bool gdcmlIconImageGenerator::Generate ( )`

Generate Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.136.3.4 `const IconImage& gdcmlIconImageGenerator::GetIconImage ( ) const` `[inline]`

Retrieve generated Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.136.3.5 `Pixmap& gdcmlIconImageGenerator::GetPixmap ( )` `[inline]`

27.136.3.6 `const Pixmap& gdcmlIconImageGenerator::GetPixmap ( ) const` `[inline]`

27.136.3.7 `void gdcmlIconImageGenerator::SetOutputDimensions ( const unsigned int dims[2] )`

Set Target dimension of output Icon.

Examples:

[ExtractIconFromFile.cxx](#).

27.136.3.8 `void gdcmlIconImageGenerator::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)`

27.136.3.9 `void gdcmlIconImageGenerator::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.

27.136.3.10 `void gdcmlIconImageGenerator::SetPixmap ( const Pixmap & p )` `[inline]`

Set/Get [File](#).

Examples:

[ExtractIconFromFile.cxx](#).

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

- [gdcmlIconImageGenerator.h](#)

## 27.137 gdcm::ignore\_char Struct Reference

```
#include <gdcmElement.h>
```

### Public Member Functions

- [ignore\\_char](#) (char c)

### Public Attributes

- char [m\\_char](#)

### 27.137.1 Constructor & Destructor Documentation

27.137.1.1 `gdcm::ignore_char::ignore_char ( char c )` `[inline]`

### 27.137.2 Member Data Documentation

27.137.2.1 `char gdcm::ignore_char::m_char`

Referenced by `gdcm::operator>>()`.

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

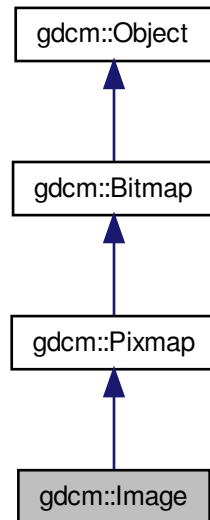
- [gdcmElement.h](#)

## 27.138 gdcm::Image Class Reference

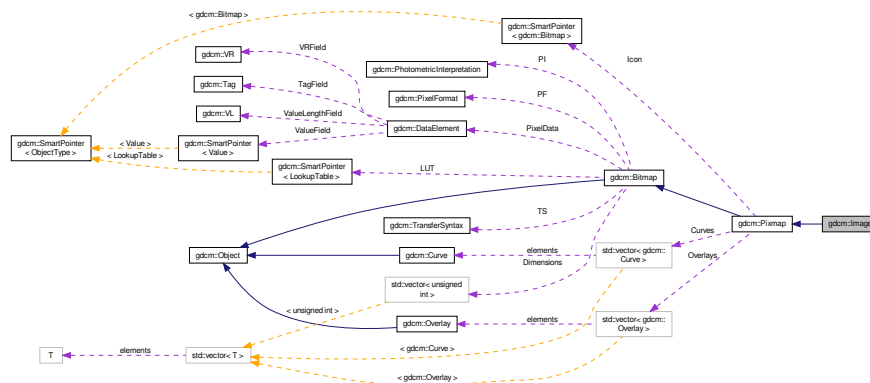
[Image](#) This is the container for an [Image](#) in the general sense. From this container you should be able to request information like:

```
#include <gdcmImage.h>
```

Inheritance diagram for `gdcm::Image`:



Collaboration diagram for `gdcm::Image`:



## Public Member Functions

- `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

### 27.138.1 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 [gdcm::Image](#) with [gdcm::JPEGImage](#) which would from the stream extract the header info and fill it to please [gdcm::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](#), [CompressLossyJPEG.cs](#), [ConvertToQImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [ExtractIconFromFile.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakeImage.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

## 27.138.2 Constructor & Destructor Documentation

27.138.2.1 `gdcm::Image::Image ( )` `[inline]`

27.138.2.2 `gdcm::Image::~~Image ( )` `[inline]`

## 27.138.3 Member Function Documentation

27.138.3.1 `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.

27.138.3.2 `double gdcm::Image::GetDirectionCosines ( unsigned int idx ) const`

27.138.3.3 `double gdcm::Image::GetIntercept ( ) const` `[inline]`

27.138.3.4 `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](#).

27.138.3.5 `double gdcm::Image::GetOrigin ( unsigned int idx ) const`

27.138.3.6 `double gdcm::Image::GetSlope ( ) const` `[inline]`

27.138.3.7 `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

27.138.3.8 `double gdcm::Image::GetSpacing ( unsigned int idx ) const`

27.138.3.9 `void gdcm::Image::Print ( std::ostream & os ) const` `[virtual]`

print

Reimplemented from [gdcm::Bitmap](#).

Examples:

[CompressImage.cxx](#), and [PatchFile.cxx](#).

27.138.3.10 `void gdcm::Image::SetDirectionCosines ( const float * dircos )`

27.138.3.11 `void gdcm::Image::SetDirectionCosines ( const double * dircos )`



27.138.3.12 void gdcm::Image::SetDirectionCosines ( unsigned int *idx*, double *dircos* )

27.138.3.13 void gdcm::Image::SetIntercept ( double *intercept* ) [inline]

intercept

27.138.3.14 void gdcm::Image::SetOrigin ( const float \* *ori* )

27.138.3.15 void gdcm::Image::SetOrigin ( const double \* *ori* )

27.138.3.16 void gdcm::Image::SetOrigin ( unsigned int *idx*, double *ori* )

27.138.3.17 void gdcm::Image::SetSlope ( double *slope* ) [inline]

slope

27.138.3.18 void gdcm::Image::SetSpacing ( const double \* *spacing* )

Examples:

[csa2img.cxx](#), and [iU22tomultisc.cxx](#).

27.138.3.19 void gdcm::Image::SetSpacing ( unsigned int *idx*, double *spacing* )

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

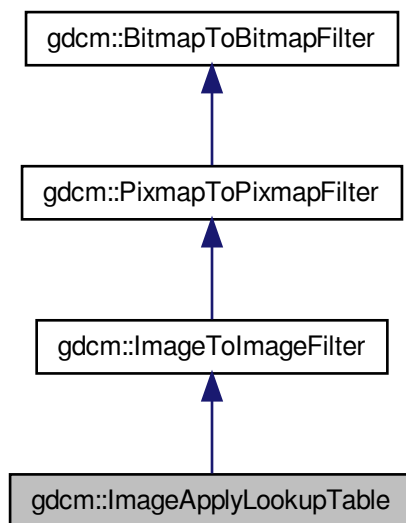
- [gdcmImage.h](#)

## 27.139 gdcm::ImageApplyLookupTable Class Reference

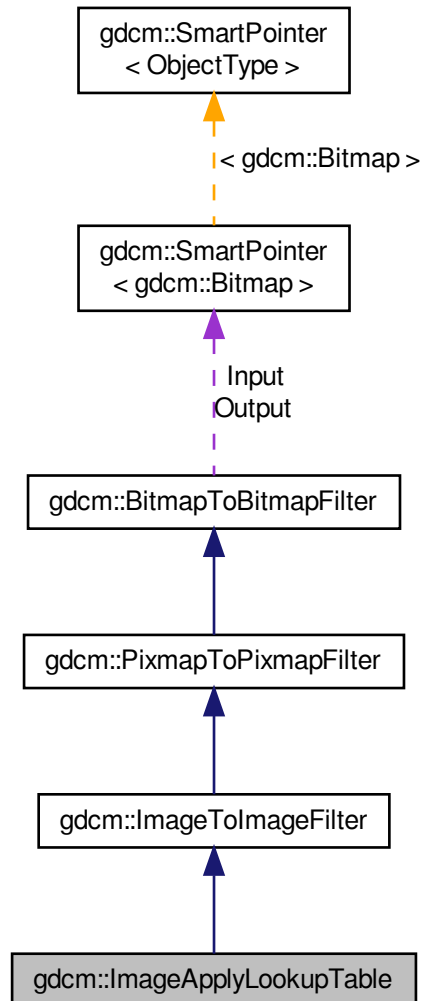
[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation=RGB](#) image.

```
#include <gdcmImageApplyLookupTable.h>
```

Inheritance diagram for `gdcm::ImageApplyLookupTable`:



Collaboration diagram for gdcm::ImageApplyLookupTable:



### Public Member Functions

- [ImageApplyLookupTable](#) ()
- [~ImageApplyLookupTable](#) ()
- `bool` [Apply](#) ()

*Apply.*

### Additional Inherited Members

### 27.139.1 Detailed Description

[ImageApplyLookupTable](#) class It applies the LUT the PixelData (only PALETTE\_COLOR images) Output will be a [PhotometricInterpretation](#)=RGB image.

### 27.139.2 Constructor & Destructor Documentation

27.139.2.1 `gdcm::ImageApplyLookupTable::ImageApplyLookupTable ( )` `[inline]`

27.139.2.2 `gdcm::ImageApplyLookupTable::~~ImageApplyLookupTable ( )` `[inline]`

### 27.139.3 Member Function Documentation

27.139.3.1 `bool gdcm::ImageApplyLookupTable::Apply ( )`

Apply.

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

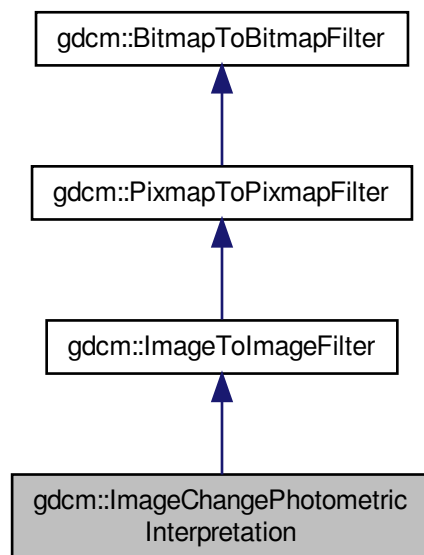
- [gdcmImageApplyLookupTable.h](#)

## 27.140 gdcm::ImageChangePhotometricInterpretation Class Reference

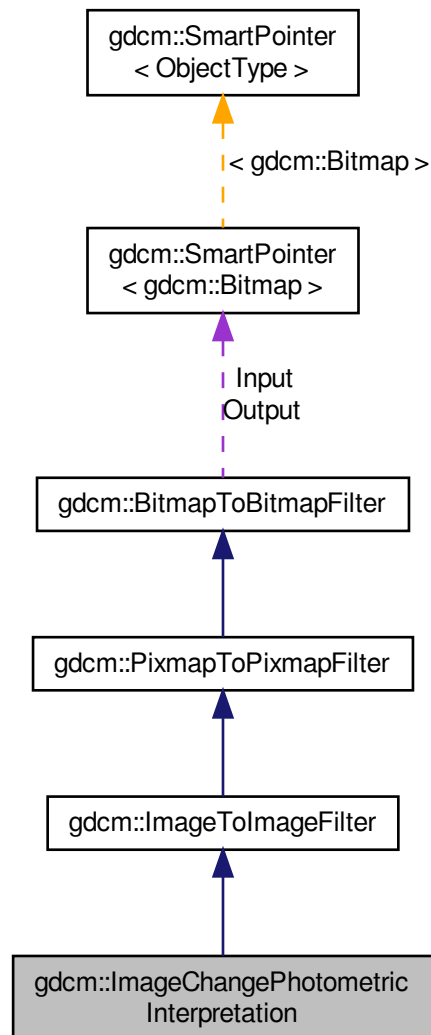
[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

```
#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])  
*colorspace conversion (based on CCIR Recommendation 601-2)*
- template<typename T >  
static void [YBR2RGB](#) (T rgb[3], const T ybr[3])

## Protected Member Functions

- bool [ChangeMonochrome](#) ()

## Additional Inherited Members

### 27.140.1 Detailed Description

[ImageChangePhotometricInterpretation](#) class Class to change the Photometric Interpretation of an input DICOM.

### 27.140.2 Constructor & Destructor Documentation

27.140.2.1 `gdcm::ImageChangePhotometricInterpretation::ImageChangePhotometricInterpretation ( )` `[inline]`

27.140.2.2 `gdcm::ImageChangePhotometricInterpretation::~~ImageChangePhotometricInterpretation ( )` `[inline]`

### 27.140.3 Member Function Documentation

27.140.3.1 `bool gdcm::ImageChangePhotometricInterpretation::Change ( )`

Change.

27.140.3.2 `bool gdcm::ImageChangePhotometricInterpretation::ChangeMonochrome ( )` `[protected]`

27.140.3.3 `const PhotometricInterpretation& gdcm::ImageChangePhotometricInterpretation::GetPhotometricInterpretation ( )`  
`const` `[inline]`

27.140.3.4 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::RGB2YBR ( T ybr[3], const T rgb[3] )`  
`[static]`

colorspace conversion (based on CCIR Recommendation 601-2)

27.140.3.5 `void gdcm::ImageChangePhotometricInterpretation::SetPhotometricInterpretation ( PhotometricInterpretation const & pi )` `[inline]`

Set/Get requested [PhotometricInterpretation](#).

27.140.3.6 `template<typename T > void gdcm::ImageChangePhotometricInterpretation::YBR2RGB ( T rgb[3], const T ybr[3] )`  
`[static]`

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

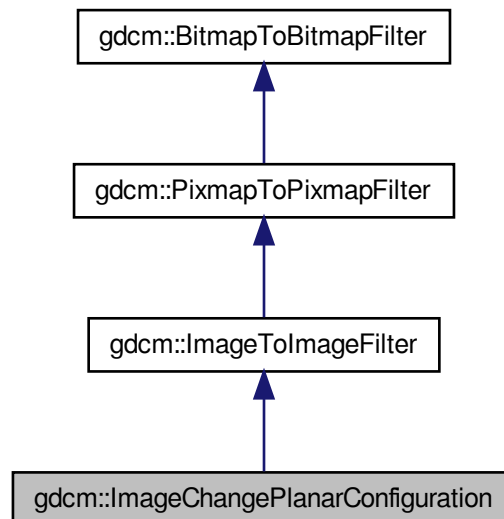
- [gdcmImageChangePhotometricInterpretation.h](#)

## 27.141 gdcm::ImageChangePlanarConfiguration Class Reference

[ImageChangePlanarConfiguration](#) class Class to change the Planar configuration of an input DICOM By default it will change into the more usual representation: PlanarConfiguration = 0.

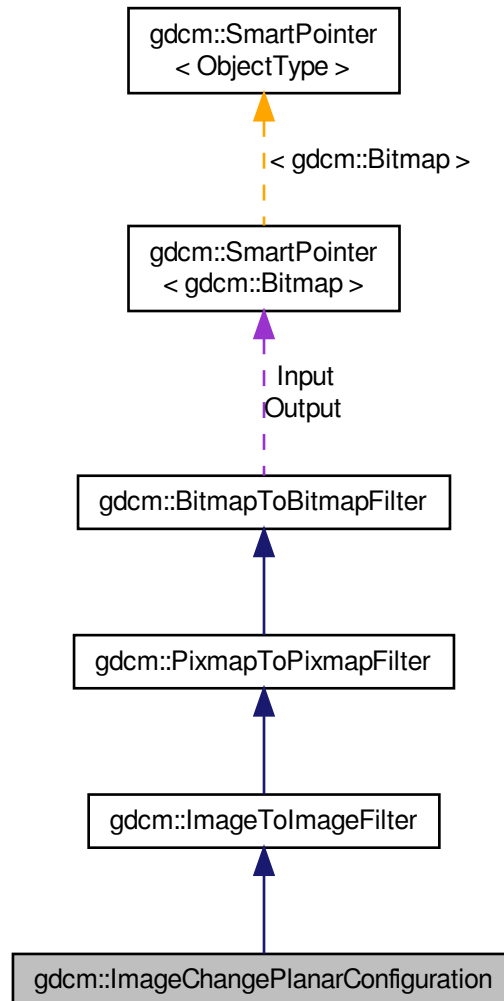
```
#include <gdcmImageChangePlanarConfiguration.h>
```

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

### 27.141.1 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.

### 27.141.2 Constructor & Destructor Documentation

27.141.2.1 `gdcm::ImageChangePlanarConfiguration::ImageChangePlanarConfiguration ( )` `[inline]`

27.141.2.2 `gdcm::ImageChangePlanarConfiguration::~~ImageChangePlanarConfiguration ( )` `[inline]`

### 27.141.3 Member Function Documentation

27.141.3.1 `bool gdcm::ImageChangePlanarConfiguration::Change ( )`

Change.

27.141.3.2 `unsigned int gdcm::ImageChangePlanarConfiguration::GetPlanarConfiguration ( ) const` `[inline]`

27.141.3.3 `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

27.141.3.4 `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

27.141.3.5 `void gdcm::ImageChangePlanarConfiguration::SetPlanarConfiguration ( unsigned int pc )` `[inline]`

Set/Get requested PlanarConfiguration.

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

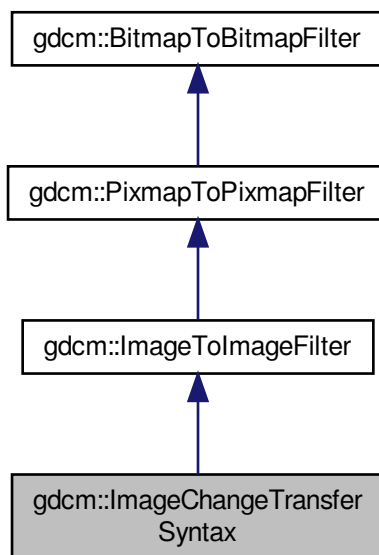
- [gdcmImageChangePlanarConfiguration.h](#)

## 27.142 gdcm::ImageChangeTransferSyntax Class Reference

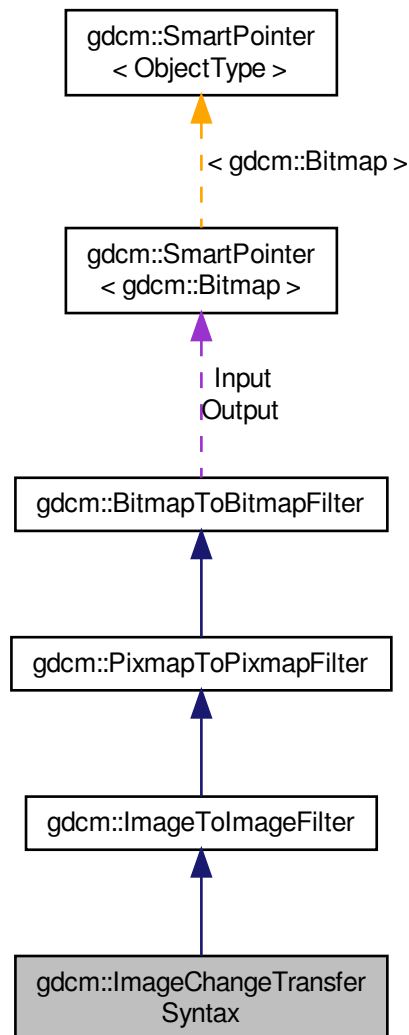
[ImageChangeTransferSyntax](#) class Class to change the transfer syntax of an input DICOM.

```
#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

#### 27.142.1 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](#).

#### 27.142.2 Constructor & Destructor Documentation

27.142.2.1 `gdcm::ImageChangeTransferSyntax::ImageChangeTransferSyntax ( )` `[inline]`

27.142.2.2 `gdcm::ImageChangeTransferSyntax::~ImageChangeTransferSyntax ( )` `[inline]`

#### 27.142.3 Member Function Documentation

27.142.3.1 `bool gdcm::ImageChangeTransferSyntax::Change ( )`

Change.

Examples:

[CompressImage.cxx](#).

27.142.3.2 `const TransferSyntax& gdcm::ImageChangeTransferSyntax::GetTransferSyntax ( ) const` `[inline]`

Get Transfer Syntax.

27.142.3.3 `void gdcmm::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

27.142.3.4 `void gdcmm::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

27.142.3.5 `void gdcmm::ImageChangeTransferSyntax::SetTransferSyntax ( const TransferSyntax & ts )` `[inline]`

Set target Transfer Syntax.

Examples:

[CompressImage.cxx](#).

27.142.3.6 `void gdcmm::ImageChangeTransferSyntax::SetUserCodec ( ImageCodec * ic )` `[inline]`

Allow user to specify exactly which codec to use. this is needed to specify special qualities or compression option.

Warning

if the codec '*ic*' is not compatible with the [TransferSyntax](#) requested, it will not be used. It is the user responsibility to check that `UserCodec->CanCode( TransferSyntax )`

27.142.3.7 `bool gdcmm::ImageChangeTransferSyntax::TryJPEG2000Codec ( const DataElement & pixelde, Bitmap const & input, Bitmap & output )` `[protected]`

27.142.3.8 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGCodec ( const DataElement & pixelde, Bitmap const & input, Bitmap & output )` `[protected]`

27.142.3.9 `bool gdcmm::ImageChangeTransferSyntax::TryJPEGLSCCodec ( const DataElement & pixelde, Bitmap const & input, Bitmap & output )` `[protected]`

27.142.3.10 `bool gdcmm::ImageChangeTransferSyntax::TryRAWCodec ( const DataElement & pixelde, Bitmap const & input, Bitmap & output )` `[protected]`

27.142.3.11 `bool gdcmm::ImageChangeTransferSyntax::TryRLECodec ( const DataElement & pixelde, Bitmap const & input, Bitmap & output )` `[protected]`

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

- [gdcmmImageChangeTransferSyntax.h](#)



- 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

- 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 [IsValid](#) ([PhotometricInterpretation](#) const &pi)

## 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 [ImageChangePhotometricInterpretation](#)

## 27.143.1 Detailed Description

[ImageCodec](#).

### Note

Main codec, this is a central place for all implementation

## 27.143.2 Member Typedef Documentation

27.143.2.1 `typedef SmartPointer<LookupTable> gdcm::ImageCodec::LUTPtr` `[protected]`

## 27.143.3 Constructor & Destructor Documentation

27.143.3.1 `gdcm::ImageCodec::ImageCodec ( )`

27.143.3.2 `gdcm::ImageCodec::~~ImageCodec ( )`

## 27.143.4 Member Function Documentation

27.143.4.1 `bool gdcm::ImageCodec::CanCode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG↔GLSCCodec](#), [gdcm::PNMCodec](#), [gdcm::PGXCodec](#), [gdcm::KAKADUCodec](#), and [gdcm::RAWCodec](#).

27.143.4.2 `bool gdcm::ImageCodec::CanDecode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

Reimplemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG2000Codec](#), [gdcm::JPEG↔GLSCCodec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), [gdcm::PGXCodec](#), and [gdcm::KAKADUCodec](#).

27.143.4.3 `virtual ImageCodec* gdcm::ImageCodec::Clone ( ) const` `[pure virtual]`

Implemented in [gdcm::JPEGCodec](#), [gdcm::RLECodec](#), [gdcm::JPEGGLSCCodec](#), [gdcm::PVRGCodec](#), [gdcm::JPEG↔G2000Codec](#), [gdcm::PNMCodec](#), [gdcm::RAWCodec](#), [gdcm::KAKADUCodec](#), and [gdcm::PGXCodec](#).

27.143.4.4 `bool gdcM::ImageCodec::Decode ( DataElement const & , DataElement & ) [virtual]`

Decode.

Reimplemented from [gdcM::Decoder](#).

Reimplemented in [gdcM::JPEGCodec](#), [gdcM::RLECodec](#), [gdcM::JPEGLSCodec](#), [gdcM::PVRGCodec](#), [gdcM::JPEG2000Codec](#), [gdcM::KAKADUCodec](#), and [gdcM::RAWCodec](#).

27.143.4.5 `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](#).

27.143.4.6 `bool gdcM::ImageCodec::DoByteSwap ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.7 `bool gdcM::ImageCodec::DoInvertMonochrome ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.8 `bool gdcM::ImageCodec::DoOverlayCleanup ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.9 `bool gdcM::ImageCodec::DoPaddedCompositePixelCode ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.10 `bool gdcM::ImageCodec::DoPlanarConfiguration ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.11 `bool gdcM::ImageCodec::DoSimpleCopy ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.12 `bool gdcM::ImageCodec::DoYBR ( std::istream & is_ , std::ostream & os ) [protected]`

27.143.4.13 `const unsigned int* gdcM::ImageCodec::GetDimensions ( ) const [inline]`

27.143.4.14 `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](#).

27.143.4.15 `bool gdcM::ImageCodec::GetLossyFlag ( ) const`

27.143.4.16 `const LookupTable& gdcM::ImageCodec::GetLUT ( ) const [inline]`

27.143.4.17 `bool gdcM::ImageCodec::GetNeedByteSwap ( ) const [inline]`

27.143.4.18 `unsigned int gdcM::ImageCodec::GetNumberOfDimensions ( ) const`

27.143.4.19 `const PhotometricInterpretation& gdcM::ImageCodec::GetPhotometricInterpretation ( ) const`

27.143.4.20 `PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) [inline]`

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.143.4.21 `const PixelFormat& gdcm::ImageCodec::GetPixelFormat ( ) const [inline]`

27.143.4.22 `unsigned int gdcm::ImageCodec::GetPlanarConfiguration ( ) const [inline]`

27.143.4.23 `bool gdcm::ImageCodec::IsLossy ( ) const`

27.143.4.24 `virtual bool gdcm::ImageCodec::IsValid ( PhotometricInterpretation const & pi ) [protected],  
[virtual]`

Reimplemented in [gdcm::JPEGCodec](#).

27.143.4.25 `void gdcm::ImageCodec::SetDimensions ( const unsigned int d[3] )`

Examples:

[ExtractIconFromFile.cxx](#).

27.143.4.26 `void gdcm::ImageCodec::SetDimensions ( const std::vector< unsigned int > & d )`

27.143.4.27 `void gdcm::ImageCodec::SetLossyFlag ( bool l )`

27.143.4.28 `void gdcm::ImageCodec::SetLUT ( LookupTable const & lut ) [inline]`

Examples:

[ExtractIconFromFile.cxx](#).

27.143.4.29 `void gdcm::ImageCodec::SetNeedByteSwap ( bool b ) [inline]`

27.143.4.30 `void gdcm::ImageCodec::SetNeedOverlayCleanup ( bool b ) [inline]`

27.143.4.31 `void gdcm::ImageCodec::SetNumberOfDimensions ( unsigned int dim )`

27.143.4.32 `void gdcm::ImageCodec::SetPhotometricInterpretation ( PhotometricInterpretation const & pi )`

Examples:

[ExtractIconFromFile.cxx](#).

27.143.4.33 `virtual void gdcm::ImageCodec::SetPixelFormat ( PixelFormat const & pf ) [inline], [virtual]`

Reimplemented in [gdcm::JPEGCodec](#).

Examples:

[ExtractIconFromFile.cxx](#).

27.143.4.34 void gdcmm::ImageCodec::SetPlanarConfiguration ( unsigned int *pc* ) [inline]

## 27.143.5 Friends And Related Function Documentation

27.143.5.1 friend class ImageChangePhotometricInterpretation [friend]

## 27.143.6 Member Data Documentation

27.143.6.1 unsigned int gdcmm::ImageCodec::Dimensions[3] [protected]

27.143.6.2 bool gdcmm::ImageCodec::LossyFlag [protected]

27.143.6.3 LUTPtr gdcmm::ImageCodec::LUT [protected]

27.143.6.4 bool gdcmm::ImageCodec::NeedByteSwap [protected]

27.143.6.5 bool gdcmm::ImageCodec::NeedOverlayCleanup [protected]

27.143.6.6 unsigned int gdcmm::ImageCodec::NumberOfDimensions [protected]

27.143.6.7 PixelFormat gdcmm::ImageCodec::PF [protected]

27.143.6.8 PhotometricInterpretation gdcmm::ImageCodec::PI [protected]

27.143.6.9 unsigned int gdcmm::ImageCodec::PlanarConfiguration [protected]

27.143.6.10 bool gdcmm::ImageCodec::RequestPaddedCompositePixelCode [protected]

27.143.6.11 bool gdcmm::ImageCodec::RequestPlanarConfiguration [protected]

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

- [gdcmmImageCodec.h](#)

## 27.144 gdcmm::ImageConverter Class Reference

[Image](#) Converter.

```
#include <gdcmmImageConverter.h>
```

### Public Member Functions

- [ImageConverter](#) ()
- [~ImageConverter](#) ()
- void [Convert](#) ()
- const [Image](#) & [GetOutput](#) () const
- void [SetInput](#) ([Image](#) const &input)

### 27.144.1 Detailed Description

[Image Converter](#).

#### Note

This is the class used to convert from on [gdcm::Image](#) to another This is typically used to convert let say YBR JPEG compressed [gdcm::Image](#) to a RAW RGB [gdcm::Image](#). So that the buffer can be directly pass to third party application. This filter is application level and not integrated directly in GDCM

### 27.144.2 Constructor & Destructor Documentation

27.144.2.1 `gdcm::ImageConverter::ImageConverter ( )`

27.144.2.2 `gdcm::ImageConverter::~~ImageConverter ( )`

### 27.144.3 Member Function Documentation

27.144.3.1 `void gdcm::ImageConverter::Convert ( )`

27.144.3.2 `const Image& gdcm::ImageConverter::GetOutput ( ) const`

27.144.3.3 `void gdcm::ImageConverter::SetInput ( Image const & input )`

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

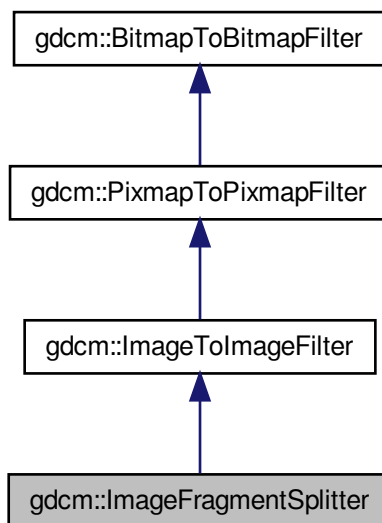
- [gdcmImageConverter.h](#)

## 27.145 gdcm::ImageFragmentSplitter Class Reference

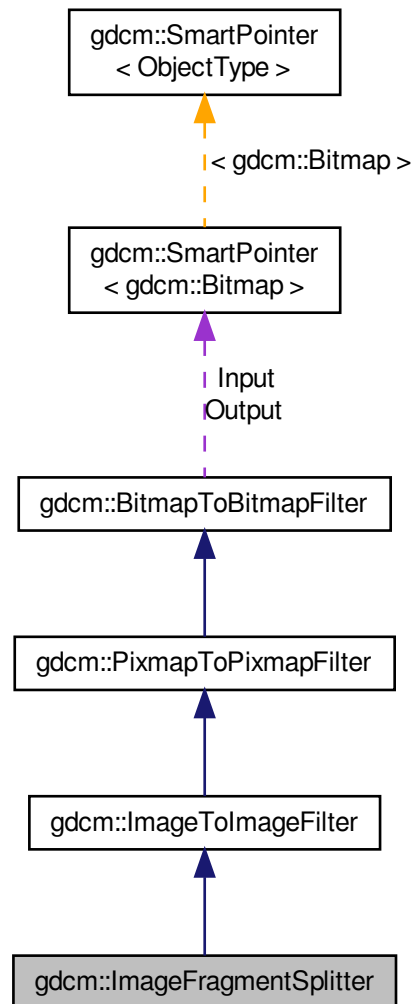
[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

```
#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

### 27.145.1 Detailed Description

[ImageFragmentSplitter](#) class For single frame image, DICOM standard allow splitting the frame into multiple fragments.

### 27.145.2 Constructor & Destructor Documentation

27.145.2.1 `gdcm::ImageFragmentSplitter::ImageFragmentSplitter ( )` [[inline](#)]

27.145.2.2 `gdcm::ImageFragmentSplitter::~~ImageFragmentSplitter ( )` [[inline](#)]

### 27.145.3 Member Function Documentation

27.145.3.1 `unsigned int gdcm::ImageFragmentSplitter::GetFragmentSizeMax ( ) const` [[inline](#)]

27.145.3.2 `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 !

27.145.3.3 `void gdcm::ImageFragmentSplitter::SetFragmentSizeMax ( unsigned int fragsize )`

FragmentSizeMax needs to be an even number.

27.145.3.4 `bool gdcm::ImageFragmentSplitter::Split ( )`

Split.

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

- [gdcmImageFragmentSplitter.h](#)

## 27.146 gdcm::ImageHelper Class Reference

[ImageHelper](#) (internal class, not intended for user level)

```
#include <gdcmImageHelper.h>
```

### Static Public Member Functions

- 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)
- 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 const [ByteValue](#) \* [GetPointerFromElement](#) ([Tag](#) const &tag, [File](#) const &f)  
*Moved from PixampReader to here. Generally used for photometric interpretation.*
- 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 [Image](#) &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 [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)

### 27.146.1 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.

### 27.146.2 Member Function Documentation

- 27.146.2.1 static bool gdcm::ImageHelper::ComputeSpacingFromImagePositionPatient ( const std::vector< double > & *imageposition*, std::vector< double > & *spacing* ) [static]

DO NOT USE.

27.146.2.2 `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](#).

27.146.2.3 `static bool gdcm::ImageHelper::GetDirectionCosinesFromDataSet ( DataSet const & ds, std::vector< double > & dircos ) [static]`

27.146.2.4 `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

27.146.2.5 `static bool gdcm::ImageHelper::GetForcePixelSpacing ( ) [static]`

27.146.2.6 `static bool gdcm::ImageHelper::GetForceRescaleInterceptSlope ( ) [static]`

27.146.2.7 `static SmartPointer<LookupTable> gdcm::ImageHelper::GetLUT ( File const & f ) [static]`

27.146.2.8 `static std::vector<double> gdcm::ImageHelper::GetOriginValue ( File const & f ) [static]`

Set/Get Origin (IPP) from/to a file.

27.146.2.9 `static PhotometricInterpretation gdcm::ImageHelper::GetPhotometricInterpretationValue ( File const & f ) [static]`

27.146.2.10 `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)

27.146.2.11 `static unsigned int gdcm::ImageHelper::GetPlanarConfigurationValue ( const File & f ) [static]`

27.146.2.12 `static const ByteValue* gdcm::ImageHelper::GetPointerFromElement ( Tag const & tag, File const & f ) [static]`

Moved from PixampReader to here. Generally used for photometric interpretation.

27.146.2.13 `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

27.146.2.14 `static Tag gdcm::ImageHelper::GetSpacingTagFromMediaStorage ( MediaStorage const & ms ) [static], [protected]`

27.146.2.15 `static std::vector<double> gdcm::ImageHelper::GetSpacingValue ( File const & f ) [static]`

Set/Get [Spacing](#) from/to a [File](#).

27.146.2.16 `static Tag gdcm::ImageHelper::GetZSpacingTagFromMediaStorage ( MediaStorage const & ms ) [static], [protected]`

27.146.2.17 `static void gdcm::ImageHelper::SetDimensionsValue ( File & f, const Image & img ) [static]`

27.146.2.18 `static void gdcm::ImageHelper::SetDirectionCosinesValue ( DataSet & ds, const std::vector< double > & dircos ) [static]`

Set Direction Cosines (IOP) from/to a file When [IOD](#) does not defines what is IOP (eg. typically Secondary Capture [Image](#) Storage) this call will simply remove the IOP attribute. Else in case of MR/CT image storage, this call will properly lookup the correct attribute to store the IOP.

27.146.2.19 `static void gdcm::ImageHelper::SetForcePixelSpacing ( bool ) [static]`

GDCM 1.x compatibility issue: When using ReWrite an MR [Image](#) Storage would be rewritten as Secondary Capture [Object](#) while still having a Pixel [Spacing](#) tag (0028,0030). If you have deal with those files, use this very special flag to handle them Unless explicitly set elsewhere by the standard, it will use value from 0028,0030 / 0018,0088 for the Pixel [Spacing](#) of the [Image](#)

27.146.2.20 `static void gdcm::ImageHelper::SetForceRescaleInterceptSlope ( bool ) [static]`

GDCM 1.x compatibility issue: when using ReWrite an MR [Image](#) Storage would be rewritten with a Rescale Slope/↔ Intercept while the standard would prohibit this (Philips Medical [System](#) is still doing that) Unless explicitly set elsewhere by the standard, it will use value from 0028,1052 / 0028,1053 for the Rescale Slope & Rescale Intercept values

27.146.2.21 `static void gdcm::ImageHelper::SetOriginValue ( DataSet & ds, const Image & img ) [static]`

27.146.2.22 `static void gdcm::ImageHelper::SetRescaleInterceptSlopeValue ( File & f, const Image & img ) [static]`

27.146.2.23 `static void gdcm::ImageHelper::SetSpacingValue ( DataSet & ds, const std::vector< double > & spacing ) [static]`

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

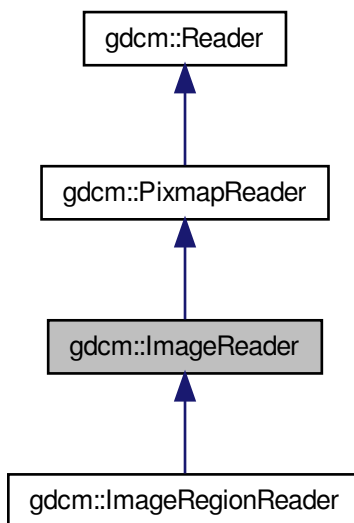
- [gdcmImageHelper.h](#)

## 27.147 gdcm::ImageReader Class Reference

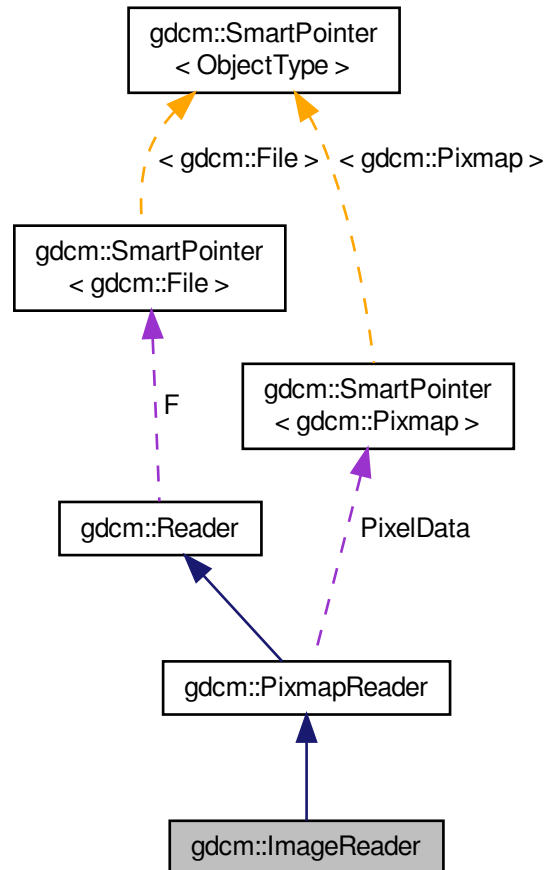
[ImageReader](#).

```
#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

### 27.147.1 Detailed Description

[ImageReader](#).

#### Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::Image](#) representation [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space.

#### See also

[Image](#)

#### Examples:

[BasicImageAnonymizer.cs](#), [CheckBigEndianBug.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [HelloVizWorld.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [ReadMultiTimesException.cxx](#), and [threadgdcm.cxx](#).

### 27.147.2 Constructor & Destructor Documentation

27.147.2.1 `gdcm::ImageReader::ImageReader ( )`

27.147.2.2 `virtual gdcm::ImageReader::~ImageReader ( ) [virtual]`

### 27.147.3 Member Function Documentation

27.147.3.1 `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](#), and [threadgdcm.cxx](#).

27.147.3.2 `Image& gdcm::ImageReader::GetImage ( )`

27.147.3.3 `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](#).

27.147.3.4 `bool gdcm::ImageReader::ReadACRNEMAImage ( ) [protected],[virtual]`

Reimplemented from [gdcm::PixmapReader](#).

27.147.3.5 `bool gdcm::ImageReader::ReadImage ( MediaStorage const & ms ) [protected],[virtual]`

Reimplemented from [gdcm::PixmapReader](#).

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

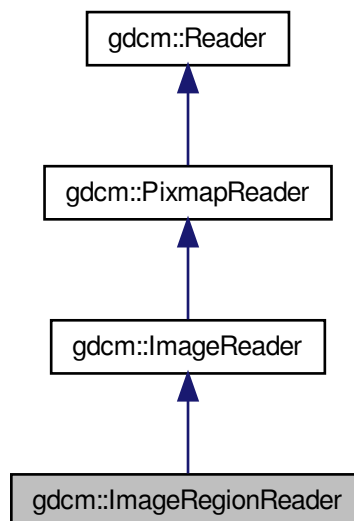
- [gdcmImageReader.h](#)

## 27.148 gdcm::ImageRegionReader Class Reference

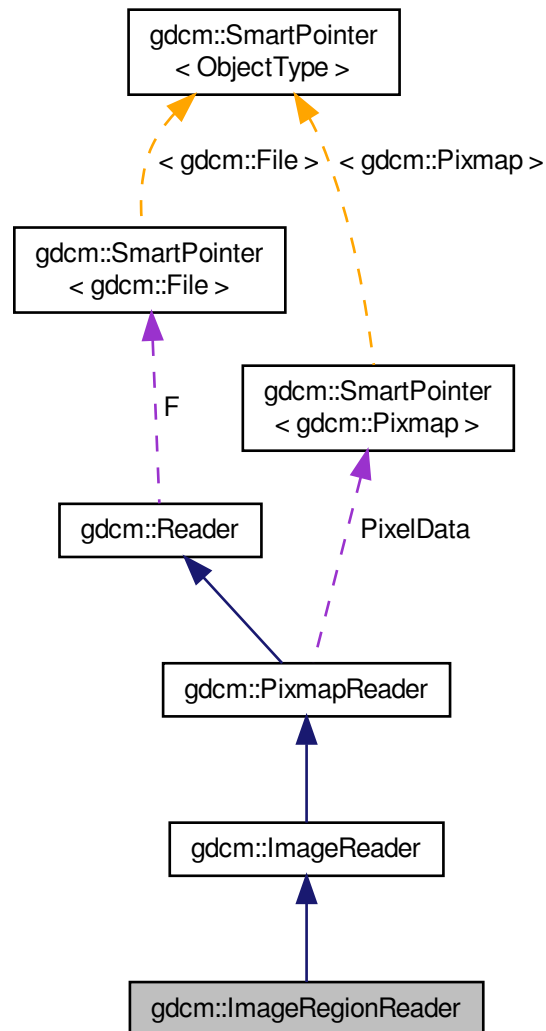
[ImageRegionReader](#).

```
#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

### 27.148.1 Detailed Description

[ImageRegionReader](#).

See also

[ImageReader](#)

Examples:

[ExtractImageRegion.cs](#), and [ExtractImageRegionWithLUT.cs](#).

### 27.148.2 Constructor & Destructor Documentation

27.148.2.1 `gdcm::ImageRegionReader::ImageRegionReader ( )`

27.148.2.2 `gdcm::ImageRegionReader::~~ImageRegionReader ( )`

### 27.148.3 Member Function Documentation

27.148.3.1 `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

27.148.3.2 `Region const& gdcm::ImageRegionReader::GetRegion ( ) const`

27.148.3.3 `bool gdcm::ImageRegionReader::Read ( ) [protected],[virtual]`

To prevent user from calling super class [Read\(\)](#) function.

Reimplemented from [gdcm::ImageReader](#).

27.148.3.4 `bool gdcm::ImageRegionReader::ReadInformation ( )`

Read meta information (not Pixel Data) from the DICOM file.

Returns

false upon error

27.148.3.5 `bool gdcM::ImageRegionReader::ReadIntoBuffer ( char * inreadbuffer, size_t buflen )`

Read into buffer:

Returns

false upon error

27.148.3.6 `void gdcM::ImageRegionReader::SetRegion ( Region const & region )`

Set/Get [Region](#) to be read.

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

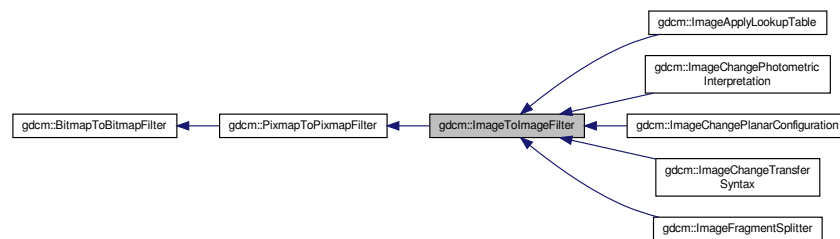
- [gdcMImageRegionReader.h](#)

## 27.149 gdcM::ImageToImageFilter Class Reference

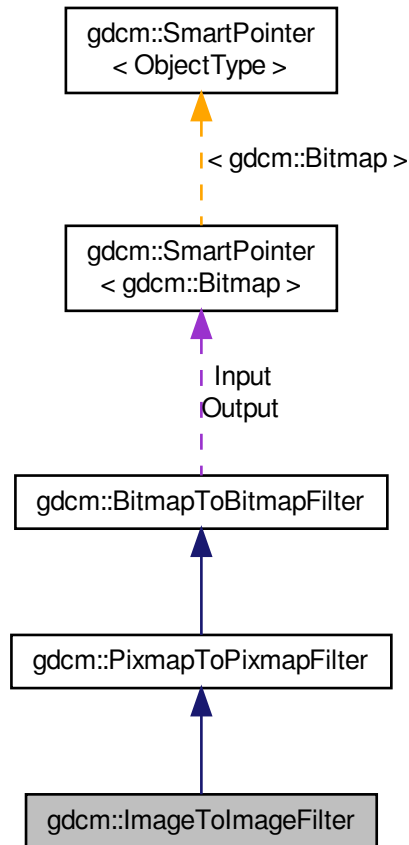
[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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

### 27.149.1 Detailed Description

[ImageToImageFilter](#) class Super class for all filter taking an image and producing an output image.

## 27.149.2 Constructor & Destructor Documentation

27.149.2.1 `gdcm::ImageToImageFilter::ImageToImageFilter ( )`

27.149.2.2 `gdcm::ImageToImageFilter::~~ImageToImageFilter ( )` `[inline]`

## 27.149.3 Member Function Documentation

27.149.3.1 `Image& gdcm::ImageToImageFilter::GetInput ( )`

27.149.3.2 `const Image& gdcm::ImageToImageFilter::GetOutput ( ) const`

Get Output image.

Examples:

[CompressImage.cxx](#).

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

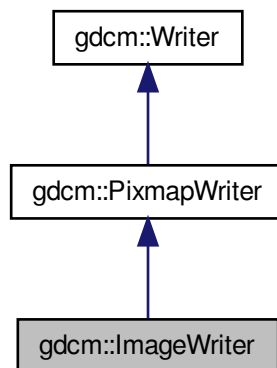
- [gdcmImageToImageFilter.h](#)

## 27.150 gdcm::ImageWriter Class Reference

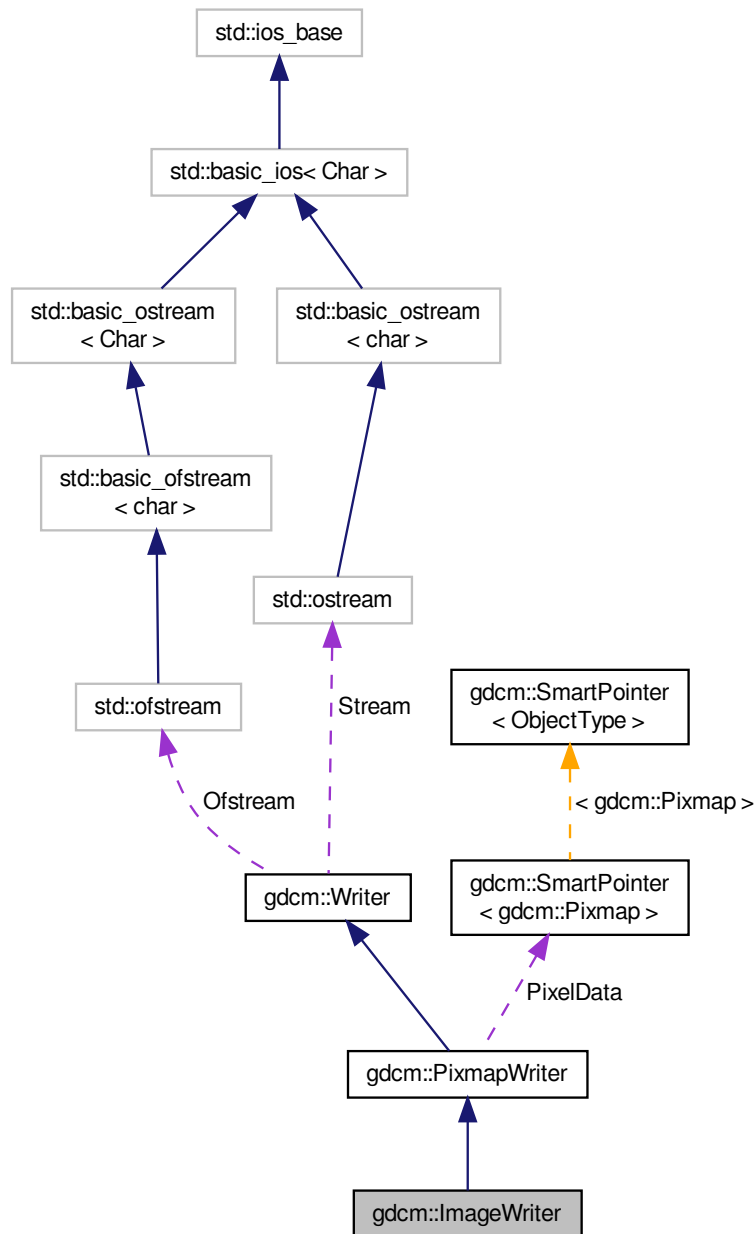
[ImageWriter](#).

```
#include <gdcmImageWriter.h>
```

Inheritance diagram for `gdcm::ImageWriter`:



Collaboration diagram for gdcm::ImageWriter:



## Public Member Functions

- [ImageWriter](#) ()
- [~ImageWriter](#) ()
- `const Image & GetImage () const`

- [Image](#) & [GetImage](#) ()
- bool [Write](#) ()  
*Write.*

## Additional Inherited Members

### 27.150.1 Detailed Description

[ImageWriter](#).

Examples:

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), and [MergeTwoFiles.cxx](#).

### 27.150.2 Constructor & Destructor Documentation

27.150.2.1 `gdcmm::ImageWriter::ImageWriter ( )`

27.150.2.2 `gdcmm::ImageWriter::~~ImageWriter ( )`

### 27.150.3 Member Function Documentation

27.150.3.1 `const Image& gdcmm::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 [gdcmm::PixmapWriter](#).

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), and [iU22tomultisc.cxx](#).

27.150.3.2 `Image& gdcmm::ImageWriter::GetImage ( )` `[inline],[virtual]`

Reimplemented from [gdcmm::PixmapWriter](#).

27.150.3.3 `bool gdcmm::ImageWriter::Write ( )` `[virtual]`

*Write.*

Reimplemented from [gdcmm::Writer](#).

Examples:

[CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [GenFakelImage.cxx](#), [HelloVizWorld.cxx](#), [iU22tomultisc.cxx](#), and [MergeTwoFiles.cxx](#).

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

- [gdcmmImageWriter.h](#)

## 27.151 gdcm::network::ImplementationClassUIDSub Class Reference

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS \(A-ASSOCIAT↔E-RQ\)](#)

```
#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

### 27.151.1 Detailed Description

[ImplementationClassUIDSub](#) PS 3.7 [Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS \(A-ASSOCIAT↔E-RQ\)](#)

### 27.151.2 Constructor & Destructor Documentation

27.151.2.1 `gdcm::network::ImplementationClassUIDSub::ImplementationClassUIDSub ( )`

### 27.151.3 Member Function Documentation

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

27.151.3.2 `std::istream& gdcm::network::ImplementationClassUIDSub::Read ( std::istream & is )`

27.151.3.3 `size_t gdcm::network::ImplementationClassUIDSub::Size ( ) const`

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

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

- [gdcmImplementationClassUIDSub.h](#)

## 27.152 gdcm::network::ImplementationUIDSub Class Reference

[ImplementationUIDSub](#) [Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS \(A-ASSOCIATE-AC\)](#)

```
#include <gdcmImplementationUIDSub.h>
```

### Public Member Functions

- [ImplementationUIDSub](#) ()
- const std::ostream & [Write](#) (std::ostream &os) const

### 27.152.1 Detailed Description

[ImplementationUIDSub Table](#) D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)

### 27.152.2 Constructor & Destructor Documentation

27.152.2.1 `gdcmm::network::ImplementationUIDSub::ImplementationUIDSub ( )`

### 27.152.3 Member Function Documentation

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

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

- [gdcmmImplementationUIDSub.h](#)

## 27.153 gdcmm::network::ImplementationVersionNameSub Class Reference

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#include <gdcmmImplementationVersionNameSub.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

### 27.153.1 Detailed Description

[ImplementationVersionNameSub Table](#) D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 27.153.2 Constructor & Destructor Documentation

27.153.2.1 `gdcmm::network::ImplementationVersionNameSub::ImplementationVersionNameSub ( )`

### 27.153.3 Member Function Documentation

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

27.153.3.2 `std::istream& gdcmm::network::ImplementationVersionNameSub::Read ( std::istream & is )`

27.153.3.3 `size_t gdcmm::network::ImplementationVersionNameSub::Size ( ) const`



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

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

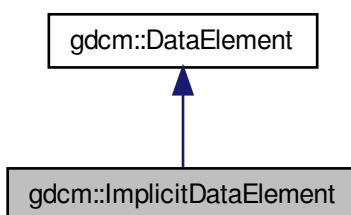
- [gdcmImplementationVersionNameSub.h](#)

## 27.154 gdcm::ImplicitDataElement Class Reference

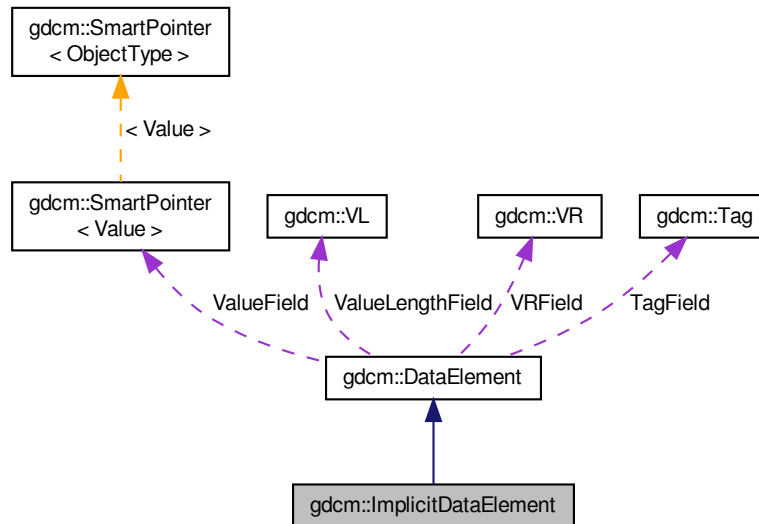
Class to represent an *Implicit VR Data Element*.

```
#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

### 27.154.1 Detailed Description

Class to represent an *Implicit VR Data Element*.

Note

bla

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

## 27.154.2 Member Function Documentation

27.154.2.1 VL gdcm::ImplicitDataElement::GetLength ( ) const

27.154.2.2 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::Read ( std::istream & *is* )

27.154.2.3 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadPreValue ( std::istream & *is* )

27.154.2.4 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValue ( std::istream & *is*, bool *readvalues* = true )

27.154.2.5 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadValueWithLength ( std::istream & *is*, VL & *length*, bool *readvalues* = true )

27.154.2.6 template<typename TSwap > std::istream& gdcm::ImplicitDataElement::ReadWithLength ( std::istream & *is*, VL & *length*, bool *readvalues* = true )

27.154.2.7 template<typename TSwap > const std::ostream& gdcm::ImplicitDataElement::Write ( std::ostream & *os* ) const

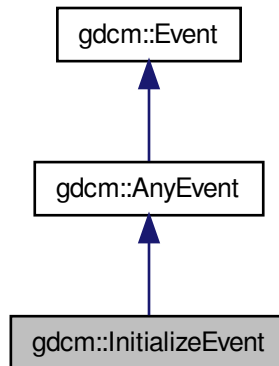
The documentation for this class was generated from the following file:

- [gdcmImplicitDataElement.h](#)

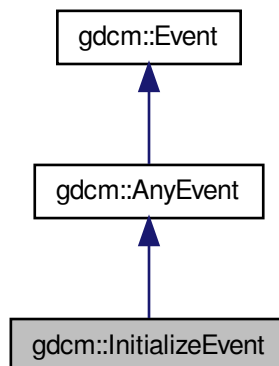
## 27.155 gdcm::InitializeEvent Class Reference

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

Inheritance diagram for `gdc::InitializeEvent`:



Collaboration diagram for `gdc::InitializeEvent`:



### Additional Inherited Members

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

- [gdcEvent.h](#)

## 27.156 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)

### 27.156.1 Detailed Description

Class for representing a [IOD](#).

#### Note

bla

#### See also

[Dict](#)

#### Examples:

[TraverseModules.cxx](#).

### 27.156.2 Member Typedef Documentation

27.156.2.1 typedef std::vector<[IODEntry](#)> [gdcm::IOD::MapIODEntry](#)

27.156.2.2 typedef MapIODEntry::size\_type [gdcm::IOD::SizeType](#)

### 27.156.3 Constructor & Destructor Documentation

27.156.3.1 [gdcm::IOD::IOD](#) ( ) `[inline]`

### 27.156.4 Member Function Documentation

27.156.4.1 `void gdcmm::IOD::AddIODEntry ( const IODEntry & iode )` `[inline]`

27.156.4.2 `void gdcmm::IOD::Clear ( )` `[inline]`

27.156.4.3 `const IODEntry& gdcmm::IOD::GetIODEntry ( SizeType idx ) const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.156.4.4 `SizeType gdcmm::IOD::GetNumberOfIODs ( ) const` `[inline]`

Examples:

[TraverseModules.cxx](#).

27.156.4.5 `Type gdcmm::IOD::GetTypeFromTag ( const Defs & defs, const Tag & tag ) const`

### 27.156.5 Friends And Related Function Documentation

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

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

- [gdcmmIOD.h](#)

## 27.157 gdcmm::IODEntry Class Reference

Class for representing a [IODEntry](#).

```
#include <gdcmmIODEntry.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)`

### 27.157.1 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](#).

### 27.157.2 Constructor & Destructor Documentation

27.157.2.1 `gdcm::IODEntry::IODEntry ( const char * name = " ", const char * ref = " ", const char * usag = " " ) [inline]`

### 27.157.3 Member Function Documentation

27.157.3.1 `const char* gdcm::IODEntry::GetIE ( ) const [inline]`

27.157.3.2 `const char* gdcm::IODEntry::GetName ( ) const [inline]`

27.157.3.3 `const char* gdcm::IODEntry::GetRef ( ) const [inline]`

Examples:

[TraverseModules.cxx](#).

27.157.3.4 `const char* gdcm::IODEntry::GetUsage ( ) const` `[inline]`

27.157.3.5 `Usage::UsageType gdcm::IODEntry::GetUsageType ( ) const`

27.157.3.6 `void gdcm::IODEntry::SetIE ( const char * ie )` `[inline]`

27.157.3.7 `void gdcm::IODEntry::SetName ( const char * name )` `[inline]`

27.157.3.8 `void gdcm::IODEntry::SetRef ( const char * ref )` `[inline]`

27.157.3.9 `void gdcm::IODEntry::SetUsage ( const char * usag )` `[inline]`

## 27.157.4 Friends And Related Function Documentation

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

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

- [gdcmIODEntry.h](#)

## 27.158 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*)



### 27.158.1 Detailed Description

Class for representing a [IODs](#).

Note

bla

See also

[IOD](#)

Examples:

[TraverseModules.cxx](#).

### 27.158.2 Member Typedef Documentation

27.158.2.1 `typedef std::map<IODName, IOD> gdcm::IODs::IODMapType`

27.158.2.2 `typedef IODMapType::const_iterator gdcm::IODs::IODMapTypeConstIterator`

27.158.2.3 `typedef std::string gdcm::IODs::IODName`

### 27.158.3 Constructor & Destructor Documentation

27.158.3.1 `gdcm::IODs::IODs ( )` `[inline]`

### 27.158.4 Member Function Documentation

27.158.4.1 `void gdcm::IODs::AddIOD ( const char * name, const IOD & module )` `[inline]`

27.158.4.2 `IODMapTypeConstIterator gdcm::IODs::Begin ( ) const` `[inline]`

27.158.4.3 `void gdcm::IODs::Clear ( )` `[inline]`

27.158.4.4 `IODMapTypeConstIterator gdcm::IODs::End ( ) const` `[inline]`

27.158.4.5 `const IOD& gdcm::IODs::GetIOD ( const char * name ) const` `[inline]`

### 27.158.5 Friends And Related Function Documentation

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

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

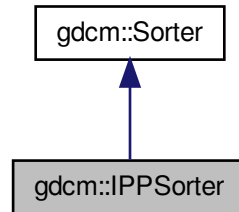
- [gdcmIODs.h](#)

## 27.159 gdcm::IPPSorter Class Reference

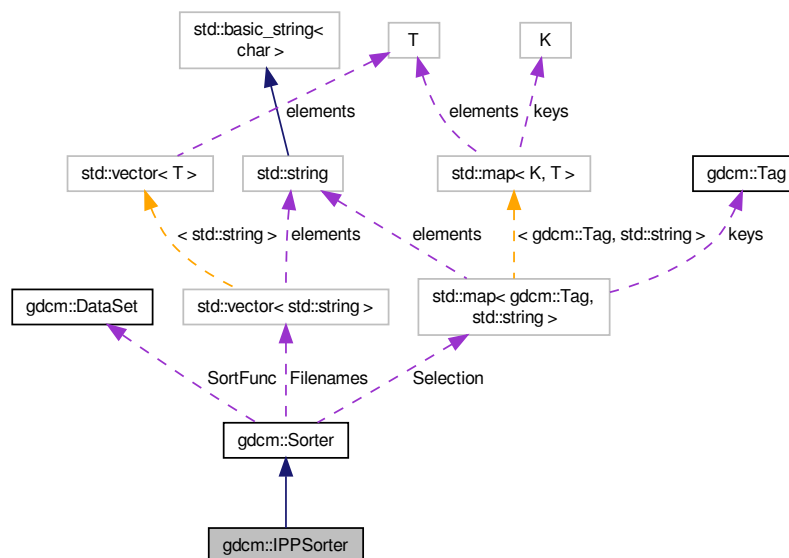
[IPPSorter](#) Implement a simple [Image](#) Position ([Patient](#)) sorter, along the [Image Orientation](#) ([Patient](#)) direction. This algorithm does NOT support duplicate and will FAIL in case of duplicate IPP.

```
#include <gdcmIPPSorter.h>
```

Inheritance diagram for `gdcm::IPPSorter`:



Collaboration diagram for `gdcm::IPPSorter`:



## Public Member Functions

- [IPPSorter](#) ()
- double [GetDirectionCosinesTolerance](#) () const
- double [GetZSpacing](#) () const
- double [GetZSpacingTolerance](#) () const
- void [SetComputeZSpacing](#) (bool b)
- void [SetDirectionCosinesTolerance](#) (double tol)

- void [SetDropDuplicatePositions](#) (bool b)
- void [SetZSpacingTolerance](#) (double tol)
- virtual bool [Sort](#) (std::vector< std::string > const &filenames)

### Protected Attributes

- bool [ComputeZSpacing](#)
- double [DirCosTolerance](#)
- bool [DropDuplicatePositions](#)
- double [ZSpacing](#)
- double [ZTolerance](#)

### Additional Inherited Members

#### 27.159.1 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

#### Examples:

[gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

#### 27.159.2 Constructor & Destructor Documentation

27.159.2.1 `gdcm::IPPSorter::IPPSorter ( )`

#### 27.159.3 Member Function Documentation

27.159.3.1 `double gdcm::IPPSorter::GetDirectionCosinesTolerance ( ) const` `[inline]`

27.159.3.2 `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:

[gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

27.159.3.3 `double gdcmm::IPPSorter::GetZSpacingTolerance ( ) const [inline]`

27.159.3.4 `void gdcmm::IPPSorter::SetComputeZSpacing ( bool b ) [inline]`

Functions related to Z-Spacing computation Set to true when sort algorithm should also perform a regular Z-Spacing computation using the [Image](#) Position ([Patient](#)) Potential reason for failure:

1. ALL slices are taken into account, if one slice is missing then ZSpacing will be set to 0 since the spacing will not be found to be regular along the [Series](#)

Examples:

[gdcmmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.159.3.5 `void gdcmm::IPPSorter::SetDirectionCosinesTolerance ( double tol ) [inline]`

Sometimes IOP along a series is slightly changing for example: "0.999081\0.0426953\0.00369272\0.0419025\0.955059\0.293439", "0.999081\0.0426953\0.00369275\0.0419025\0.955059\0.293439", "0.999081\0.0426952\0.00369272\0.0419025\0.955059\0.293439", We need an API to define the tolerance which is allowed. Internally the cross vector of each direction cosines is computed. The tolerance then define the the distance in between 1. 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).

27.159.3.6 `void gdcmm::IPPSorter::SetDropDuplicatePositions ( bool b ) [inline]`

Makes the [IPPSorter](#) ignore multiple images located at the same position. Only the first occurrence will be kept. DropDuplicatePositions defaults to false.

27.159.3.7 `void gdcmm::IPPSorter::SetZSpacingTolerance ( double tol ) [inline]`

1. Another reason for failure is that that Z-Spacing is only slightly changing (eg 1e-3) along the serie, a human can determine that this is ok and change the tolerance from its default value: 1e-6

Examples:

[gdcmmorthoplanes.cxx](#), and [reslicesphere.cxx](#).

27.159.3.8 `virtual bool gdcmm::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 achieved. Warning this does *NOT* imply that spacing is consistent, it only means the file are sorted according to IPP You should check if ZSpacing is 0 or not to deduce if file are actually a 3D volume

Reimplemented from [gdcmm::Sorter](#).

Examples:

[gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

## 27.159.4 Member Data Documentation

27.159.4.1 `bool gdcm::IPPSorter::ComputeZSpacing` [protected]

27.159.4.2 `double gdcm::IPPSorter::DirCosTolerance` [protected]

27.159.4.3 `bool gdcm::IPPSorter::DropDuplicatePositions` [protected]

27.159.4.4 `double gdcm::IPPSorter::ZSpacing` [protected]

27.159.4.5 `double gdcm::IPPSorter::ZTolerance` [protected]

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

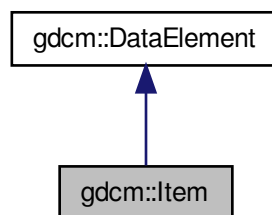
- [gdcmIPPSorter.h](#)

## 27.160 gdcm::Item Class Reference

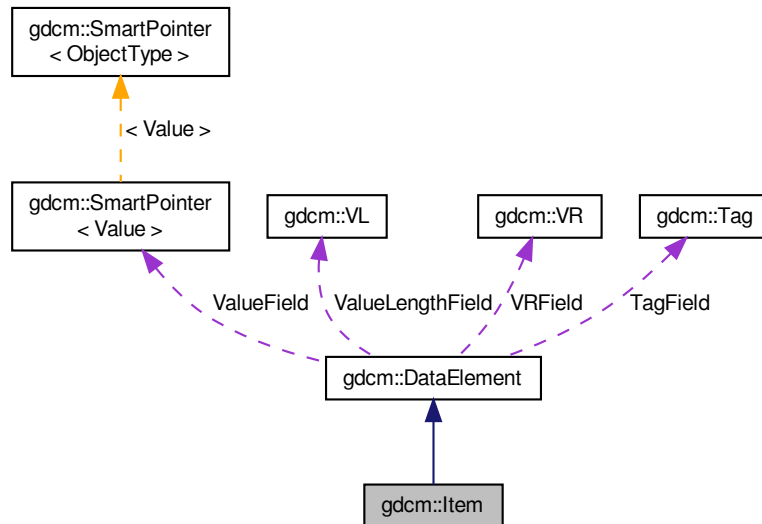
Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of VR SQ shall be encoded as a DICOM Standart Data [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.

```
#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

### 27.160.1 Detailed Description

Class to represent an [Item](#) A component of the value of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set . See PS 3.5 7.5.1 [Item](#) Encoding Rules Each [Item](#) of a Data [Element](#) of [VR](#) SQ shall be encoded as a DICOM Standard Data [Element](#) with a specific Data [Element](#) Tag of [Value](#) (FFFE,E000). The [Item](#) Tag is followed by a 4 byte [Item](#) Length field encoded in one of the following two ways Explicit/ Implicit.

#### Note

ITEM: A component of the [Value](#) of a Data [Element](#) that is of [Value](#) Representation Sequence of Items. An [Item](#) contains a Data Set.

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [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](#), [LargeVRDSExplicit.cxx](#), and [NewSequence.cs](#).

### 27.160.2 Constructor & Destructor Documentation

27.160.2.1 `gdcm::Item::Item ( )` [\[inline\]](#)

27.160.2.2 `gdcm::Item::Item ( Item const & val )` [\[inline\]](#)

### 27.160.3 Member Function Documentation

27.160.3.1 `void gdcm::Item::Clear ( )` [\[inline\]](#)

References `gdcm::DataElement::Clear()`.

Referenced by `gdcm::SequenceOfItems::Read()`.

27.160.3.2 `bool gdcm::Item::FindDataElement ( const Tag & t ) const` [\[inline\]](#)

#### Examples:

[ReadAndDumpDICOMDIR.cxx](#).

27.160.3.3 `const DataElement& gdcm::Item::GetDataElement ( const Tag & t ) const` [\[inline\]](#)

#### Examples:

[ReadAndDumpDICOMDIR.cxx](#).

27.160.3.4 `template<typename TDE > VL gdcm::Item::GetLength ( ) const`

27.160.3.5 `const DataSet& gdcm::Item::GetNestedDataSet ( ) const` [\[inline\]](#)

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.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](#), [LargeVRDSExplicit.cxx](#), and [NewSequence.cs](#).

[cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenSeqs.cxx](#), [GetSequenceData.cxx](#), [GetSubSequenceData.cxx](#), and [LargeVRDSExplicit.cxx](#).

Referenced by `gdcm::SequenceOfItems::Read()`.

**27.160.3.6** `DataSet& gdcm::Item::GetNestedDataSet ( ) [inline]`

**27.160.3.7** `void gdcm::Item::InsertDataElement ( const DataElement & de ) [inline]`

**27.160.3.8** `template<typename TDE , typename TSwap > std::istream& gdcm::Item::Read ( std::istream & is ) [inline]`

References `gdcm::DataSet::Clear()`, `gdcmDebugMacro`, `gdcmErrorMacro`, `gdcmWarningMacro`, `gdcm::DataSet::IsEmpty()`, and `gdcm::SwapperDoOp::Swap()`.

Referenced by `gdcm::SequenceOfItems::Read()`.

**27.160.3.9** `void gdcm::Item::SetNestedDataSet ( const DataSet & nested ) [inline]`

**27.160.3.10** `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()`.

## 27.160.4 Friends And Related Function Documentation

**27.160.4.1** `std::ostream& operator<< ( std::ostream & os, const Item & val ) [friend]`

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

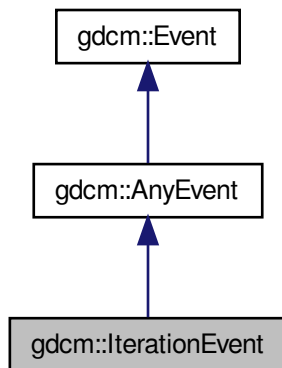
- [gdcmItem.h](#)

## 27.161 gdcm::IterationEvent Class Reference

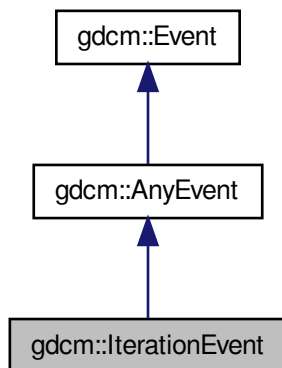
```
#include <gdcmEvent.h>
```



Inheritance diagram for gdcm::IterationEvent:



Collaboration diagram for gdcm::IterationEvent:



### Additional Inherited Members

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

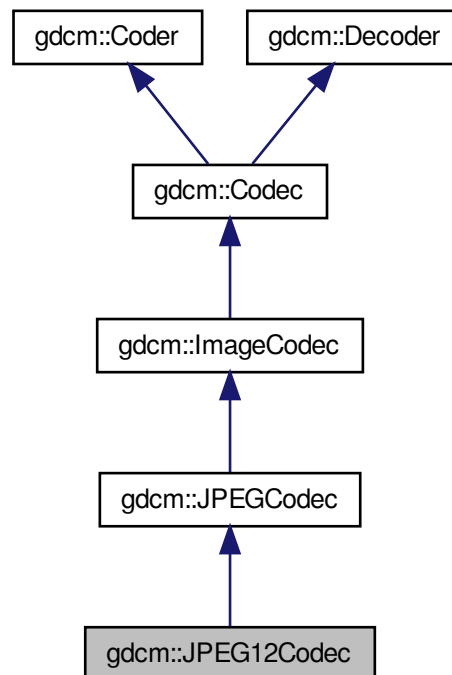
- [gdcmEvent.h](#)

## 27.162 gdcm::JPEG12Codec Class Reference

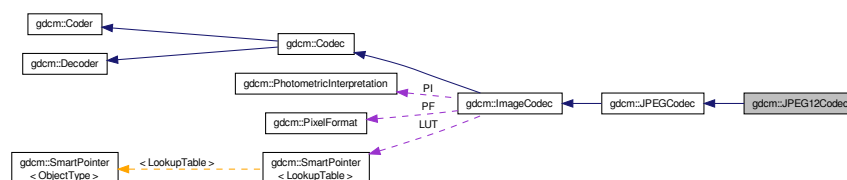
Class to do JPEG 12bits (lossy & lossless)

```
#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

#### 27.162.1 Detailed Description

Class to do JPEG 12bits (lossy & lossless)

#### Note

internal class

#### 27.162.2 Constructor & Destructor Documentation

27.162.2.1 `gdcm::JPEG12Codec::JPEG12Codec ( )`

27.162.2.2 `gdcm::JPEG12Codec::~~JPEG12Codec ( )`

#### 27.162.3 Member Function Documentation

27.162.3.1 `bool gdcm::JPEG12Codec::DecodeByStreams ( std::istream & is, std::ostream & os )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.162.3.2 `virtual bool gdcm::JPEG12Codec::EncodeBuffer ( std::ostream & os, const char * data, size_t datalen )`  
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.162.3.3 `bool gdcm::JPEG12Codec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.162.3.4 `bool gdcm::JPEG12Codec::InternalCode ( const char * input, unsigned long len, std::ostream & os )` [virtual]

Reimplemented from [gdcm::Coder](#).

27.162.3.5 `bool gdcm::JPEG12Codec::IsStateSuspension ( ) const` [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

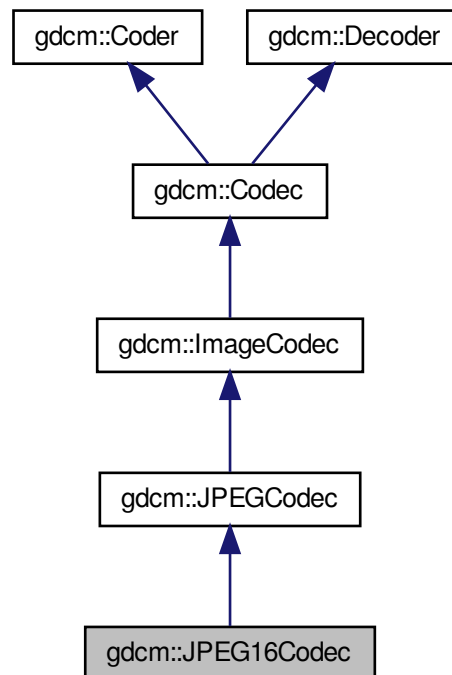
- [gdcmJPEG12Codec.h](#)

## 27.163 gdcm::JPEG16Codec Class Reference

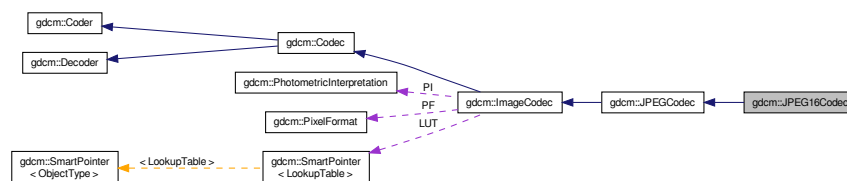
Class to do JPEG 16bits (lossless)

```
#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

### 27.163.1 Detailed Description

Class to do JPEG 16bits (lossless)

Note

internal class

### 27.163.2 Constructor & Destructor Documentation

27.163.2.1 `gdcm::JPEG16Codec::JPEG16Codec ( )`

27.163.2.2 `gdcm::JPEG16Codec::~~JPEG16Codec ( )`

### 27.163.3 Member Function Documentation

27.163.3.1 `bool gdcm::JPEG16Codec::DecodeByStreams ( std::istream & is, std::ostream & os )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.163.3.2 `virtual bool gdcm::JPEG16Codec::EncodeBuffer ( std::ostream & os, const char * data, size_t datalen )`  
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.163.3.3 `bool gdcm::JPEG16Codec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.163.3.4 `bool gdcm::JPEG16Codec::InternalCode ( const char * input, unsigned long len, std::ostream & os )` [virtual]

Reimplemented from [gdcm::Coder](#).

27.163.3.5 `bool gdcM::JPEG16Codec::IsStateSuspension ( ) const` [protected],[virtual]

Reimplemented from [gdcM::JPEGCodec](#).

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

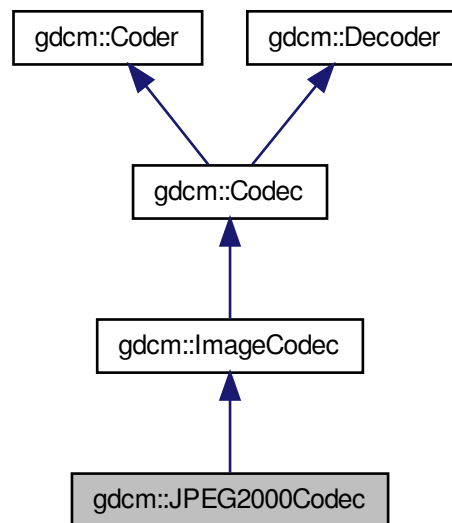
- [gdcMJPEG16Codec.h](#)

## 27.164 gdcM::JPEG2000Codec Class Reference

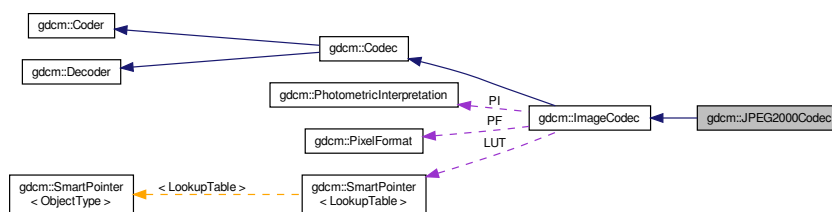
Class to do JPEG 2000.

```
#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 [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)

## Friends

- class [Bitmap](#)
- class [ImageRegionReader](#)

## Additional Inherited Members

### 27.164.1 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](#)

## 27.164.2 Constructor & Destructor Documentation

27.164.2.1 `gdcm::JPEG2000Codec::JPEG2000Codec ( )`

27.164.2.2 `gdcm::JPEG2000Codec::~~JPEG2000Codec ( )`

## 27.164.3 Member Function Documentation

27.164.3.1 `bool gdcm::JPEG2000Codec::CanCode ( TransferSyntax const & ) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.164.3.2 `bool gdcm::JPEG2000Codec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.164.3.3 `virtual ImageCodec* gdcm::JPEG2000Codec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

27.164.3.4 `bool gdcm::JPEG2000Codec::Code ( DataElement const & in_, DataElement & out_ )` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.164.3.5 `bool gdcm::JPEG2000Codec::Decode ( DataElement const &, DataElement & )` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.164.3.6 `bool gdcm::JPEG2000Codec::DecodeByStreams ( std::istream & is, std::ostream & os )` [protected],  
[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.164.3.7 `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]

27.164.3.8 `virtual bool gdcm::JPEG2000Codec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).



27.164.3.9 double gdcm::JPEG2000Codec::GetQuality ( unsigned int *idx* = 0 ) const

27.164.3.10 double gdcm::JPEG2000Codec::GetRate ( unsigned int *idx* = 0 ) const

27.164.3.11 void gdcm::JPEG2000Codec::SetNumberOfResolutions ( unsigned int *nres* )

27.164.3.12 void gdcm::JPEG2000Codec::SetQuality ( unsigned int *idx*, double *q* )

27.164.3.13 void gdcm::JPEG2000Codec::SetRate ( unsigned int *idx*, double *rate* )

27.164.3.14 void gdcm::JPEG2000Codec::SetReversible ( bool *res* )

27.164.3.15 void gdcm::JPEG2000Codec::SetTileSize ( unsigned int *tx*, unsigned int *ty* )

## 27.164.4 Friends And Related Function Documentation

27.164.4.1 friend class **Bitmap** [friend]

27.164.4.2 friend class **ImageRegionReader** [friend]

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

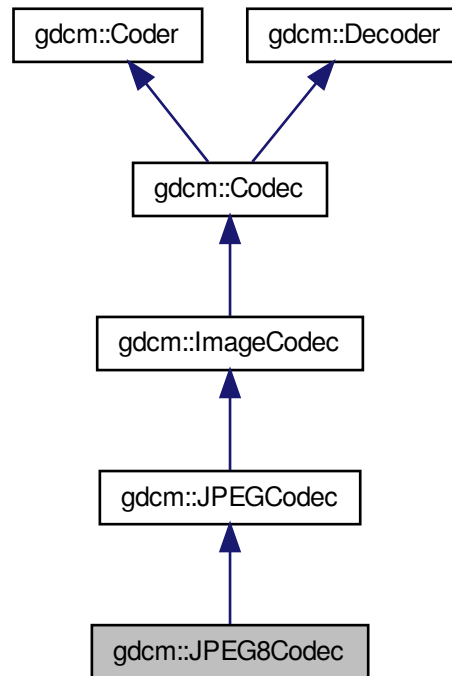
- [gdcmJPEG2000Codec.h](#)

## 27.165 gdcm::JPEG8Codec Class Reference

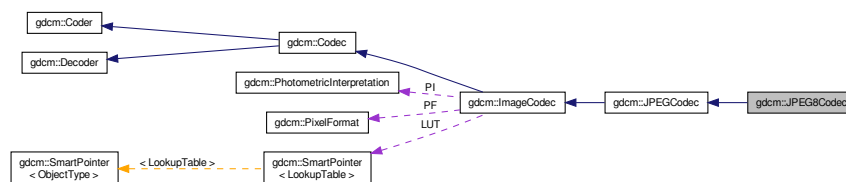
Class to do JPEG 8bits (lossy & lossless)

```
#include <gdcmJPEG8Codec.h>
```

Inheritance diagram for `gdcm::JPEG8Codec`:



Collaboration diagram for `gdcm::JPEG8Codec`:



## Public Member Functions

- [JPEG8Codec](#) ()
- [~JPEG8Codec](#) ()
- [DecodeByStreams](#) (std::istream &is, std::ostream &os)
- [GetHeaderInfo](#) (std::istream &is, [TransferSyntax](#) &ts)
- [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

### 27.165.1 Detailed Description

Class to do JPEG 8bits (lossy & lossless)

#### Note

internal class

### 27.165.2 Constructor & Destructor Documentation

27.165.2.1 `gdcm::JPEG8Codec::JPEG8Codec ( )`

27.165.2.2 `gdcm::JPEG8Codec::~~JPEG8Codec ( )`

### 27.165.3 Member Function Documentation

27.165.3.1 `bool gdcm::JPEG8Codec::DecodeByStreams ( std::istream & is, std::ostream & os )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.165.3.2 `virtual bool gdcm::JPEG8Codec::EncodeBuffer ( std::ostream & os, const char * data, size_t datalen )`  
[protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.165.3.3 `bool gdcm::JPEG8Codec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

27.165.3.4 `bool gdcm::JPEG8Codec::InternalCode ( const char * input, unsigned long len, std::ostream & os )` [virtual]

Reimplemented from [gdcm::Coder](#).

27.165.3.5 `bool gdcm::JPEG8Codec::IsStateSuspension ( ) const` [protected], [virtual]

Reimplemented from [gdcm::JPEGCodec](#).

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

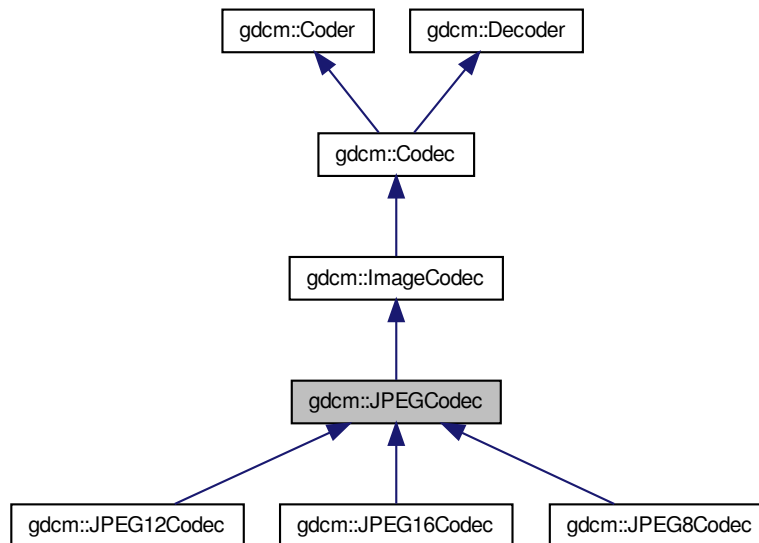
- [gdcmJPEG8Codec.h](#)

## 27.166 gdcm::JPEGCodec Class Reference

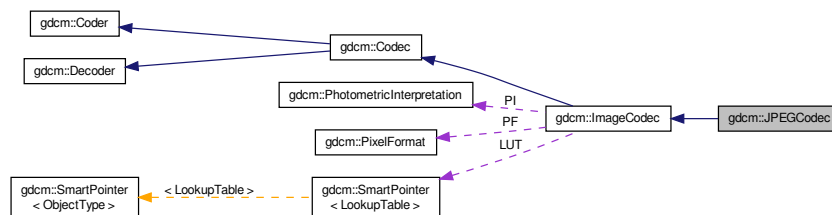
JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispatch in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#). It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.

```
#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 [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)
- virtual bool [IsStateSuspension](#) () const
- bool [IsValid](#) ([PhotometricInterpretation](#) const &pi)
- void [SetBitSample](#) (int bit)

## Protected Attributes

- int [BitSample](#)
- bool [Lossless](#)
- int [Quality](#)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 27.166.1 Detailed Description

JPEG codec Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::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/6b93af410f8c9](http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/6b93af410f8c9)

**Examples:**

[GetJPEGSamplePrecision.cxx](#).

**27.166.2 Constructor & Destructor Documentation**

**27.166.2.1** `gdcm::JPEGCodec::JPEGCodec ( )`

**27.166.2.2** `gdcm::JPEGCodec::~~JPEGCodec ( )`

**27.166.3 Member Function Documentation**

**27.166.3.1** `bool gdcm::JPEGCodec::CanCode ( TransferSyntax const & ) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

**27.166.3.2** `bool gdcm::JPEGCodec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

**27.166.3.3** `virtual ImageCodec* gdcm::JPEGCodec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

**27.166.3.4** `bool gdcm::JPEGCodec::Code ( DataElement const & in, DataElement & out )` [virtual]

Compress into JPEG.

Reimplemented from [gdcm::Coder](#).

**27.166.3.5** `void gdcm::JPEGCodec::ComputeOffsetTable ( bool b )`

Compute the offset table:

27.166.3.6 `bool gdcm::JPEGCodec::Decode ( DataElement const & , DataElement & )` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.166.3.7 `bool gdcm::JPEGCodec::DecodeByStreams ( std::istream & is, std::ostream & os )` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.166.3.8 `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]

27.166.3.9 `virtual bool gdcm::JPEGCodec::EncodeBuffer ( std::ostream & out, const char * inbuffer, size_t inlen )` [virtual]

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

27.166.3.10 `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](#).

27.166.3.11 `bool gdcm::JPEGCodec::GetLossless ( )` const

27.166.3.12 `double gdcm::JPEGCodec::GetQuality ( )` const

27.166.3.13 `virtual bool gdcm::JPEGCodec::IsStateSuspension ( )` const [protected],[virtual]

Reimplemented in [gdcm::JPEG12Codec](#), [gdcm::JPEG16Codec](#), and [gdcm::JPEG8Codec](#).

27.166.3.14 `bool gdcm::JPEGCodec::IsValid ( PhotometricInterpretation const & pi )` [protected],[virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.166.3.15 `void gdcm::JPEGCodec::SetBitSample ( int bit )` [protected]

27.166.3.16 `void gdcm::JPEGCodec::SetLossless ( bool l )`

27.166.3.17 `void gdcm::JPEGCodec::SetPixelFormat ( PixelFormat const & pf )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.166.3.18 void gdcM::JPEGCodec::SetQuality ( double *q* )

## 27.166.4 Friends And Related Function Documentation

27.166.4.1 friend class ImageRegionReader [friend]

## 27.166.5 Member Data Documentation

27.166.5.1 int gdcM::JPEGCodec::BitSample [protected]

27.166.5.2 bool gdcM::JPEGCodec::Lossless [protected]

27.166.5.3 int gdcM::JPEGCodec::Quality [protected]

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

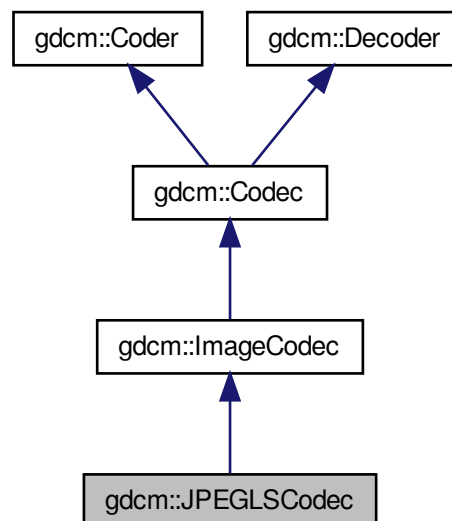
- [gdcMJPEGCodec.h](#)

## 27.167 gdcM::JPEGLSCodec Class Reference

JPEG-LS.

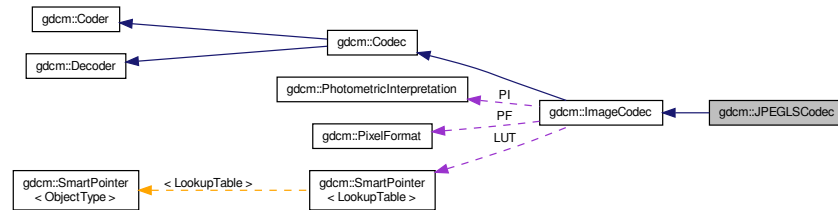
```
#include <gdcMJPEGLSCodec.h>
```

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 [DecodeExtent](#) (char \*buffer, unsigned int xmin, unsigned int xmax, unsigned int ymin, unsigned int ymax, unsigned int zmin, unsigned int zmax, std::istream &is)

## Friends

- class [ImageRegionReader](#)

## Additional Inherited Members

### 27.167.1 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 <http://charls.codeplex.com>

### 27.167.2 Constructor & Destructor Documentation

27.167.2.1 `gdcm::JPEGLSCodec::JPEGLSCodec ( )`

27.167.2.2 `gdcm::JPEGLSCodec::~~JPEGLSCodec ( )`

### 27.167.3 Member Function Documentation

27.167.3.1 `bool gdcm::JPEGLSCodec::CanCode ( TransferSyntax const & ) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.2 `bool gdcm::JPEGLSCodec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.3 `virtual ImageCodec* gdcm::JPEGLSCodec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

27.167.3.4 `bool gdcm::JPEGLSCodec::Code ( DataElement const & in_, DataElement & out_ )` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.167.3.5 `bool gdcm::JPEGLSCodec::Decode ( DataElement const & , DataElement & )` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.6 `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 )`

27.167.3.7 `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]

27.167.3.8 `unsigned long gdcm::JPEGLSCodec::GetBufferLength ( ) const` [inline]

27.167.3.9 `bool gdcm::JPEGLSCodec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.167.3.10 `bool gdcm::JPEGLSCodec::GetLossless ( ) const`

27.167.3.11 `void gdcm::JPEGLSCodec::SetBufferLength ( unsigned long l )` [inline]

27.167.3.12 `void gdcm::JPEGLSCodec::SetLossless ( bool l )`

27.167.3.13 `void gdcm::JPEGLSCodec::SetLossyError ( int error )`

[0-3] generally

## 27.167.4 Friends And Related Function Documentation

27.167.4.1 `friend class ImageRegionReader` [friend]

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

- [gdcmJPEGLSCodec.h](#)

## 27.168 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)`

### 27.168.1 Detailed Description

Examples:

[QIDO-RS.cxx](#).

## 27.168.2 Constructor & Destructor Documentation

27.168.2.1 `gdcm::JSON::JSON ( )`

27.168.2.2 `gdcm::JSON::~~JSON ( )`

## 27.168.3 Member Function Documentation

27.168.3.1 `bool gdcm::JSON::Code ( DataSet const & in, std::ostream & os )`

27.168.3.2 `bool gdcm::JSON::Decode ( std::istream & is, DataSet & out )`

27.168.3.3 `bool gdcm::JSON::GetPrettyPrint ( ) const`

27.168.3.4 `void gdcm::JSON::PrettyPrintOff ( )`

27.168.3.5 `void gdcm::JSON::PrettyPrintOn ( )`

Examples:

[QIDO-RS.cxx](#).

27.168.3.6 `void gdcm::JSON::SetPrettyPrint ( bool onoff )`

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

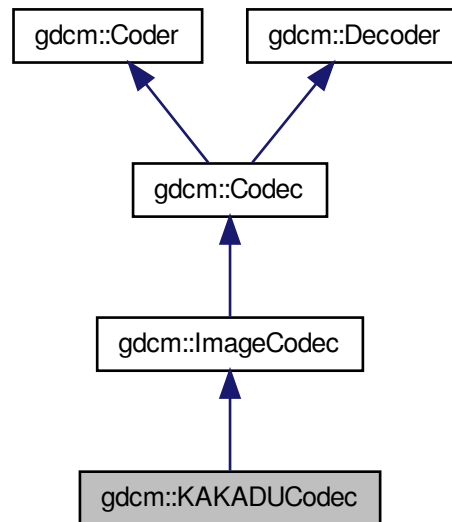
- [gdcmJSON.h](#)

## 27.169 gdcm::KAKADUCodec Class Reference

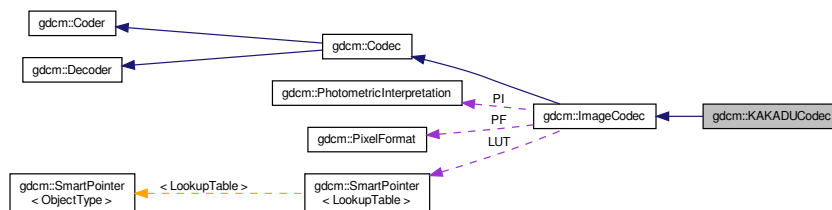
[KAKADUCodec](#).

```
#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

### 27.169.1 Detailed Description

[KAKADUCodec](#).

### 27.169.2 Constructor & Destructor Documentation

27.169.2.1 `gdcm::KAKADUCodec::KAKADUCodec ( )`

27.169.2.2 `gdcm::KAKADUCodec::~~KAKADUCodec ( )`

### 27.169.3 Member Function Documentation

27.169.3.1 `bool gdcm::KAKADUCodec::CanCode ( TransferSyntax const & ) const` [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.2 `bool gdcm::KAKADUCodec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.169.3.3 `virtual ImageCodec* gdcm::KAKADUCodec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

27.169.3.4 `bool gdcm::KAKADUCodec::Code ( DataElement const & in_, DataElement & out_ )` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.169.3.5 `bool gdcm::KAKADUCodec::Decode ( DataElement const &, DataElement & )` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

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

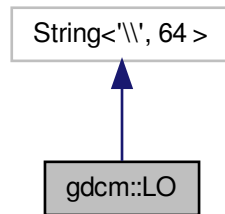
- [gdcmKAKADUCodec.h](#)

## 27.170 gdcm::LO Class Reference

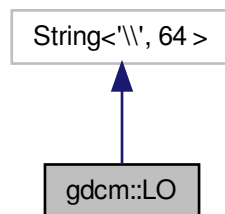
[LO](#).

```
#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=npow)
- bool [IsValid](#) () const

## 27.170.1 Detailed Description

[LO](#).

Note

TODO

## 27.170.2 Member Typedef Documentation

27.170.2.1 typedef Superclass::const\_iterator [gdcm::LO::const\\_iterator](#)

27.170.2.2 typedef Superclass::const\_reference [gdcm::LO::const\\_reference](#)

27.170.2.3 typedef Superclass::const\_reverse\_iterator [gdcm::LO::const\\_reverse\\_iterator](#)

27.170.2.4 typedef Superclass::difference\_type [gdcm::LO::difference\\_type](#)

27.170.2.5 typedef Superclass::iterator [gdcm::LO::iterator](#)

27.170.2.6 typedef Superclass::pointer [gdcm::LO::pointer](#)

27.170.2.7 typedef Superclass::reference [gdcm::LO::reference](#)

27.170.2.8 typedef Superclass::reverse\_iterator [gdcm::LO::reverse\\_iterator](#)

27.170.2.9 typedef Superclass::size\_type [gdcm::LO::size\\_type](#)

27.170.2.10 typedef [String](#)<'\',64> [gdcm::LO::Superclass](#)

27.170.2.11 typedef Superclass::value\_type [gdcm::LO::value\\_type](#)

## 27.170.3 Constructor & Destructor Documentation

27.170.3.1 [gdcm::LO::LO](#) ( ) [\[inline\]](#)

27.170.3.2 [gdcm::LO::LO](#) ( const [value\\_type](#) \* s ) [\[inline\]](#)



27.170.3.3 `gdcm::LO::LO ( const value_type * s, size_type n )` `[inline]`

27.170.3.4 `gdcm::LO::LO ( const Superclass & s, size_type pos = 0, size_type n = npos )` `[inline]`

## 27.170.4 Member Function Documentation

27.170.4.1 `bool gdcm::LO::IsValid ( ) const` `[inline]`

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

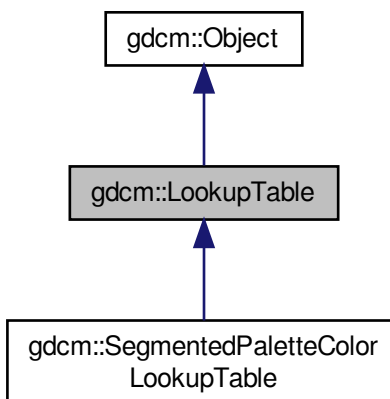
- [gdcmLO.h](#)

## 27.171 gdcm::LookupTable Class Reference

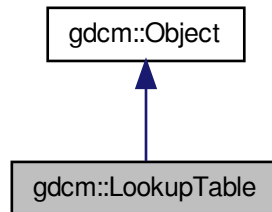
[LookupTable](#) class.

```
#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

### 27.171.1 Detailed Description

[LookupTable](#) class.

Examples:

[ExtractImageRegionWithLUT.cs](#), and [ScanDirectory.java](#).

### 27.171.2 Member Enumeration Documentation

#### 27.171.2.1 enum gdcm::LookupTable::LookupTableType

Enumerator

**RED**

**GREEN**

**BLUE**

**GRAY**

**UNKNOWN**

### 27.171.3 Constructor & Destructor Documentation

#### 27.171.3.1 gdcm::LookupTable::LookupTable ( )

#### 27.171.3.2 gdcm::LookupTable::~~LookupTable ( )

27.171.3.3 `gdcm::LookupTable::LookupTable ( LookupTable const & lut ) [inline]`

#### 27.171.4 Member Function Documentation

27.171.4.1 `void gdcm::LookupTable::Allocate ( unsigned short bitsample = 8 )`

Allocate the LUT.

27.171.4.2 `void gdcm::LookupTable::Clear ( )`

Clear the LUT.

27.171.4.3 `void gdcm::LookupTable::Decode ( std::istream & is, std::ostream & os ) const`

Decode the LUT.

27.171.4.4 `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

27.171.4.5 `unsigned short gdcm::LookupTable::GetBitSample ( ) const [inline]`

return the bit sample

27.171.4.6 `bool gdcm::LookupTable::GetBufferAsRGBA ( unsigned char * rgba ) const`

return the LUT as RGBA buffer

27.171.4.7 `void gdcm::LookupTable::GetLUT ( LookupTableType type, unsigned char * array, unsigned int & length ) const`

27.171.4.8 `void gdcm::LookupTable::GetLUTDescriptor ( LookupTableType type, unsigned short & length, unsigned short & subscript, unsigned short & bitsize ) const`

27.171.4.9 `unsigned int gdcm::LookupTable::GetLUTLength ( LookupTableType type ) const`

27.171.4.10 `const unsigned char* gdcm::LookupTable::GetPointer ( ) const`

return a raw pointer to the LUT

27.171.4.11 `void gdcm::LookupTable::InitializeBlueLUT ( unsigned short length, unsigned short subscript, unsigned short bitsize )`

27.171.4.12 `bool gdcm::LookupTable::Initialized ( ) const`

return whether the LUT has been initialized

27.171.4.13 void gdcm::LookupTable::InitializeGreenLUT ( unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize* )

27.171.4.14 void gdcm::LookupTable::InitializeLUT ( LookupTableType *type*, unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize* )

Generic interface:

27.171.4.15 void gdcm::LookupTable::InitializeRedLUT ( unsigned short *length*, unsigned short *subscript*, unsigned short *bitsize* )

RED / GREEN / BLUE specific:

27.171.4.16 void gdcm::LookupTable::Print ( std::ostream & ) const [inline],[virtual]

Reimplemented from [gdcm::Object](#).

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

27.171.4.17 void gdcm::LookupTable::SetBlueLUT ( const unsigned char \* *blue*, unsigned int *length* )

27.171.4.18 void gdcm::LookupTable::SetGreenLUT ( const unsigned char \* *green*, unsigned int *length* )

27.171.4.19 virtual void gdcm::LookupTable::SetLUT ( LookupTableType *type*, const unsigned char \* *array*, unsigned int *length* ) [virtual]

Reimplemented in [gdcm::SegmentedPaletteColorLookupTable](#).

27.171.4.20 void gdcm::LookupTable::SetRedLUT ( const unsigned char \* *red*, unsigned int *length* )

27.171.4.21 bool gdcm::LookupTable::WriteBufferAsRGBA ( const unsigned char \* *rgba* )

Write the LUT as RGBA.

## 27.171.5 Member Data Documentation

27.171.5.1 unsigned short gdcm::LookupTable::BitSample [protected]

27.171.5.2 bool gdcm::LookupTable::IncompleteLUT [protected]

27.171.5.3 LookupTableInternal\* gdcm::LookupTable::Internal [protected]

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

- [gdcmLookupTable.h](#)

## 27.172 gdcm::Scanner::Itstr Struct Reference

```
#include <gdcmScanner.h>
```

## Public Member Functions

- bool [operator\(\)](#) (const char \*s1, const char \*s2) const

### 27.172.1 Member Function Documentation

27.172.1.1 bool [gdcmm::Scanner::ltstr::operator\(\)](#) ( const char \* s1, const char \* s2 ) const [\[inline\]](#)

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

- [gdcmmScanner.h](#)

## 27.173 gdcmm::Macro Class Reference

Class for representing a [Macro](#).

```
#include <gdcmmMacro.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)

### 27.173.1 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](#)

## 27.173.2 Member Typedef Documentation

27.173.2.1 `typedef std::vector<std::string> gdcmmacros::Macro::ArrayIncludeMacroType`

27.173.2.2 `typedef std::map<Tag, MacroEntry> gdcmmacros::Macro::MapModuleEntry`

## 27.173.3 Constructor & Destructor Documentation

27.173.3.1 `gdcmmacros::Macro ( ) [inline]`

## 27.173.4 Member Function Documentation

27.173.4.1 `void gdcmmacros::Macro::AddMacroEntry ( const Tag & tag, const MacroEntry & module ) [inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

27.173.4.2 `void gdcmmacros::Macro::Clear ( ) [inline]`

27.173.4.3 `bool gdcmmacros::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.

27.173.4.4 `const MacroEntry& gdcmmacros::Macro::GetMacroEntry ( const Tag & tag ) const`

27.173.4.5 `const char* gdcmmacros::Macro::GetName ( ) const [inline]`

27.173.4.6 `void gdcmmacros::Macro::SetName ( const char * name ) [inline]`

27.173.4.7 `bool gdcmmacros::Macro::Verify ( const DataSet & ds, Usage const & usage ) const`

## 27.173.5 Friends And Related Function Documentation

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

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

- [gdcmmacros.h](#)

## 27.174 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)

### 27.174.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples:

[TraverseModules.cxx](#).

### 27.174.2 Member Typedef Documentation

27.174.2.1 typedef std::map<std::string, [Macro](#)> [gdcmmacros::Macros::ModuleMapType](#)

### 27.174.3 Constructor & Destructor Documentation

27.174.3.1 [gdcmmacros::Macros](#) ( ) [\[inline\]](#)

### 27.174.4 Member Function Documentation

27.174.4.1 void [gdcmmacros::AddMacro](#) ( const char \* *ref*, const [Macro](#) & *module* ) [\[inline\]](#)

27.174.4.2 void [gdcmmacros::Clear](#) ( ) [\[inline\]](#)

27.174.4.3 const [Macro](#)& [gdcmmacros::GetMacro](#) ( const char \* *name* ) const [\[inline\]](#)

27.174.4.4 bool [gdcmmacros::IsEmpty](#) ( ) const [\[inline\]](#)



### 27.174.5 Friends And Related Function Documentation

27.174.5.1 `std::ostream& operator<< ( std::ostream &_os, const Macros &_val )` [*friend*]

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

- [gdcmMacros.h](#)

## 27.175 gdcm::network::MaximumLengthSub Class Reference

[MaximumLengthSub](#) Annex D [Table D.1-1](#) MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#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

### 27.175.1 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)

### 27.175.2 Constructor & Destructor Documentation

27.175.2.1 `gdcm::network::MaximumLengthSub::MaximumLengthSub ( )`

### 27.175.3 Member Function Documentation

27.175.3.1 `uint32_t gdcm::network::MaximumLengthSub::GetMaximumLength ( )` const [*inline*]

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

27.175.3.3 `std::istream& gdcm::network::MaximumLengthSub::Read ( std::istream & is )`

27.175.3.4 `void gdcm::network::MaximumLengthSub::SetMaximumLength ( uint32_t maximumlength )`

27.175.3.5 `size_t gdcm::network::MaximumLengthSub::Size ( )` const

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

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

- [gdcmMaximumLengthSub.h](#)

## 27.176 gdcm::MD5 Class Reference

Class for [MD5](#).

```
#include <gdcmMD5.h>
```

### Public Member Functions

- [MD5](#) ()
- [~MD5](#) ()

### Static Public Member Functions

- static bool [Compute](#) (const char \*buffer, unsigned long buf\_len, char digest\_str[33])
- static bool [ComputeFile](#) (const char \*filename, char digest\_str[33])

#### 27.176.1 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

#### 27.176.2 Constructor & Destructor Documentation

27.176.2.1 [gdcm::MD5::MD5 \( \)](#)

27.176.2.2 [gdcm::MD5::~~MD5 \( \)](#)

#### 27.176.3 Member Function Documentation

27.176.3.1 [static bool gdcm::MD5::Compute \( const char \\* \*buffer\*, unsigned long \*buf\\_len\*, char \*digest\\_str\*\[33\] \)](#) [static]

27.176.3.2 [static bool gdcm::MD5::ComputeFile \( const char \\* \*filename\*, char \*digest\\_str\*\[33\] \)](#) [static]

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

- [gdcmMD5.h](#)

## 27.177 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,`

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)

### 27.177.1 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](#), [StreamImageReaderTest.cxx](#), and [TestReader.cxx](#).

### 27.177.2 Member Enumeration Documentation

#### 27.177.2.1 enum gdcm::MediaStorage::MSType

##### Enumerator

***MediaStorageDirectoryStorage***  
***ComputedRadiographylImageStorage***  
***DigitalXRayImageStorageForPresentation***  
***DigitalXRayImageStorageForProcessing***  
***DigitalMammographylImageStorageForPresentation***  
***DigitalMammographylImageStorageForProcessing***  
***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*  
*FujiPrivateCRLImageStorage*  
*OphthalmicPhotography8BitImageStorage*  
*OphthalmicTomographyImageStorage*  
*VLMicroscopicImageStorage*  
*MS\_END*

Examples:

[GenerateStandardSOPClasses.cxx](#).

27.177.2.2 enum gdcmm::MediaStorage::ObjectType

Enumerator

*NoObject*  
*Video*  
*Waveform*  
*Audio*  
*PDF*  
*URI*  
*Segmentation*  
*ObjectEnd*



### 27.177.3 Constructor & Destructor Documentation

27.177.3.1 `gdcm::MediaStorage::MediaStorage ( MStype type = MS_END ) [inline]`

### 27.177.4 Member Function Documentation

27.177.4.1 `const char* gdcm::MediaStorage::GetModality ( ) const`

27.177.4.2 `unsigned int gdcm::MediaStorage::GetModalityDimension ( ) const`

27.177.4.3 `static const char* gdcm::MediaStorage::GetMSString ( MStype ts ) [static]`

Return the Media [String](#) associated. Will return NULL for MS\_END.

Examples:

[GenerateStandardSOPClasses.cxx](#).

Referenced by `gdcm::operator<<()`.

27.177.4.4 `static MStype gdcm::MediaStorage::GetMStype ( const char * str ) [static]`

Examples:

[TestReader.cxx](#).

27.177.4.5 `static unsigned int gdcm::MediaStorage::GetNumberOfModality ( ) [static]`

27.177.4.6 `static unsigned int gdcm::MediaStorage::GetNumberOfMSString ( ) [static]`

27.177.4.7 `static unsigned int gdcm::MediaStorage::GetNumberOfMStype ( ) [static]`

27.177.4.8 `const char* gdcm::MediaStorage::GetString ( ) const`

Return the Media [String](#) of the object.

Examples:

[CreateJPIPDataSet.cxx](#), [EncapsulateFileInRawData.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), and [StreamImageReaderTest.cxx](#).

27.177.4.9 `void gdcm::MediaStorage::GuessFromModality ( const char * modality, unsigned int dimension = 2 )`

27.177.4.10 `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

27.177.4.11 `bool gdcm::MediaStorage::IsUndefined ( ) const [inline]`

Examples:

[TestReader.cxx](#).

27.177.4.12 `gdcm::MediaStorage::operator MType ( ) const [inline]`

27.177.4.13 `bool gdcm::MediaStorage::SetFromDataSet ( DataSet const & ds )`

Advanced user only (functions should be protected level...) Those function are lower level than SetFromFile

27.177.4.14 `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](#), and [TestReader.cxx](#).

27.177.4.15 `bool gdcm::MediaStorage::SetFromHeader ( FileMetaInformation const & fmi )`

27.177.4.16 `bool gdcm::MediaStorage::SetFromModality ( DataSet const & ds )`

27.177.4.17 `void gdcm::MediaStorage::SetFromSourceImageSequence ( DataSet const & ds ) [protected]`

## 27.177.5 Friends And Related Function Documentation

27.177.5.1 `std::ostream& operator<< ( std::ostream & os, const MediaStorage & ms ) [friend]`

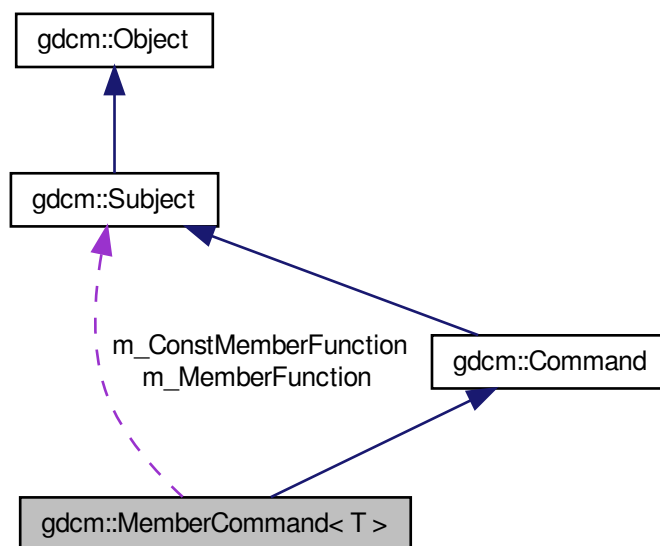
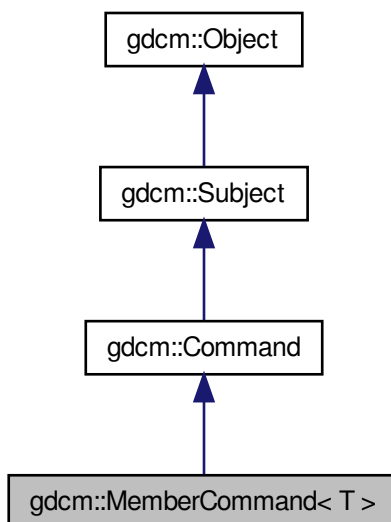
The documentation for this class was generated from the following file:

- [gdcmMediaStorage.h](#)

## 27.178 gdcm::MemberCommand< T > Class Template Reference

[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcmCommand.h>
```



## 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](#)

### 27.178.1 Detailed Description

```
template<class T>class gdcmmembercommand< 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](#).

### 27.178.2 Member Typedef Documentation

27.178.2.1 `template<class T > typedef MemberCommand gdcmmembercommand< T >::Self`

Standard class typedefs.

27.178.2.2 `template<class T> typedef void(T::* gdcmmembercommand< T >::TConstMemberFunctionPointer)(const Subject *, const Event &)`

27.178.2.3 `template<class T> typedef void(T::* gdcmmembercommand< T >::TMemberFunctionPointer)(Subject *, const Event &)`

pointer to a member function that takes a `Subject*` and the event

### 27.178.3 Constructor & Destructor Documentation

27.178.3.1 `template<class T> gdcmmembercommand< T >::MemberCommand ( ) [inline], [protected]`

Referenced by `gdcmmembercommand< T >::New()`.

27.178.3.2 `template<class T> virtual gdcmmembercommand< T >::~MemberCommand ( ) [inline], [protected], [virtual]`

### 27.178.4 Member Function Documentation

27.178.4.1 `template<class T> virtual void gdcmmembercommand< T >::Execute ( Subject * caller, const Event & event ) [inline], [virtual]`

Invoke the member function.

Implements [gdcmmembercommand::Command](#).

References `gdcmmembercommand< T >::m_MemberFunction`.

27.178.4.2 `template<class T> virtual void gdcmmembercommand< T >::Execute ( const Subject * caller, const Event & event ) [inline], [virtual]`

Invoke the member function with a const object.

Implements [gdcmmembercommand::Command](#).

References `gdcmmembercommand< T >::m_ConstMemberFunction`.

27.178.4.3 `template<class T> static SmartPointer<MemberCommand> gdcmmembercommand< T >::New ( ) [inline], [static]`

Method for creation through the object factory.

References `gdcmmembercommand< T >::MemberCommand()`.

27.178.4.4 `template<class T> void gdcmmembercommand< 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.

References `gdcmmembercommand< T >::m_MemberFunction`, and `gdcmmembercommand< T >::m_This`.

27.178.4.5 `template<class T> void gdcM::MemberCommand< T >::SetCallbackFunction ( T * object,  
TConstMemberFunctionPointer memberFunction ) [inline]`

References `gdcM::MemberCommand< T >::m_ConstMemberFunction`, and `gdcM::MemberCommand< T >::m_This`.

## 27.178.5 Member Data Documentation

27.178.5.1 `template<class T> TConstMemberFunctionPointer gdcM::MemberCommand< T  
>::m_ConstMemberFunction [protected]`

Referenced by `gdcM::MemberCommand< T >::Execute()`, and `gdcM::MemberCommand< T >::SetCallbackFunction()`.

27.178.5.2 `template<class T> TMemberFunctionPointer gdcM::MemberCommand< T >::m_MemberFunction  
[protected]`

Referenced by `gdcM::MemberCommand< T >::Execute()`, and `gdcM::MemberCommand< T >::SetCallbackFunction()`.

27.178.5.3 `template<class T> T* gdcM::MemberCommand< T >::m_This [protected]`

Referenced by `gdcM::MemberCommand< T >::SetCallbackFunction()`.

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

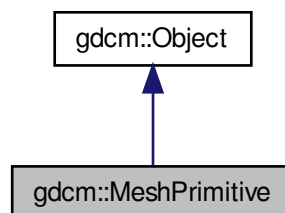
- [gdcMCommand.h](#)

## 27.179 gdcM::MeshPrimitive Class Reference

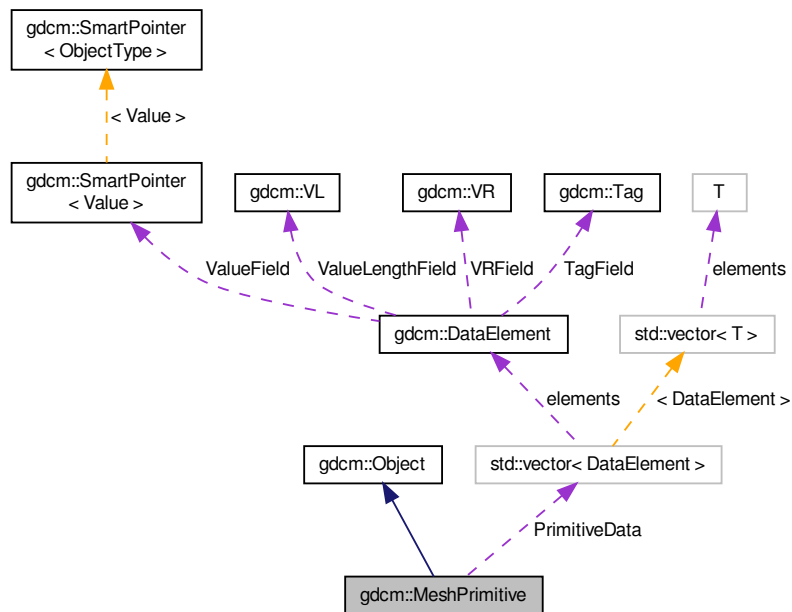
This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

```
#include <gdcMMeshPrimitive.h>
```

Inheritance diagram for `gdcM::MeshPrimitive`:



Collaboration diagram for gdcm::MeshPrimitive:



## Public Types

- enum `MPTType` {  
`VERTEX` = 0,  
`EDGE`,  
`TRIANGLE`,  
`TRIANGLE_STRIP`,  
`TRIANGLE_FAN`,  
`LINE`,  
`FACET`,  
`MPTType_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](#) ()
- [MPTType](#) [GetPrimitiveType](#) () const
- void [SetPrimitiveData](#) ([DataElement](#) const &de)
- void [SetPrimitiveData](#) (const unsigned int idx, [DataElement](#) const &de)
- void [SetPrimitivesData](#) ([PrimitivesData](#) const &DEs)
- void [SetPrimitiveType](#) (const [MPTType](#) type)

### Static Public Member Functions

- static [MPTType](#) [GetMPTType](#) (const char \*type)
- static const char \* [GetMPTTypeString](#) (const [MPTType](#) type)

### Protected Attributes

- [PrimitivesData](#) [PrimitiveData](#)
- [MPTType](#) [PrimitiveType](#)

### Additional Inherited Members

#### 27.179.1 Detailed Description

This class defines surface mesh primitives. It is designed from surface mesh primitives macro.

See also

PS 3.3 C.27.4

#### 27.179.2 Member Typedef Documentation

27.179.2.1 `typedef std::vector< DataElement > gdcm::MeshPrimitive::PrimitivesData`

#### 27.179.3 Member Enumeration Documentation

27.179.3.1 `enum gdcm::MeshPrimitive::MPTType`

This enumeration defines primitive types.

See also

PS 3.3 C.27.4.1

Enumerator

***VERTEX***  
***EDGE***  
***TRIANGLE***  
***TRIANGLE\_STRIP***  
***TRIANGLE\_FAN***  
***LINE***  
***FACET***  
***MPTType\_END***



### 27.179.4 Constructor & Destructor Documentation

27.179.4.1 `gdcm::MeshPrimitive::MeshPrimitive ( )`

27.179.4.2 `virtual gdcm::MeshPrimitive::~~MeshPrimitive ( )` `[virtual]`

### 27.179.5 Member Function Documentation

27.179.5.1 `void gdcm::MeshPrimitive::AddPrimitiveData ( DataElement const & de )`

27.179.5.2 `static MPTYPE gdcm::MeshPrimitive::GetMPTYPE ( const char * type )` `[static]`

27.179.5.3 `static const char* gdcm::MeshPrimitive::GetMPTYPEString ( const MPTYPE type )` `[static]`

27.179.5.4 `unsigned int gdcm::MeshPrimitive::GetNumberOfPrimitivesData ( ) const`

27.179.5.5 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( ) const`

27.179.5.6 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( )`

27.179.5.7 `const DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( const unsigned int idx ) const`

27.179.5.8 `DataElement& gdcm::MeshPrimitive::GetPrimitiveData ( const unsigned int idx )`

27.179.5.9 `const PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( ) const`

27.179.5.10 `PrimitivesData& gdcm::MeshPrimitive::GetPrimitivesData ( )`

27.179.5.11 `MPTYPE gdcm::MeshPrimitive::GetPrimitiveType ( ) const`

27.179.5.12 `void gdcm::MeshPrimitive::SetPrimitiveData ( DataElement const & de )`

27.179.5.13 `void gdcm::MeshPrimitive::SetPrimitiveData ( const unsigned int idx, DataElement const & de )`

27.179.5.14 `void gdcm::MeshPrimitive::SetPrimitivesData ( PrimitivesData const & DEs )`

27.179.5.15 `void gdcm::MeshPrimitive::SetPrimitiveType ( const MPTYPE type )`

### 27.179.6 Member Data Documentation

27.179.6.1 `PrimitivesData gdcm::MeshPrimitive::PrimitiveData` `[protected]`

27.179.6.2 `MPTYPE gdcm::MeshPrimitive::PrimitiveType` `[protected]`

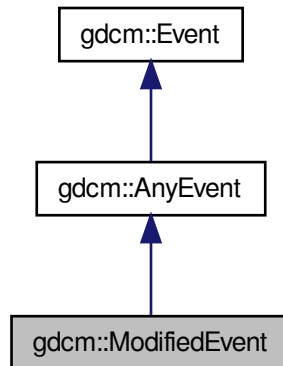
The documentation for this class was generated from the following file:

- [gdcmMeshPrimitive.h](#)

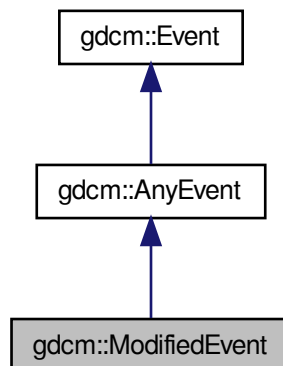
## 27.180 gdcm::ModifiedEvent Class Reference

```
#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:

- [gdcmEvent.h](#)

## 27.181 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)

#### 27.181.1 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](#).

## 27.181.2 Member Typedef Documentation

27.181.2.1 `typedef std::vector<std::string> gdcmmodule::ArrayIncludeMacrosType`

27.181.2.2 `typedef std::map<Tag, ModuleEntry> gdcmmodule::MapModuleEntry`

## 27.181.3 Constructor & Destructor Documentation

27.181.3.1 `gdcmmodule::Module ( ) [inline]`

## 27.181.4 Member Function Documentation

27.181.4.1 `void gdcmmodule::AddMacro ( const char * include ) [inline]`

27.181.4.2 `void gdcmmodule::AddModuleEntry ( const Tag & tag, const ModuleEntry & module ) [inline]`

Will add a [ModuleEntry](#) directly at root-level. See [Macro](#) for nested-included level.

27.181.4.3 `void gdcmmodule::Clear ( ) [inline]`

27.181.4.4 `bool gdcmmodule::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](#).

27.181.4.5 `const ModuleEntry& gdcmmodule::GetModuleEntryInMacros ( Macros const & macros, const Tag & tag ) const`

Examples:

[TraverseModules.cxx](#).

27.181.4.6 `const char* gdcmmodule::GetName ( ) const [inline]`

27.181.4.7 `void gdcmmodule::SetName ( const char * name ) [inline]`

27.181.4.8 `bool gdcmmodule::Verify ( const DataSet & ds, Usage const & usage ) const`

## 27.181.5 Friends And Related Function Documentation

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

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

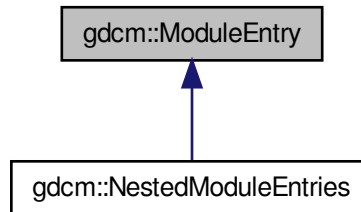
- [gdcmmodule.h](#)

## 27.182 gdcm::ModuleEntry Class Reference

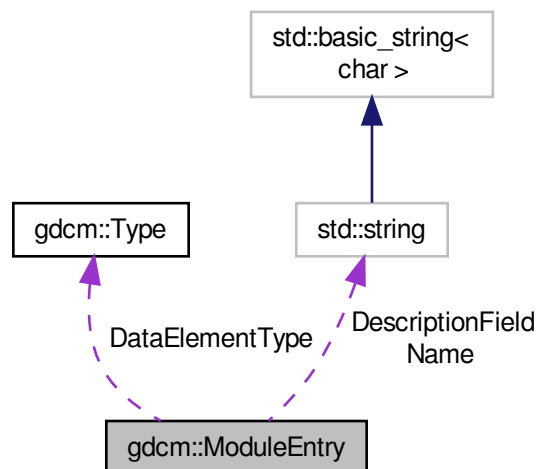
Class for representing a [ModuleEntry](#).

```
#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)

### 27.182.1 Detailed Description

Class for representing a [ModuleEntry](#).

#### Note

bla

#### See also

[DictEntry](#)

#### Examples:

[TraverseModules.cxx](#).

### 27.182.2 Member Typedef Documentation

#### 27.182.2.1 typedef std::string gdcm::ModuleEntry::Description

### 27.182.3 Constructor & Destructor Documentation

#### 27.182.3.1 gdcm::ModuleEntry::ModuleEntry ( const char \* *name* = " ", const char \* *type* = "3", const char \* *description* = " " ) [inline]

References [gdcm::Type::GetTypeType\(\)](#).

27.182.3.2 `virtual gdcm::ModuleEntry::~~ModuleEntry ( ) [inline],[virtual]`

### 27.182.4 Member Function Documentation

27.182.4.1 `const Description& gdcm::ModuleEntry::GetDescription ( ) const [inline]`

27.182.4.2 `const char* gdcm::ModuleEntry::GetName ( ) const [inline]`

27.182.4.3 `const Type& gdcm::ModuleEntry::GetType ( ) const [inline]`

Examples:

[TraverseModules.cxx](#).

27.182.4.4 `void gdcm::ModuleEntry::SetDescription ( const char * d ) [inline]`

27.182.4.5 `void gdcm::ModuleEntry::SetName ( const char * name ) [inline]`

27.182.4.6 `void gdcm::ModuleEntry::SetType ( const Type & type ) [inline]`

### 27.182.5 Friends And Related Function Documentation

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

### 27.182.6 Member Data Documentation

27.182.6.1 `Type gdcm::ModuleEntry::DataElementType [protected]`

Referenced by `gdcm::operator<<()`.

27.182.6.2 `Description gdcm::ModuleEntry::DescriptionField [protected]`

Referenced by `gdcm::operator<<()`.

27.182.6.3 `std::string gdcm::ModuleEntry::Name [protected]`

Referenced by `gdcm::operator<<()`.

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

- [gdcmModuleEntry.h](#)

## 27.183 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)

### 27.183.1 Detailed Description

Class for representing a [Modules](#).

Note

bla

See also

[Module](#)

Examples:

[TraverseModules.cxx](#).

### 27.183.2 Member Typedef Documentation

27.183.2.1 typedef std::map<std::string, [Module](#)> [gdcm::Modules::ModuleMapType](#)

### 27.183.3 Constructor & Destructor Documentation

27.183.3.1 [gdcm::Modules::Modules](#) ( ) [\[inline\]](#)

### 27.183.4 Member Function Documentation

27.183.4.1 void [gdcm::Modules::AddModule](#) ( const char \* *ref*, const [Module](#) & *module* ) [\[inline\]](#)

27.183.4.2 void [gdcm::Modules::Clear](#) ( ) [\[inline\]](#)

27.183.4.3 const [Module](#)& [gdcm::Modules::GetModule](#) ( const char \* *name* ) const [\[inline\]](#)

27.183.4.4 bool [gdcm::Modules::IsEmpty](#) ( ) const [\[inline\]](#)



### 27.183.5 Friends And Related Function Documentation

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

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

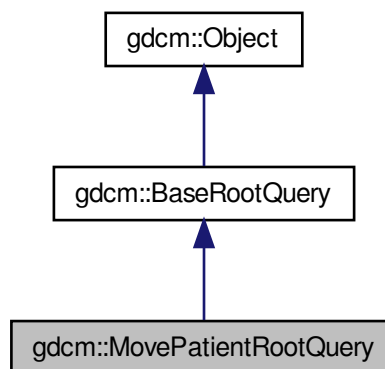
- [gdcmModules.h](#)

## 27.184 gdcm::MovePatientRootQuery Class Reference

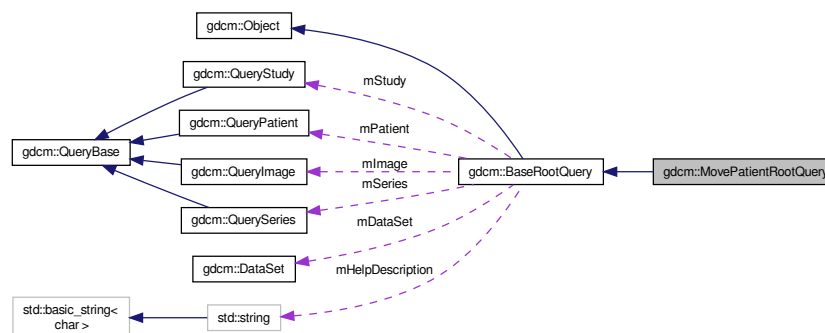
[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

```
#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

### 27.184.1 Detailed Description

[MovePatientRootQuery](#) contains: the class which will produce a dataset for c-move with patient root.

### 27.184.2 Constructor & Destructor Documentation

27.184.2.1 `gdcm::MovePatientRootQuery::MovePatientRootQuery ( )`

### 27.184.3 Member Function Documentation

27.184.3.1 `UIDs::TSName gdcm::MovePatientRootQuery::GetAbstractSyntaxUID ( ) const` [virtual]

Implements [gdcm::BaseRootQuery](#).

27.184.3.2 `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](#).

27.184.3.3 `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 dcmTk

Implements [gdcm::BaseRootQuery](#).

27.184.3.4 `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](#).

## 27.184.4 Friends And Related Function Documentation

### 27.184.4.1 friend class QueryFactory [friend]

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

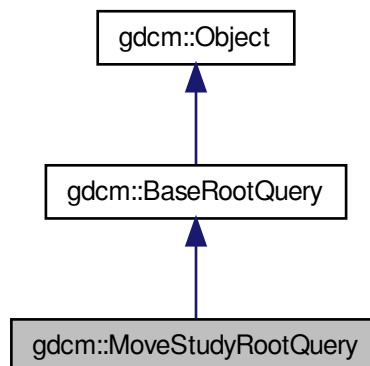
- [gdcmMovePatientRootQuery.h](#)

## 27.185 gdcm::MoveStudyRootQuery Class Reference

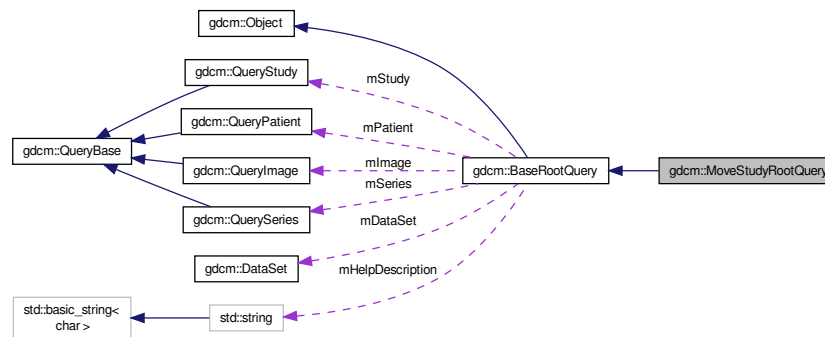
[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

```
#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

### 27.185.1 Detailed Description

[MoveStudyRootQuery](#) contains: the class which will produce a dataset for C-MOVE with study root.

### 27.185.2 Constructor & Destructor Documentation

27.185.2.1 `gdcm::MoveStudyRootQuery::MoveStudyRootQuery ( )`

### 27.185.3 Member Function Documentation

27.185.3.1 `UIDs::TSName gdcm::MoveStudyRootQuery::GetAbstractSyntaxUID ( ) const` `[virtual]`

Implements [gdcm::BaseRootQuery](#).

27.185.3.2 `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](#).

27.185.3.3 `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 dcmTk

Implements [gdcm::BaseRootQuery](#).

27.185.3.4 `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](#).

## 27.185.4 Friends And Related Function Documentation

27.185.4.1 `friend class QueryFactory` [friend]

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

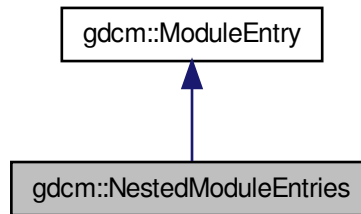
- [gdcmMoveStudyRootQuery.h](#)

## 27.186 gdcm::NestedModuleEntries Class Reference

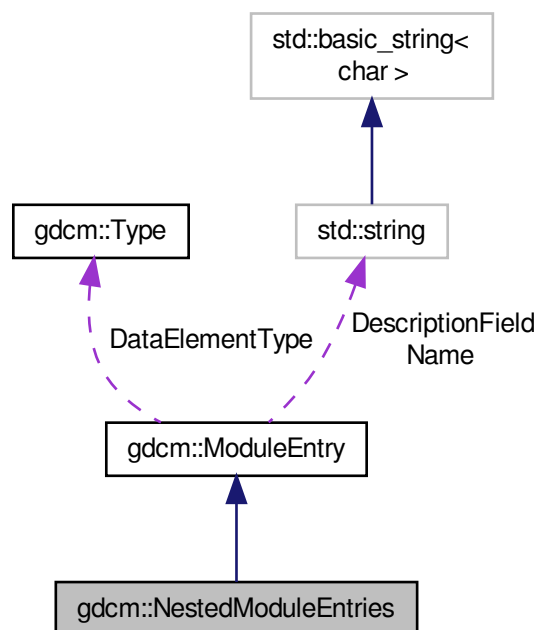
Class for representing a [NestedModuleEntries](#).

```
#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

### 27.186.1 Detailed Description

Class for representing a [NestedModuleEntries](#).

#### Note

bla

#### See also

[ModuleEntry](#)

### 27.186.2 Member Typedef Documentation

27.186.2.1 `typedef std::vector<ModuleEntry>::size_type gdcmm::NestedModuleEntries::SizeType`

### 27.186.3 Constructor & Destructor Documentation

27.186.3.1 `gdcmm::NestedModuleEntries::NestedModuleEntries ( const char * name = " ", const char * type = "3", const char * description = " " )` `[inline]`

### 27.186.4 Member Function Documentation

27.186.4.1 `void gdcmm::NestedModuleEntries::AddModuleEntry ( const ModuleEntry & me )` `[inline]`

27.186.4.2 `const ModuleEntry& gdcmm::NestedModuleEntries::GetModuleEntry ( SizeType idx ) const` `[inline]`

27.186.4.3 `ModuleEntry& gdcmm::NestedModuleEntries::GetModuleEntry ( SizeType idx )` `[inline]`

27.186.4.4 `SizeType gdcmm::NestedModuleEntries::GetNumberOfModuleEntries ( )` `[inline]`

### 27.186.5 Friends And Related Function Documentation

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

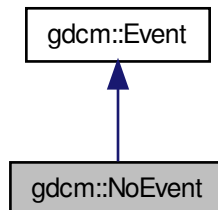
The documentation for this class was generated from the following file:

- [gdcmmNestedModuleEntries.h](#)

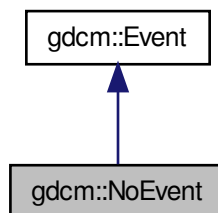
## 27.187 gdcM::NoEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::NoEvent:



Collaboration diagram for gdcM::NoEvent:



### Additional Inherited Members

#### 27.187.1 Detailed Description

Define some common GDCM events

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

- [gdcMEvent.h](#)

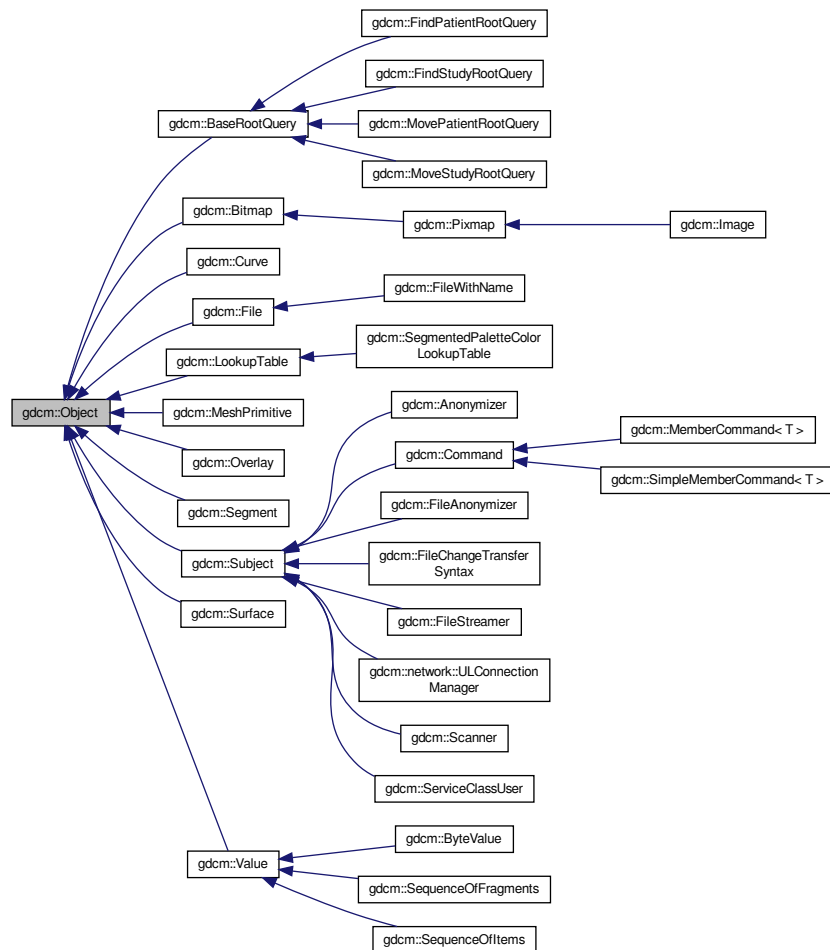
## 27.188 gdcM::Object Class Reference

[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`

### 27.188.1 Detailed Description

[Object](#).

#### Note

main superclass for object that want to use [SmartPointer](#) invasive ref counting system

#### See also

[SmartPointer](#)

### 27.188.2 Constructor & Destructor Documentation

27.188.2.1 `gdcM::Object::Object ( )` `[inline]`

27.188.2.2 `virtual gdcM::Object::~~Object ( )` `[inline], [virtual]`

27.188.2.3 `gdcM::Object::Object ( const Object & )` `[inline]`

Special requirement for copy/cstor, assignment operator.

### 27.188.3 Member Function Documentation

27.188.3.1 `void gdcM::Object::operator= ( const Object & )` `[inline]`

27.188.3.2 `virtual void gdcM::Object::Print ( std::ostream & ) const` `[inline], [virtual]`

Reimplemented in [gdcM::SequenceOfFragments](#), [gdcM::ByteValue](#), [gdcM::SequenceOfItems](#), [gdcM::BaseRootQuery](#), [gdcM::Scanner](#), [gdcM::Image](#), [gdcM::Curve](#), [gdcM::Overlay](#), [gdcM::Bitmap](#), [gdcM::LookupTable](#), [gdcM::Pixmap](#), and [gdcM::SegmentedPaletteColorLookupTable](#).

#### Examples:

[ReadAndDumpDICOMDIR.cxx](#).

Referenced by `gdcM::operator<<()`.

27.188.3.3 `void gdcM::Object::Register ( )` `[inline], [protected]`

27.188.3.4 `void gdcM::Object::UnRegister ( )` `[inline], [protected]`

### 27.188.4 Friends And Related Function Documentation

27.188.4.1 `std::ostream& operator<< ( std::ostream & os, const Object & obj )` [*friend*]

27.188.4.2 `template<class ObjectType > friend class SmartPointer` [*friend*]

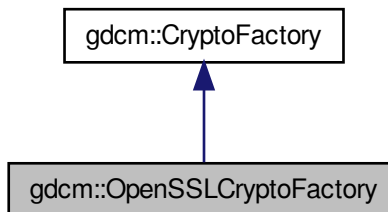
The documentation for this class was generated from the following file:

- [gdcmObject.h](#)

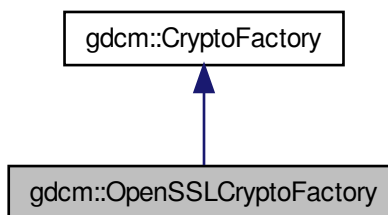
## 27.189 gdcm::OpenSSLCryptoFactory Class Reference

```
#include <gdcmOpenSSLCryptoFactory.h>
```

Inheritance diagram for gdcm::OpenSSLCryptoFactory:



Collaboration diagram for gdcm::OpenSSLCryptoFactory:



### Public Member Functions

- [OpenSSLCryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Protected Member Functions

- void [InitOpenSSL](#) ()

## Additional Inherited Members

### 27.189.1 Constructor & Destructor Documentation

27.189.1.1 `gdcmm::OpenSSLCryptoFactory::OpenSSLCryptoFactory ( CryptoLib id )` `[inline]`

References `gdcmDebugMacro`.

### 27.189.2 Member Function Documentation

27.189.2.1 `CryptographicMessageSyntax* gdcmm::OpenSSLCryptoFactory::CreateCMSProvider ( )` `[inline]`,  
`[virtual]`

Implements [gdcmm::CryptoFactory](#).

27.189.2.2 `void gdcmm::OpenSSLCryptoFactory::InitOpenSSL ( )` `[protected]`

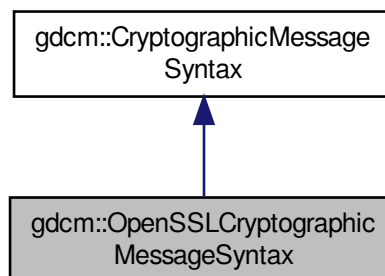
The documentation for this class was generated from the following file:

- [gdcmmOpenSSLCryptoFactory.h](#)

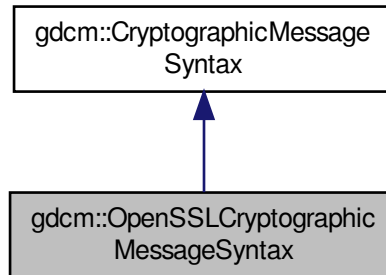
## 27.190 gdcmm::OpenSSLCryptographicMessageSyntax Class Reference

```
#include <gdcmmOpenSSLCryptographicMessageSyntax.h>
```

Inheritance diagram for `gdcmm::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

### 27.190.1 Constructor & Destructor Documentation

27.190.1.1 gdcM::OpenSSLCryptographicMessageSyntax::OpenSSLCryptographicMessageSyntax ( )

27.190.1.2 gdcM::OpenSSLCryptographicMessageSyntax::~~OpenSSLCryptographicMessageSyntax ( )

### 27.190.2 Member Function Documentation

27.190.2.1 bool gdcM::OpenSSLCryptographicMessageSyntax::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](#).

27.190.2.2 `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](#).

27.190.2.3 `CipherTypes gdcM::OpenSSLCryptographicMessageSyntax::GetCipherType ( ) const` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.190.2.4 `bool gdcM::OpenSSLCryptographicMessageSyntax::ParseCertificateFile ( const char * filename )` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.190.2.5 `bool gdcM::OpenSSLCryptographicMessageSyntax::ParseKeyFile ( const char * filename )` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

27.190.2.6 `void gdcM::OpenSSLCryptographicMessageSyntax::SetCipherType ( CipherTypes type )` [virtual]

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcM::CryptographicMessageSyntax](#).

27.190.2.7 `bool gdcM::OpenSSLCryptographicMessageSyntax::SetPassword ( const char * pass, size_t passLen )` [virtual]

Implements [gdcM::CryptographicMessageSyntax](#).

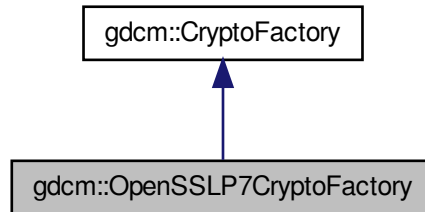
The documentation for this class was generated from the following file:

- [gdcMOpenSSLCryptographicMessageSyntax.h](#)

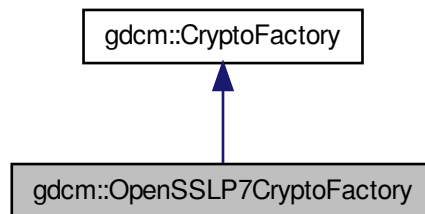
## 27.191 gdcM::OpenSSLP7CryptoFactory Class Reference

```
#include <gdcMOpenSSLP7CryptoFactory.h>
```

Inheritance diagram for gdcmm::OpenSSLP7CryptoFactory:



Collaboration diagram for gdcmm::OpenSSLP7CryptoFactory:



## Public Member Functions

- [OpenSSLP7CryptoFactory](#) ([CryptoLib](#) id)
- [CryptographicMessageSyntax](#) \* [CreateCMSProvider](#) ()

## Additional Inherited Members

### 27.191.1 Constructor & Destructor Documentation

27.191.1.1 `gdcmm::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory ( CryptoLib id )` `[inline]`

References `gdcmmDebugMacro`.

### 27.191.2 Member Function Documentation

27.191.2.1 **CryptographicMessageSyntax\*** `gdcmm::OpenSSLP7CryptoFactory::CreateCMSProvider ( )` `[inline]`,  
`[virtual]`

Implements [gdcmm::CryptoFactory](#).

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

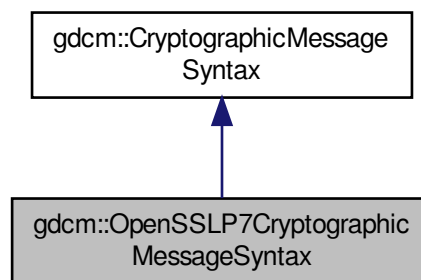
- [gdcmmOpenSSLP7CryptoFactory.h](#)

## 27.192 `gdcmm::OpenSSLP7CryptographicMessageSyntax` Class Reference

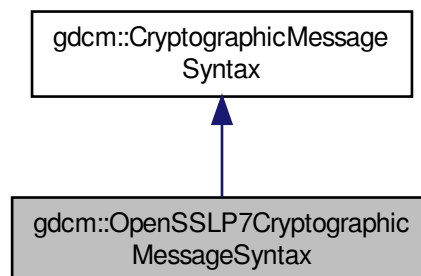
Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7\_encrypt functionalities.

```
#include <gdcmmOpenSSLP7CryptographicMessageSyntax.h>
```

Inheritance diagram for `gdcmm::OpenSSLP7CryptographicMessageSyntax`:



Collaboration diagram for `gdcmm::OpenSSLP7CryptographicMessageSyntax`:





## Public Member Functions

- [OpenSSLP7CryptographicMessageSyntax](#) ()
- [~OpenSSLP7CryptographicMessageSyntax](#) ()
- 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

### 27.192.1 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)

### 27.192.2 Constructor & Destructor Documentation

27.192.2.1 gdcmm::OpenSSLP7CryptographicMessageSyntax::OpenSSLP7CryptographicMessageSyntax ( )

27.192.2.2 gdcmm::OpenSSLP7CryptographicMessageSyntax::~~OpenSSLP7CryptographicMessageSyntax ( )

### 27.192.3 Member Function Documentation

27.192.3.1 bool gdcmm::OpenSSLP7CryptographicMessageSyntax::Decrypt ( char \* output, size\_t & outlen, const char \* array, size\_t len ) const [virtual]

decrypt content from a PKCS#7 envelopedData structure

Implements [gdcmm::CryptographicMessageSyntax](#).

27.192.3.2 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](#).

27.192.3.3 [CipherTypes](#) gdcmm::OpenSSLP7CryptographicMessageSyntax::GetCipherType ( ) const [virtual]

Implements [gdcmm::CryptographicMessageSyntax](#).

27.192.3.4 `bool gdcmm::OpenSSL7CryptographicMessageSyntax::ParseCertificateFile ( const char * filename )` [virtual]

Implements [gdcmm::CryptographicMessageSyntax](#).

27.192.3.5 `bool gdcmm::OpenSSL7CryptographicMessageSyntax::ParseKeyFile ( const char * filename )` [virtual]

Implements [gdcmm::CryptographicMessageSyntax](#).

27.192.3.6 `void gdcmm::OpenSSL7CryptographicMessageSyntax::SetCipherType ( CipherTypes type )` [virtual]

Set Cipher [Type](#). Default is: AES256\_CIPHER

Implements [gdcmm::CryptographicMessageSyntax](#).

27.192.3.7 `bool gdcmm::OpenSSL7CryptographicMessageSyntax::SetPassword ( const char *, size_t )` [inline],  
[virtual]

Implements [gdcmm::CryptographicMessageSyntax](#).

References `gdcmmWarningMacro`.

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

- [gdcmmOpenSSL7CryptographicMessageSyntax.h](#)

## 27.193 gdcmm::Orientation Class Reference

class to handle [Orientation](#)

```
#include <gdcmmOrientation.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)

### 27.193.1 Detailed Description

class to handle [Orientation](#)

### 27.193.2 Member Enumeration Documentation

#### 27.193.2.1 enum gdcm::Orientation::OrientationType

Enumerator

**UNKNOWN**

**AXIAL**

**CORONAL**

**SAGITTAL**

**OBLIQUE**

### 27.193.3 Constructor & Destructor Documentation

#### 27.193.3.1 gdcm::Orientation::Orientation ( )

#### 27.193.3.2 gdcm::Orientation::~~Orientation ( )

### 27.193.4 Member Function Documentation

#### 27.193.4.1 static const char\* gdcm::Orientation::GetLabel ( [OrientationType](#) type ) [static]

Return the label of an [Orientation](#).

27.193.4.2 `static char gdcm::Orientation::GetMajorAxisFromPatientRelativeDirectionCosine ( double x, double y, double z )`  
`[static], [protected]`

27.193.4.3 `static double gdcm::Orientation::GetObliquityThresholdCosineValue ( )` `[static]`

27.193.4.4 `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

27.193.4.5 `void gdcm::Orientation::Print ( std::ostream & ) const`

Print.

Referenced by `gdcm::operator<<()`.

27.193.4.6 `static void gdcm::Orientation::SetObliquityThresholdCosineValue ( double val )` `[static]`

ObliquityThresholdCosineValue stuff.

## 27.193.5 Friends And Related Function Documentation

27.193.5.1 `std::ostream& operator<< ( std::ostream &_os, const Orientation &o )` `[friend]`

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

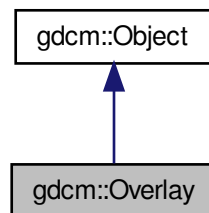
- [gdcmOrientation.h](#)

## 27.194 gdcm::Overlay Class Reference

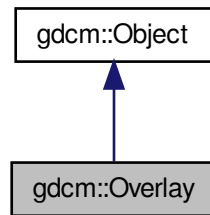
[Overlay](#) class.

```
#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 not the OverlayData is in the Pixel Data:*
- bool [IsZero](#) () const
 

*return true if all bits are set to 0*
- 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

### 27.194.1 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:

### 27.194.2 Member Enumeration Documentation

#### 27.194.2.1 enum gdcm::Overlay::OverlayType

Enumerator

***Invalid***  
***Graphics***  
***ROI***

### 27.194.3 Constructor & Destructor Documentation

27.194.3.1 gdcm::Overlay::Overlay ( )

27.194.3.2 gdcm::Overlay::~~Overlay ( )

27.194.3.3 gdcm::Overlay::Overlay ( Overlay const & ov )

### 27.194.4 Member Function Documentation

27.194.4.1 void gdcm::Overlay::Decompress ( std::ostream & os ) const

Decode the internal OverlayData (packed bits) into unpacked representation.

27.194.4.2 unsigned short gdcm::Overlay::GetBitPosition ( ) const

return bit position

27.194.4.3 unsigned short gdcm::Overlay::GetBitsAllocated ( ) const

return bits allocated

27.194.4.4 unsigned short gdcm::Overlay::GetColumns ( ) const

get columns

27.194.4.5 `const char* gdcm::Overlay::GetDescription ( ) const`

get description

27.194.4.6 `unsigned short gdcm::Overlay::GetGroup ( ) const`

Get Group number.

27.194.4.7 `const signed short* gdcm::Overlay::GetOrigin ( ) const`

get origin

27.194.4.8 `const ByteValue& gdcm::Overlay::GetOverlayData ( ) const`

Return the [Overlay](#) Data as [ByteValue](#): Not thread safe

27.194.4.9 `static const char* gdcm::Overlay::GetOverlayTypeAsString ( OverlayType ot ) [static]`

27.194.4.10 `static OverlayType gdcm::Overlay::GetOverlayTypeFromString ( const char * ) [static]`

27.194.4.11 `unsigned short gdcm::Overlay::GetRows ( ) const`

get rows

27.194.4.12 `const char* gdcm::Overlay::GetType ( ) const`

get type

27.194.4.13 `OverlayType gdcm::Overlay::GetTypeAsEnum ( ) const`

27.194.4.14 `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\(\)](#)

27.194.4.15 `size_t gdcm::Overlay::GetUnpackBufferLength ( ) const`

Retrieve the size of the buffer needed to hold the [Overlay](#) as specified by Col & Row parameters

27.194.4.16 `bool gdcm::Overlay::GrabOverlayFromPixelData ( DataSet const & ds )`

27.194.4.17 `bool gdcm::Overlay::IsEmpty ( ) const`

Return whether or not the [Overlay](#) is empty:

27.194.4.18 `bool gdcm::Overlay::IsInPixelData ( ) const`

return if the [Overlay](#) is stored in the pixel data or not



27.194.4.19 void gdcmm::Overlay::IsInPixelData ( bool *b* )

Set whether or no the OverlayData is in the Pixel Data:

27.194.4.20 bool gdcmm::Overlay::IsZero ( ) const

return true if all bits are set to 0

27.194.4.21 void gdcmm::Overlay::Print ( std::ostream & ) const [virtual]

Print.

Reimplemented from [gdcmm::Object](#).

27.194.4.22 void gdcmm::Overlay::SetBitPosition ( unsigned short *bitposition* )

set bit position

27.194.4.23 void gdcmm::Overlay::SetBitsAllocated ( unsigned short *bitsallocated* )

set bits allocated

27.194.4.24 void gdcmm::Overlay::SetColumns ( unsigned short *columns* )

set columns

27.194.4.25 void gdcmm::Overlay::SetDescription ( const char \* *description* )

set description

27.194.4.26 void gdcmm::Overlay::SetFrameOrigin ( unsigned short *frameorigin* )

set frame origin

27.194.4.27 void gdcmm::Overlay::SetGroup ( unsigned short *group* )

Set Group number.

27.194.4.28 void gdcmm::Overlay::SetNumberOfFrames ( unsigned int *numberofframes* )

set number of frames

27.194.4.29 void gdcmm::Overlay::SetOrigin ( const signed short *origin*[2] )

set origin

27.194.4.30 void `gdcm::Overlay::SetOverlay` ( const char \* *array*, size\_t *length* )

set overlay from byte array + length

27.194.4.31 void `gdcm::Overlay::SetRows` ( unsigned short *rows* )

set rows

27.194.4.32 void `gdcm::Overlay::SetType` ( const char \* *type* )

set type

27.194.4.33 void `gdcm::Overlay::Update` ( const `DataElement` & *de* )

Update overlay from data element *de*:

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

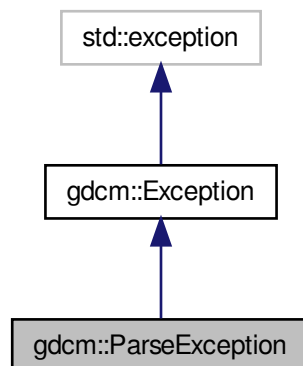
- [gdcmOverlay.h](#)

## 27.195 `gdcm::ParseException` Class Reference

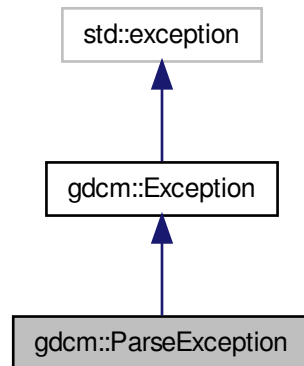
[ParseException](#) Standard exception handling object.

```
#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)

## 27.195.1 Detailed Description

[ParseException](#) Standard exception handling object.

## 27.195.2 Constructor & Destructor Documentation

27.195.2.1 `gdcm::ParseException::ParseException ( )` `[inline]`

27.195.2.2 `virtual gdcm::ParseException::~~ParseException ( ) throw ()` `[inline], [virtual]`

## 27.195.3 Member Function Documentation

27.195.3.1 `const DataElement& gdcm::ParseException::GetLastElement ( ) const` `[inline]`

27.195.3.2 `ParseException& gdcm::ParseException::operator= ( const ParseException & orig )` `[inline]`

Assignment operator.

27.195.3.3 void `gdcmm::ParseException::SetLastElement ( DataElement & de )` `[inline]`

Equivalence operator.

Referenced by `gdcmm::Fragment::ReadBacktrack()`, and `gdcmm::Fragment::ReadValue()`.

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

- [gdcmmParseException.h](#)

## 27.196 gdcmm::Parser Class Reference

[Parser](#) ala XML\_Parser from expat (SAX)

```
#include <gdcmmParser.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](#) ()

### 27.196.1 Detailed Description

[Parser](#) ala XML\_Parser from expat (SAX)

Detailed description here

Note

Simple API for DICOM

### 27.196.2 Member Typedef Documentation

27.196.2.1 `typedef void(* gdcm::Parser::EndElementHandler)(void *userData, const Tag &name)`

27.196.2.2 `typedef void(* gdcm::Parser::StartElementHandler)(void *userData, const Tag &tag, const char *atts[])`

### 27.196.3 Member Enumeration Documentation

27.196.3.1 `enum gdcm::Parser::ErrorType`

Enumerator

***NoError***

***NoMemoryError***

***SyntaxError***

***NoElementsError***

***TagMismatchError***

***DuplicateAttributeError***

***JunkAfterDocElementError***

***UndefinedEntityError***

***UnexpectedStateError***

### 27.196.4 Constructor & Destructor Documentation

27.196.4.1 `gdcm::Parser::Parser ( )` `[inline]`

27.196.4.2 `gdcm::Parser::~~Parser ( )` `[inline]`

### 27.196.5 Member Function Documentation

27.196.5.1 `char* gdcm::Parser::GetBuffer ( int len )` `[protected]`

27.196.5.2 `unsigned long gdcm::Parser::GetCurrentByteIndex ( )` `const`

27.196.5.3 `ErrorType gdcm::Parser::GetErrorCode ( )` `const`

27.196.5.4 `static const char* gdcm::Parser::GetErrorString ( ErrorType const & err )` `[static]`

27.196.5.5 `void* gdcm::Parser::GetUserData ( )` `const`

27.196.5.6 `bool gdcM::Parser::Parse ( const char * s, int len, bool isFinal )`

27.196.5.7 `bool gdcM::Parser::ParseBuffer ( int len, bool isFinal )` [protected]

27.196.5.8 `ErrorType gdcM::Parser::Process ( )` [protected]

27.196.5.9 `void gdcM::Parser::SetElementHandler ( StartElementHandler start, EndElementHandler end )`

27.196.5.10 `void gdcM::Parser::SetUserData ( void * userData )`

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

- [gdcMParser.h](#)

## 27.197 gdcM::Patient Class Reference

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

```
#include <gdcMPatient.h>
```

### Public Member Functions

- [Patient \(\)](#)

#### 27.197.1 Detailed Description

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

#### 27.197.2 Constructor & Destructor Documentation

27.197.2.1 `gdcM::Patient::Patient ( )` [inline]

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

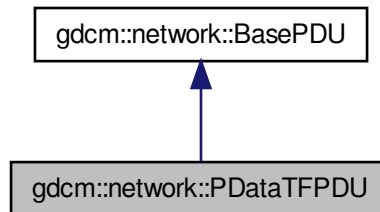
- [gdcMPatient.h](#)

## 27.198 gdcM::network::PDataTFPDU Class Reference

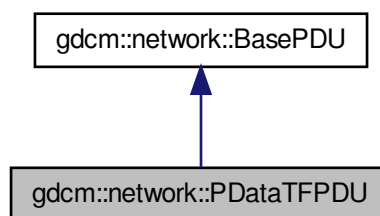
[PDataTFPDU Table](#) 9-22 P-DATA-TF PDU FIELDS.

```
#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)`

### 27.198.1 Detailed Description

[PDataTFPDU Table](#) 9-22 P-DATA-TF PDU FIELDS.

### 27.198.2 Member Typedef Documentation

27.198.2.1 `typedef std::vector<PresentationDataValue>::size_type gdcmm::network::PDataTFPDU::SizeType`

### 27.198.3 Constructor & Destructor Documentation

27.198.3.1 `gdcmm::network::PDataTFPDU::PDataTFPDU ( )`

### 27.198.4 Member Function Documentation

27.198.4.1 `void gdcmm::network::PDataTFPDU::AddPresentationDataValue ( PresentationDataValue const & pdv )`  
[inline]

27.198.4.2 `SizeType gdcmm::network::PDataTFPDU::GetNumberOfPresentationDataValues ( ) const` [inline]

27.198.4.3 `PresentationDataValue const& gdcmm::network::PDataTFPDU::GetPresentationDataValue ( SizeType i ) const`  
[inline]

27.198.4.4 `bool gdcmm::network::PDataTFPDU::IsLastFragment ( ) const` [virtual]

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

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

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

27.198.4.6 `std::istream& gdcmm::network::PDataTFPDU::Read ( std::istream & is )` [virtual]

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

27.198.4.7 `std::istream& gdcmm::network::PDataTFPDU::ReadInto ( std::istream & is, std::ostream & os )` [protected]

27.198.4.8 `size_t gdcmm::network::PDataTFPDU::Size ( ) const` [virtual]

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

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

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

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



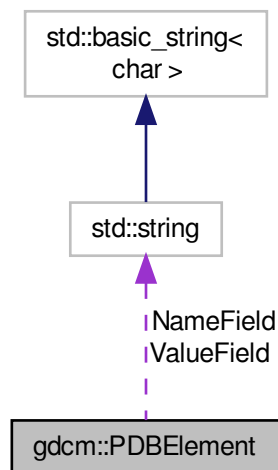
- [gdcmPDataTFPDU.h](#)

## 27.199 gdcm::PDBElement Class Reference

Class to represent a PDB [Element](#).

```
#include <gdcmPDBElement.h>
```

Collaboration diagram for gdcm::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)`

### 27.199.1 Detailed Description

Class to represent a PDB [Element](#).

See also

[PDBHeader](#)

### 27.199.2 Constructor & Destructor Documentation

27.199.2.1 `gdcm::PDBElement::PDBElement ( )` `[inline]`

### 27.199.3 Member Function Documentation

27.199.3.1 `const char* gdcm::PDBElement::GetName ( ) const` `[inline]`

Set/Get Name.

27.199.3.2 `const char* gdcm::PDBElement::GetValue ( ) const` `[inline]`

Set/Get [Value](#).

27.199.3.3 `bool gdcm::PDBElement::operator== ( const PDBElement &de ) const` `[inline]`

References [NameField](#), and [ValueField](#).

27.199.3.4 `void gdcm::PDBElement::SetName ( const char * name )` `[inline]`

27.199.3.5 `void gdcm::PDBElement::SetValue ( const char * value )` `[inline]`

### 27.199.4 Friends And Related Function Documentation

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

### 27.199.5 Member Data Documentation

27.199.5.1 `std::string gdcm::PDBElement::NameField` `[protected]`

Referenced by `gdcm::operator<<()`, and `operator==()`.

27.199.5.2 `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](#)

## 27.200 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)

### 27.200.1 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](#)

## 27.200.2 Constructor & Destructor Documentation

27.200.2.1 `gdcm::PDBHeader::PDBHeader ( )` `[inline]`

27.200.2.2 `gdcm::PDBHeader::~~PDBHeader ( )` `[inline]`

## 27.200.3 Member Function Documentation

27.200.3.1 `bool gdcm::PDBHeader::FindPDBElementByName ( const char * name )`

Return true if the PDB element matching name is found or not.

27.200.3.2 `const PDBElement& gdcm::PDBHeader::GetPDBEEnd ( ) const` `[protected]`

27.200.3.3 `const PDBElement& gdcm::PDBHeader::GetPDBElementByName ( const char * name )`

Lookup in the PDB header if a PDB element match the name 'name':

Warning

Case Sensitive

27.200.3.4 `static const PrivateTag& gdcm::PDBHeader::GetPDBInfoTag ( )` `[static]`

Return the Private [Tag](#) where the PDB header is stored within a DICOM [DataSet](#).

27.200.3.5 `bool gdcm::PDBHeader::LoadFromDataElement ( DataElement const & de )`

Load the PDB Header from a [DataElement](#) of a [DataSet](#).

27.200.3.6 `void gdcm::PDBHeader::Print ( std::ostream & os ) const`

Print.

Referenced by `gdcm::operator<<()`.

## 27.200.4 Friends And Related Function Documentation

27.200.4.1 `std::ostream& operator<< ( std::ostream & _os, const PDBHeader & d )` `[friend]`

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

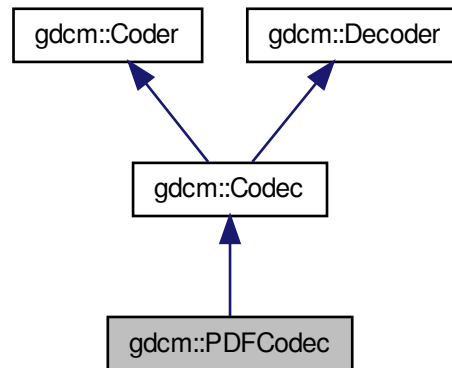
- [gdcmPDBHeader.h](#)

## 27.201 gdcm::PDFCodec Class Reference

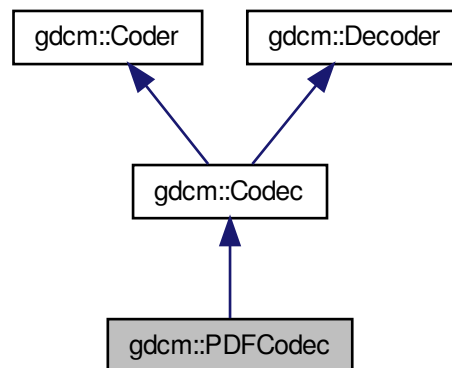
[PDFCodec](#) class.

```
#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

### 27.201.1 Detailed Description

[PDFCodec](#) class.

### 27.201.2 Constructor & Destructor Documentation

27.201.2.1 `gdcm::PDFCodec::PDFCodec ( )`

27.201.2.2 `gdcm::PDFCodec::~~PDFCodec ( )`

### 27.201.3 Member Function Documentation

27.201.3.1 `bool gdcm::PDFCodec::CanCode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this coder support this transfer syntax (can code it)

Implements [gdcm::Coder](#).

27.201.3.2 `bool gdcm::PDFCodec::CanDecode ( TransferSyntax const & ) const` `[inline], [virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Implements [gdcm::Decoder](#).

27.201.3.3 `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](#)

## 27.202 gdcm::network::PDUFactory Class Reference

[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.

```
#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)
- static std::vector< [BasePDU](#) \* > [CreateCStoreRSPPDU](#) (const [DataSet](#) \*inDataSet, const [BasePDU](#) \*inPC)
- static [EEventID](#) [DetermineEventByPDU](#) (const [BasePDU](#) \*inPDU)
- static std::vector< [PresentationDataValue](#) > [GetPDVs](#) (const std::vector< [BasePDU](#) \* > &inDataPDUs)

### 27.202.1 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.

### 27.202.2 Member Function Documentation

- 27.202.2.1 static [BasePDU](#)\* [gdcmm::network::PDUFactory::ConstructAbortPDU](#) ( ) [static]
- 27.202.2.2 static [BasePDU](#)\* [gdcmm::network::PDUFactory::ConstructPDU](#) ( uint8\_t itemtype ) [static]
- 27.202.2.3 static [BasePDU](#)\* [gdcmm::network::PDUFactory::ConstructReleasePDU](#) ( ) [static]
- 27.202.2.4 static std::vector<[BasePDU](#)\*> [gdcmm::network::PDUFactory::CreateCEchoPDU](#) ( const [ULConnection](#) &inConnection ) [static]
- 27.202.2.5 static std::vector<[BasePDU](#)\*> [gdcmm::network::PDUFactory::CreateCFindPDU](#) ( const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \* inRootQuery ) [static]
- 27.202.2.6 static std::vector<[BasePDU](#)\*> [gdcmm::network::PDUFactory::CreateCMovePDU](#) ( const [ULConnection](#) &inConnection, const [BaseRootQuery](#) \* inRootQuery ) [static]
- 27.202.2.7 static std::vector<[BasePDU](#)\*> [gdcmm::network::PDUFactory::CreateCStoreRQPDU](#) ( const [ULConnection](#) &inConnection, const [File](#) & file ) [static]
- 27.202.2.8 static std::vector<[BasePDU](#)\*> [gdcmm::network::PDUFactory::CreateCStoreRSPPDU](#) ( const [DataSet](#) \* inDataSet, const [BasePDU](#) \* inPC ) [static]
- 27.202.2.9 static [EEventID](#) [gdcmm::network::PDUFactory::DetermineEventByPDU](#) ( const [BasePDU](#) \* inPDU ) [static]
- 27.202.2.10 static std::vector<[PresentationDataValue](#)> [gdcmm::network::PDUFactory::GetPDVs](#) ( const std::vector< [BasePDU](#) \* > & inDataPDUs ) [static]

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

- [gdcmmPDUFactory.h](#)

## 27.203 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](#) = '^'

### 27.203.1 Detailed Description

[PersonName](#) class.

### 27.203.2 Member Function Documentation

27.203.2.1 unsigned int gdcm::PersonName::GetMaxLength ( ) const [\[inline\]](#)

27.203.2.2 unsigned int gdcm::PersonName::GetNumberOfComponents ( ) const [\[inline\]](#)

27.203.2.3 void gdcm::PersonName::Print ( std::ostream & os ) const [\[inline\]](#)

27.203.2.4 void gdcm::PersonName::SetBlob ( const std::vector< char > & v ) [\[inline\]](#)

27.203.2.5 void gdcm::PersonName::SetComponents ( const char \* comp1 = " ", const char \* comp2 = " ", const char \* comp3 = " ", const char \* comp4 = " ", const char \* comp5 = " " ) [\[inline\]](#)

27.203.2.6 void gdcm::PersonName::SetComponents ( const char \* components[] ) [\[inline\]](#)

### 27.203.3 Member Data Documentation



27.203.3.1 `char gdcm::PersonName::Component[MaxNumberOfComponents][MaxLength+1]`

27.203.3.2 `const unsigned int gdcm::PersonName::MaxLength = 64` `[static]`

27.203.3.3 `const unsigned int gdcm::PersonName::MaxNumberOfComponents = 5` `[static]`

27.203.3.4 `const char gdcm::PersonName::Padding = ''` `[static]`

27.203.3.5 `const char gdcm::PersonName::Separator = '^'` `[static]`

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

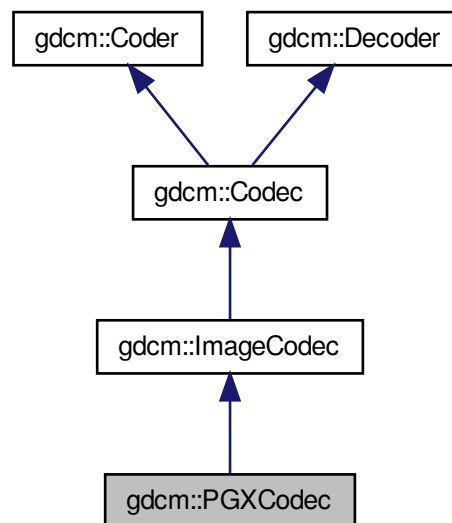
- [gdcmPersonName.h](#)

## 27.204 gdcm::PGXCodec Class Reference

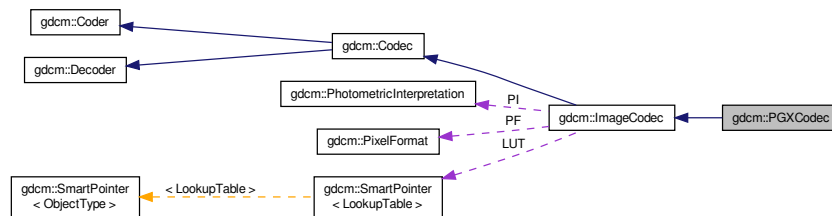
Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

```
#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

### 27.204.1 Detailed Description

Class to do PGX See PGX as used in JPEG 2000 implementation and reference images.

### 27.204.2 Constructor & Destructor Documentation

27.204.2.1 `gdcm::PGXCodec::PGXCodec ( )`

27.204.2.2 `gdcm::PGXCodec::~~PGXCodec ( )`

### 27.204.3 Member Function Documentation

27.204.3.1 `bool gdcm::PGXCodec::CanCode ( TransferSyntax const & ) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.204.3.2 `bool gdcm::PGXCodec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.204.3.3 `virtual ImageCodec* gdcm::PGXCodec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

27.204.3.4 `bool gdcm::PGXCodec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcm::ImageCodec](#).

27.204.3.5 `bool gdcm::PGXCodec::Read ( const char * filename, DataElement & out ) const`

27.204.3.6 `bool gdcm::PGXCodec::Write ( const char * filename, const DataElement & out ) const`

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

- [gdcmPGXCodec.h](#)

## 27.205 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
- [PIType](#) [GetType](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsSameColorSpace](#) ([PhotometricInterpretation](#) const &pi) const
- [operator PIType](#) () const

## Static Public Member Functions

- static const char \* [GetPIString](#) ([PIType](#) pi)
- static [PIType](#) [GetPIType](#) (const char \*pi)
- static bool [IsRetired](#) ([PIType](#) pi)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [PhotometricInterpretation](#) &pi)

## 27.205.1 Detailed Description

Class to represent an [PhotometricInterpretation](#).

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [HelloVizWorld.cxx](#), and [iU22tomultisc.cxx](#).

## 27.205.2 Member Enumeration Documentation

### 27.205.2.1 enum gdcm::PhotometricInterpretation::PIType

Enumerator

**UNKNOWN**  
**MONOCHROME1**  
**MONOCHROME2**  
**PALETTE\_COLOR**  
**RGB**  
**HSV**  
**ARGB**  
**CMYK**  
**YBR\_FULL**  
**YBR\_FULL\_422**  
**YBR\_PARTIAL\_422**  
**YBR\_PARTIAL\_420**  
**YBR\_ICT**  
**YBR\_RCT**  
**PI\_END**

### 27.205.3 Constructor & Destructor Documentation

27.205.3.1 `gdcm::PhotometricInterpretation::PhotometricInterpretation ( PType pi = UNKNOWN )` `[inline]`

### 27.205.4 Member Function Documentation

27.205.4.1 `static const char* gdcm::PhotometricInterpretation::GetPIString ( PType pi )` `[static]`

Referenced by `gdcm::operator<<()`.

27.205.4.2 `static PType gdcm::PhotometricInterpretation::GetPType ( const char * pi )` `[static]`

27.205.4.3 `unsigned short gdcm::PhotometricInterpretation::GetSamplesPerPixel ( )` `const`

return the value for Sample Per Pixel associated with a particular Photometric Interpretation

27.205.4.4 `const char* gdcm::PhotometricInterpretation::GetString ( )` `const`

27.205.4.5 `PType gdcm::PhotometricInterpretation::GetType ( )` `const` `[inline]`

27.205.4.6 `bool gdcm::PhotometricInterpretation::IsLossless ( )` `const`

27.205.4.7 `bool gdcm::PhotometricInterpretation::IsLossy ( )` `const`

27.205.4.8 `static bool gdcm::PhotometricInterpretation::IsRetired ( PType pi )` `[static]`

27.205.4.9 `bool gdcm::PhotometricInterpretation::IsSameColorSpace ( PhotometricInterpretation const & pi )` `const`

27.205.4.10 `gdcm::PhotometricInterpretation::operator PType ( )` `const` `[inline]`

### 27.205.5 Friends And Related Function Documentation

27.205.5.1 `std::ostream& operator<< ( std::ostream & os, const PhotometricInterpretation & pi )` `[friend]`

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

- [gdcmPhotometricInterpretation.h](#)

## 27.206 gdcm::PixelFormat Class Reference

[PixelFormat](#).

```
#include <gdcmPixelFormat.h>
```

## Public Types

- enum [ScalarType](#) {  
[UINT8](#),  
[INT8](#),  
[UINT12](#),  
[INT12](#),  
[UINT16](#),  
[INT16](#),  
[UINT32](#),  
[INT32](#),  
[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 [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)

### 27.206.1 Detailed Description

[PixelFormat](#).

#### Note

By default the Pixel [Type](#) will be instantiated with the following parameters:

- SamplesPerPixel : 1
- BitsAllocated : 8
- BitsStored : 8
- HighBit : 7
- PixelRepresentation : 0

#### Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [csa2img.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GetJPEGSamplePrecision.cxx](#), [iU22tomultisc.cxx](#), and [threadgdcm.cxx](#).

### 27.206.2 Member Enumeration Documentation

#### 27.206.2.1 enum gdcm::PixelFormat::ScalarType

##### Enumerator

**UINT8**

**INT8**

**UINT12**

**INT12**

**UINT16**

**INT16**

**UINT32**

**INT32**

**FLOAT16**

***FLOAT32******FLOAT64******SINGLEBIT******UNKNOWN***

### 27.206.3 Constructor & Destructor Documentation

27.206.3.1 `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]`

27.206.3.2 `gdcm::PixelFormat::PixelFormat ( ScalarType st )`

### 27.206.4 Member Function Documentation

27.206.4.1 `unsigned short gdcm::PixelFormat::GetBitsAllocated ( ) const` `[inline]`

BitsAllocated see [Tag](#) (0028,0100) US Bits Allocated.

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.206.4.2 `unsigned short gdcm::PixelFormat::GetBitsStored ( ) const` `[inline]`

BitsStored see [Tag](#) (0028,0101) US Bits Stored.

Examples:

[GetJPEGSamplePrecision.cxx](#).

27.206.4.3 `unsigned short gdcm::PixelFormat::GetHighBit ( ) const` `[inline]`

HighBit see [Tag](#) (0028,0102) US High Bit.

27.206.4.4 `int64_t gdcm::PixelFormat::GetMax ( ) const`

return the max possible of the pixel

27.206.4.5 `int64_t gdcm::PixelFormat::GetMin ( ) const`

return the min possible of the pixel

27.206.4.6 `unsigned short gdcm::PixelFormat::GetPixelRepresentation ( ) const` `[inline]`

PixelRepresentation: 0 or 1, see [Tag](#) (0028,0103) US Pixel Representation.



#### 27.206.4.7 `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](#).

#### 27.206.4.8 `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](#).

#### 27.206.4.9 `ScalarType gdcm::PixelFormat::GetScalarType ( ) const`

ScalarType does not take into account the sample per pixel.

#### 27.206.4.10 `const char* gdcm::PixelFormat::GetScalarTypeAsString ( ) const`

#### 27.206.4.11 `bool gdcm::PixelFormat::IsValid ( ) const`

return IsValid

#### 27.206.4.12 `gdcm::PixelFormat::operator ScalarType ( ) const` `[inline]`

#### 27.206.4.13 `bool gdcm::PixelFormat::operator!= ( ScalarType st ) const` `[inline]`

#### 27.206.4.14 `bool gdcm::PixelFormat::operator!= ( const PixelFormat & pf ) const` `[inline]`

#### 27.206.4.15 `bool gdcm::PixelFormat::operator== ( ScalarType st ) const` `[inline]`

#### 27.206.4.16 `bool gdcm::PixelFormat::operator== ( const PixelFormat & pf ) const` `[inline]`

#### 27.206.4.17 `void gdcm::PixelFormat::Print ( std::ostream & os ) const`

Print.

Referenced by `gdcm::operator<<()`.

27.206.4.18 void gdcM::PixelFormat::SetBitsAllocated ( unsigned short *ba* ) [inline]

27.206.4.19 void gdcM::PixelFormat::SetBitsStored ( unsigned short *bs* ) [inline]

27.206.4.20 void gdcM::PixelFormat::SetHighBit ( unsigned short *hb* ) [inline]

27.206.4.21 void gdcM::PixelFormat::SetPixelRepresentation ( unsigned short *pr* ) [inline]

27.206.4.22 void gdcM::PixelFormat::SetSamplesPerPixel ( unsigned short *spp* ) [inline]

Examples:

[CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), and [GenFakeImage.cxx](#).

References [gdcMAssertMacro](#).

27.206.4.23 void gdcM::PixelFormat::SetScalarType ( ScalarType *st* )

Set [PixelFormat](#) based only on the ScalarType

Warning

: You need to call SetScalarType *before* SetSamplesPerPixel

27.206.4.24 bool gdcM::PixelFormat::Validate ( ) [protected]

When image with 24/24/23 was read, need to validate.

Referenced by [gdcM::Bitmap::SetPixelFormat\(\)](#).

## 27.206.5 Friends And Related Function Documentation

27.206.5.1 friend class [Bitmap](#) [friend]

27.206.5.2 std::ostream& operator<< ( std::ostream &\_os, const [PixelFormat](#) &*pf* ) [friend]

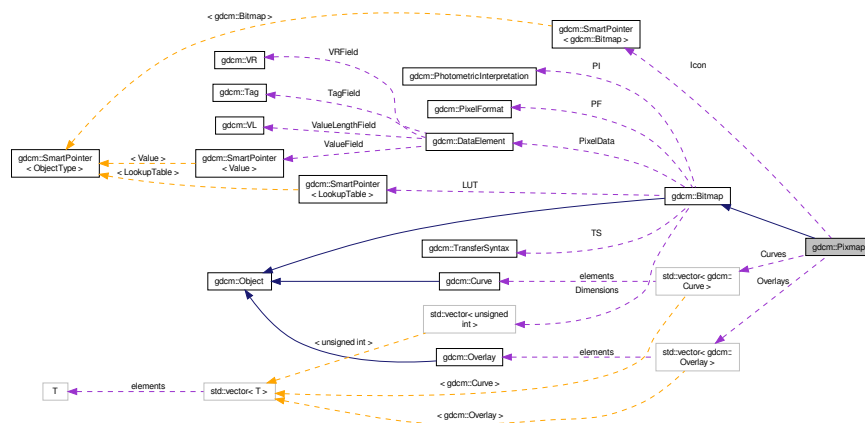
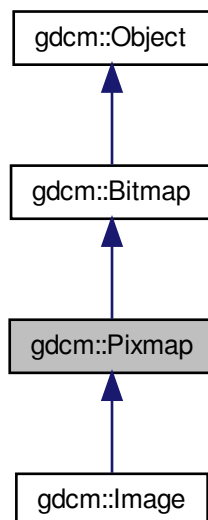
The documentation for this class was generated from the following file:

- [gdcMPixelFormat.h](#)

## 27.207 gdcM::Pixmap Class Reference

[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)

```
#include <gdcMPixmap.h>
```



- `Pixmap ()`
- `~Pixmap ()`
- `bool AreOverlaysInPixelData () const`  
*returns if Overlays are stored in the unused bit of the pixel data:*

- [Curve](#) & [GetCurve](#) (size\_t i=0)  
*Curve: group 50xx.*
- const [Curve](#) & [GetCurve](#) (size\_t i=0) const
- const [IconImage](#) & [GetIconImage](#) () const  
*Set/Get Icon Image.*
- [IconImage](#) & [GetIconImage](#) ()
- size\_t [GetNumberOfCurves](#) () const
- size\_t [GetNumberOfOverlays](#) () const
- [Overlay](#) & [GetOverlay](#) (size\_t i=0)  
*Overlay: group 60xx.*
- const [Overlay](#) & [GetOverlay](#) (size\_t i=0) const
- void [Print](#) (std::ostream &) const
- void [RemoveOverlay](#) (size\_t i)
- void [SetIconImage](#) ([IconImage](#) const &ii)
- void [SetNumberOfCurves](#) (size\_t n)
- void [SetNumberOfOverlays](#) (size\_t n)

### Protected Attributes

- std::vector< [Curve](#) > [Curves](#)
- [SmartPointer](#)< [IconImage](#) > [Icon](#)
- std::vector< [Overlay](#) > [Overlays](#)

### Additional Inherited Members

#### 27.207.1 Detailed Description

[Pixmap](#) class A bitmap based image. Used as parent for both [IconImage](#) and the main Pixel Data [Image](#) It does not contains any World Space information (IPP, IOP)

See also

[PixmapReader](#)

Examples:

[FileChangeTS.cs](#), and [StandardizeFiles.cs](#).

#### 27.207.2 Constructor & Destructor Documentation

27.207.2.1 `gdcm::Pixmap::Pixmap ( )`

27.207.2.2 `gdcm::Pixmap::~~Pixmap ( )`

#### 27.207.3 Member Function Documentation

27.207.3.1 `bool gdcm::Pixmap::AreOverlaysInPixelData ( ) const` [virtual]

returns if Overlays are stored in the unused bit of the pixel data:

Reimplemented from [gdcm::Bitmap](#).

27.207.3.2 **Curve&** gdcm::Pixmap::GetCurve ( size\_t *i* = 0 ) [inline]

[Curve](#): group 50xx.

27.207.3.3 **const Curve&** gdcm::Pixmap::GetCurve ( size\_t *i* = 0 ) const [inline]

27.207.3.4 **const IconImage&** gdcm::Pixmap::GetIconImage ( ) const [inline]

Set/Get Icon [Image](#).

27.207.3.5 **IconImage&** gdcm::Pixmap::GetIconImage ( ) [inline]

27.207.3.6 **size\_t** gdcm::Pixmap::GetNumberOfCurves ( ) const [inline]

27.207.3.7 **size\_t** gdcm::Pixmap::GetNumberOfOverlays ( ) const [inline]

27.207.3.8 **Overlay&** gdcm::Pixmap::GetOverlay ( size\_t *i* = 0 ) [inline]

[Overlay](#): group 60xx.

27.207.3.9 **const Overlay&** gdcm::Pixmap::GetOverlay ( size\_t *i* = 0 ) const [inline]

27.207.3.10 **void** gdcm::Pixmap::Print ( std::ostream & ) const [virtual]

Reimplemented from [gdcm::Bitmap](#).

27.207.3.11 **void** gdcm::Pixmap::RemoveOverlay ( size\_t *i* ) [inline]

27.207.3.12 **void** gdcm::Pixmap::SetIconImage ( IconImage const & *ii* ) [inline]

27.207.3.13 **void** gdcm::Pixmap::SetNumberOfCurves ( size\_t *n* ) [inline]

27.207.3.14 **void** gdcm::Pixmap::SetNumberOfOverlays ( size\_t *n* ) [inline]

## 27.207.4 Member Data Documentation

27.207.4.1 **std::vector<Curve>** gdcm::Pixmap::Curves [protected]

27.207.4.2 **SmartPointer<IconImage>** gdcm::Pixmap::Icon [protected]

27.207.4.3 **std::vector<Overlay>** gdcm::Pixmap::Overlays [protected]

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

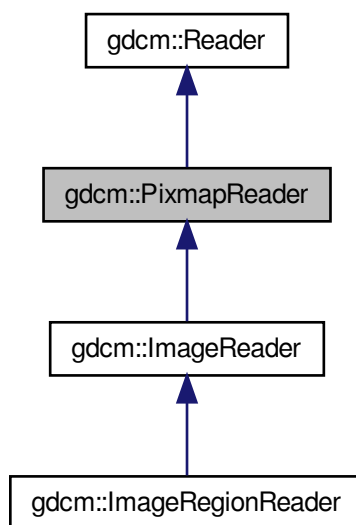
- [gdcmPixmap.h](#)

## 27.208 gdcm::PixmapReader Class Reference

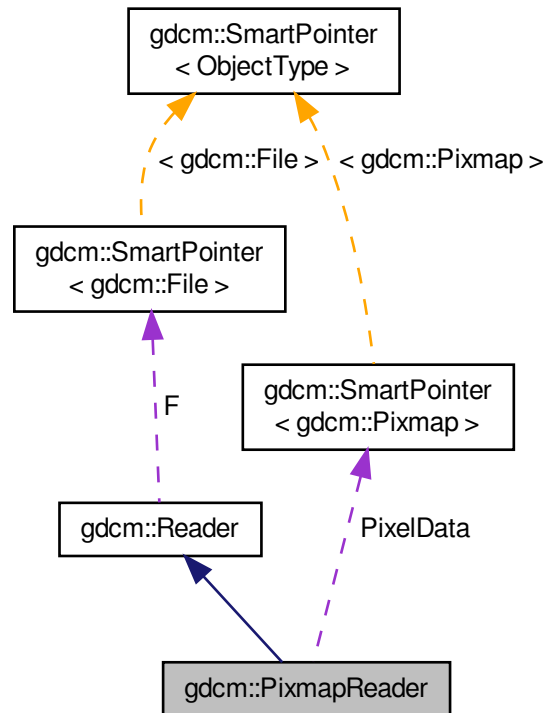
[PixmapReader](#).

```
#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](#)

### 27.208.1 Detailed Description

[PixmapReader](#).

#### Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::Pixmap](#) representation By default it is also loading the lookup table and overlay when found as they impact the rendering of 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](#)

### 27.208.2 Constructor & Destructor Documentation

27.208.2.1 `gdcm::PixmapReader::PixmapReader ( )`

27.208.2.2 `virtual gdcm::PixmapReader::~~PixmapReader ( ) [virtual]`

### 27.208.3 Member Function Documentation

27.208.3.1 `const Pixmap& gdcm::PixmapReader::GetPixmap ( ) const`

Return the read image (need to call [Read\(\)](#) first)

27.208.3.2 `Pixmap& gdcm::PixmapReader::GetPixmap ( )`

27.208.3.3 `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](#).

27.208.3.4 `virtual bool gdcm::PixmapReader::ReadACRNEMAIImage ( ) [protected], [virtual]`

Reimplemented in [gdcm::ImageReader](#).

27.208.3.5 `virtual bool gdcm::PixmapReader::ReadImage ( MediaStorage const & ms ) [protected], [virtual]`

Reimplemented in [gdcm::ImageReader](#).



27.208.3.6 `bool gdcm::PixmapReader::ReadImageInternal ( MediaStorage const & ms, bool handlepixeldata = true )`  
[protected]

## 27.208.4 Member Data Documentation

27.208.4.1 `SmartPointer<Pixmap> gdcm::PixmapReader::PixelData` [protected]

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

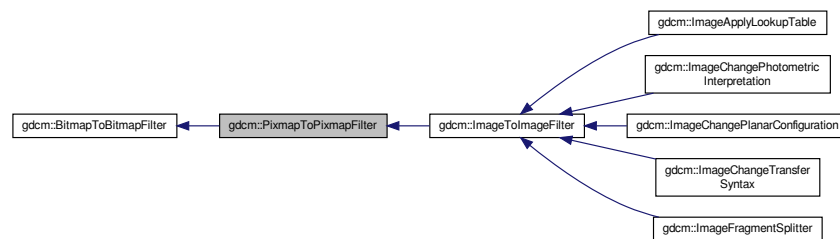
- [gdcmPixmapReader.h](#)

## 27.209 gdcm::PixmapToPixmapFilter Class Reference

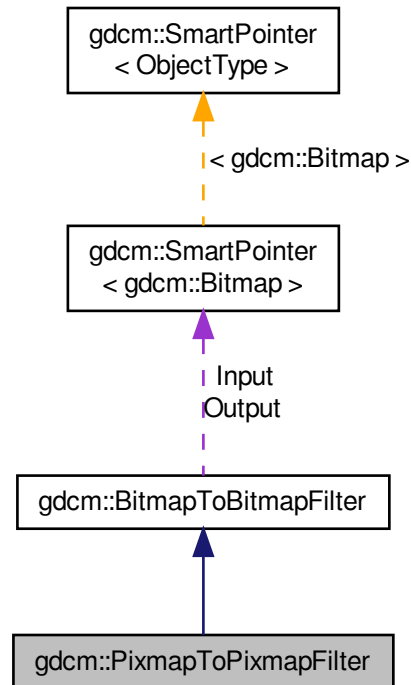
[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

```
#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

### 27.209.1 Detailed Description

[PixmapToPixmapFilter](#) class Super class for all filter taking an image and producing an output image.

### 27.209.2 Constructor & Destructor Documentation

#### 27.209.2.1 `gdcm::PixmapToPixmapFilter::PixmapToPixmapFilter ( )`

27.209.2.2 `gdcm::PixmapToPixmapFilter::~~PixmapToPixmapFilter ( ) [inline]`

### 27.209.3 Member Function Documentation

27.209.3.1 `Pixmap& gdcm::PixmapToPixmapFilter::GetInput ( )`

27.209.3.2 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutput ( ) const`

Get Output image.

27.209.3.3 `const Pixmap& gdcm::PixmapToPixmapFilter::GetOutputAsPixmap ( ) const`

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

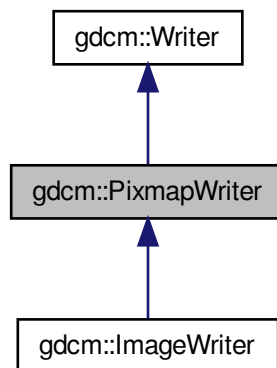
- [gdcmPixmapToPixmapFilter.h](#)

## 27.210 gdcm::PixmapWriter Class Reference

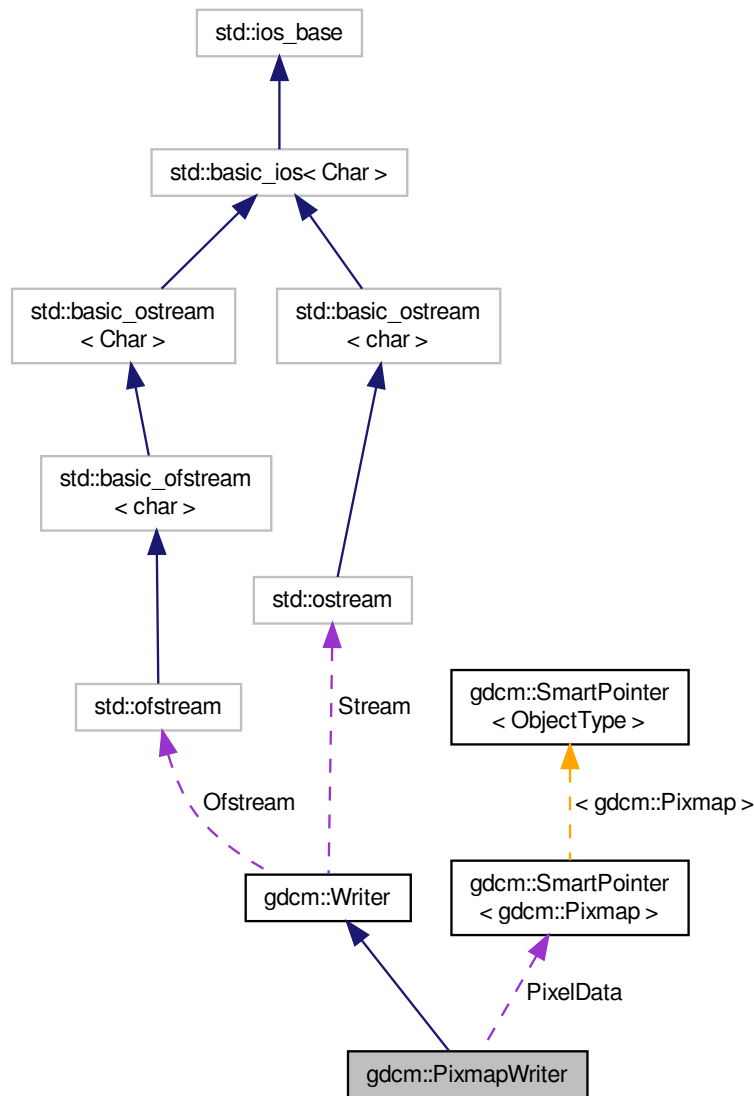
[PixmapWriter](#) This class will takes two inputs:

```
#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](#) ()

### Protected Attributes

- [SmartPointer](#)< [Pixmap](#) > [PixelData](#)

## 27.210.1 Detailed Description

[PixmapWriter](#) This class will takes two inputs:

1. The DICOM [DataSet](#)
2. The [Image](#) input It will override any info from the [Image](#) over the [DataSet](#).

For instance when one read in a lossy compressed image and write out as unencapsulated (ie implicitly lossless) then some attribute are definitely needed to mark this dataset as Lossy (typically 0028,2114)

Examples:

[FileChangeTS.cs](#).

## 27.210.2 Constructor & Destructor Documentation

27.210.2.1 `gdcm::PixmapWriter::PixmapWriter ( )`

27.210.2.2 `gdcm::PixmapWriter::~~PixmapWriter ( )`

## 27.210.3 Member Function Documentation

27.210.3.1 `void gdcm::PixmapWriter::DolconImage ( DataSet & ds, Pixmap const & image )` `[protected]`

27.210.3.2 `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](#).

27.210.3.3 `virtual Pixmap& gdcm::PixmapWriter::GetImage ( )` `[inline],[virtual]`

Reimplemented in [gdcm::ImageWriter](#).

27.210.3.4 `const Pixmap& gdcM::PixmapWriter::GetPixmap ( ) const` [inline]

27.210.3.5 `Pixmap& gdcM::PixmapWriter::GetPixmap ( )` [inline]

27.210.3.6 `bool gdcM::PixmapWriter::PrepareWrite ( )` [protected]

27.210.3.7 `virtual void gdcM::PixmapWriter::SetImage ( Pixmap const & img )` [virtual]

Examples:

[CompressImage.cxx](#), [GenFakelImage.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), and [MergeTwoFiles.cxx](#).

27.210.3.8 `void gdcM::PixmapWriter::SetPixmap ( Pixmap const & img )`

27.210.3.9 `bool gdcM::PixmapWriter::Write ( )` [virtual]

Write.

Reimplemented from [gdcM::Writer](#).

## 27.210.4 Member Data Documentation

27.210.4.1 `SmartPointer<Pixmap> gdcM::PixmapWriter::PixelData` [protected]

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

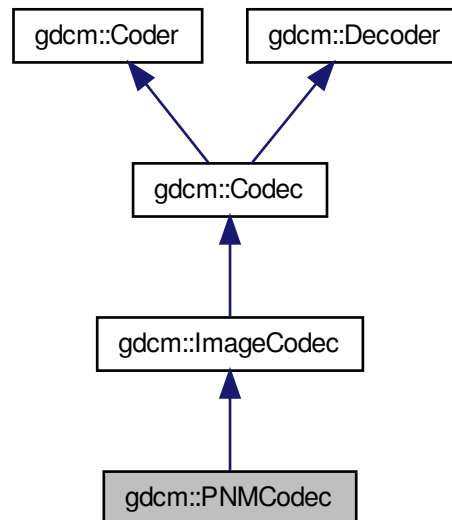
- [gdcMPixmapWriter.h](#)

## 27.211 gdcM::PNMCodec Class Reference

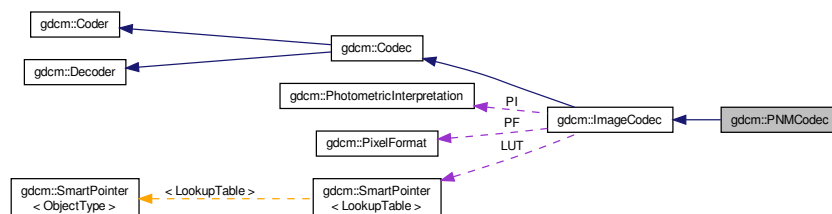
Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.↵

```
#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

### 27.211.1 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](#).

### 27.211.2 Constructor & Destructor Documentation

27.211.2.1 `gdcm::PNMCodec::PNMCodec ( )`

27.211.2.2 `gdcm::PNMCodec::~~PNMCodec ( )`

### 27.211.3 Member Function Documentation

27.211.3.1 `bool gdcm::PNMCodec::CanCode ( TransferSyntax const & ) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.211.3.2 `bool gdcm::PNMCodec::CanDecode ( TransferSyntax const & ) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.211.3.3 `virtual ImageCodec* gdcm::PNMCodec::Clone ( ) const` `[virtual]`

Implements [gdcm::ImageCodec](#).

27.211.3.4 `unsigned long gdcm::PNMCodec::GetBufferLength ( ) const` `[inline]`

27.211.3.5 `bool gdcm::PNMCodec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).



27.211.3.6 `bool gdcm::PNMCodec::Read ( const char * filename, DataElement & out ) const`

27.211.3.7 `void gdcm::PNMCodec::SetBufferLength ( unsigned long l ) [inline]`

27.211.3.8 `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](#)

## 27.212 gdcm::Preamble Class Reference

DICOM [Preamble](#) (Part 10)

```
#include <gdcmPreamble.h>
```

### Public Member Functions

- [Preamble](#) ()
- [Preamble](#) ([Preamble](#) const &)
- [~Preamble](#) ()
- void [Clear](#) ()
- void [Create](#) ()
- const char \* [GetInternal](#) () const
- [VL GetLength](#) () const
- bool [IsEmpty](#) () const
- [Preamble](#) & [operator=](#) ([Preamble](#) const &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [Remove](#) ()
- void [Valid](#) ()
- std::ostream const & [Write](#) (std::ostream &os) const

### Protected Member Functions

- bool [IsValid](#) () const

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [Preamble](#) &\_val)

### 27.212.1 Detailed Description

DICOM [Preamble](#) (Part 10)

### 27.212.2 Constructor & Destructor Documentation

27.212.2.1 `gdcm::Preamble::Preamble ( )`

27.212.2.2 `gdcm::Preamble::~~Preamble ( )`

27.212.2.3 `gdcm::Preamble::Preamble ( Preamble const & )` `[inline]`

### 27.212.3 Member Function Documentation

27.212.3.1 `void gdcm::Preamble::Clear ( )`

27.212.3.2 `void gdcm::Preamble::Create ( )`

27.212.3.3 `const char* gdcm::Preamble::GetInternal ( ) const` `[inline]`

27.212.3.4 `VL gdcm::Preamble::GetLength ( ) const` `[inline]`

27.212.3.5 `bool gdcm::Preamble::IsEmpty ( ) const` `[inline]`

27.212.3.6 `bool gdcm::Preamble::IsValid ( ) const` `[inline]`, `[protected]`

27.212.3.7 `Preamble& gdcm::Preamble::operator= ( Preamble const & )` `[inline]`

27.212.3.8 `void gdcm::Preamble::Print ( std::ostream & os ) const`

27.212.3.9 `std::istream& gdcm::Preamble::Read ( std::istream & is )`

27.212.3.10 `void gdcm::Preamble::Remove ( )`

27.212.3.11 `void gdcm::Preamble::Valid ( )`

27.212.3.12 `std::ostream const& gdcm::Preamble::Write ( std::ostream & os ) const`

### 27.212.4 Friends And Related Function Documentation

27.212.4.1 `std::ostream& operator<< ( std::ostream & _os, const Preamble & _val )` `[friend]`

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

- [gdcmPreamble.h](#)

## 27.213 gdcm::PresentationContext Class Reference

[PresentationContext](#).

```
#include <gdcmPresentationContext.h>
```

## 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 \*as)
- void [SetPresentationContextID](#) (uint8\_t id)

### 27.213.1 Detailed Description

[PresentationContext](#).

See also

[PresentationContextAC](#) [PresentationContextRQ](#)

### 27.213.2 Member Typedef Documentation

27.213.2.1 typedef TransferSyntaxArrayType::size\_type gdcm::PresentationContext::SizeType

27.213.2.2 typedef std::vector<std::string> gdcm::PresentationContext::TransferSyntaxArrayType

### 27.213.3 Constructor & Destructor Documentation

27.213.3.1 gdcm::PresentationContext::PresentationContext ( )

27.213.3.2 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 ).

### 27.213.4 Member Function Documentation

27.213.4.1 void gdcm::PresentationContext::AddTransferSyntax ( const char \* tsstr )

```

27.213.4.2  const char* gdcmm::PresentationContext::GetAbstractSyntax ( ) const  [inline]

27.213.4.3  SizeType gdcmm::PresentationContext::GetNumberOfTransferSyntaxes ( ) const  [inline]

27.213.4.4  uint8_t gdcmm::PresentationContext::GetPresentationContextID ( ) const

27.213.4.5  const char* gdcmm::PresentationContext::GetTransferSyntax ( SizeType i ) const  [inline]

27.213.4.6  bool gdcmm::PresentationContext::operator==( const PresentationContext & pc ) const  [inline]

27.213.4.7  void gdcmm::PresentationContext::Print ( std::ostream & os ) const

27.213.4.8  void gdcmm::PresentationContext::SetAbstractSyntax ( const char * as )  [inline]

27.213.4.9  void gdcmm::PresentationContext::SetPresentationContextID ( uint8_t id )

```

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

- [gdcmmPresentationContext.h](#)

## 27.214 gdcmm::network::PresentationContextAC Class Reference

[PresentationContextAC Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmmPresentationContextAC.h>
```

### Public Member Functions

- [PresentationContextAC](#) ()
- uint8\_t [GetPresentationContextID](#) () const
- uint8\_t [GetReason](#) () const
- [TransferSyntaxSub](#) const & [GetTransferSyntax](#) () const
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- void [SetPresentationContextID](#) (uint8\_t id)
- void [SetReason](#) (uint8\_t r)
- void [SetTransferSyntax](#) ([TransferSyntaxSub](#) const &ts)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 27.214.1 Detailed Description

[PresentationContextAC Table](#) 9-18 PRESENTATION CONTEXT ITEM FIELDS.

See also

[PresentationContext](#)

## 27.214.2 Constructor & Destructor Documentation

27.214.2.1 `gdcm::network::PresentationContextAC::PresentationContextAC ( )`

## 27.214.3 Member Function Documentation

27.214.3.1 `uint8_t gdcm::network::PresentationContextAC::GetPresentationContextID ( ) const` `[inline]`

27.214.3.2 `uint8_t gdcm::network::PresentationContextAC::GetReason ( ) const` `[inline]`

27.214.3.3 `TransferSyntaxSub const& gdcm::network::PresentationContextAC::GetTransferSyntax ( ) const` `[inline]`

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

27.214.3.5 `std::istream& gdcm::network::PresentationContextAC::Read ( std::istream & is )`

27.214.3.6 `void gdcm::network::PresentationContextAC::SetPresentationContextID ( uint8_t id )`

27.214.3.7 `void gdcm::network::PresentationContextAC::SetReason ( uint8_t r )` `[inline]`

27.214.3.8 `void gdcm::network::PresentationContextAC::SetTransferSyntax ( TransferSyntaxSub const & ts )`

27.214.3.9 `size_t gdcm::network::PresentationContextAC::Size ( ) const`

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

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

- [gdcmPresentationContextAC.h](#)

## 27.215 gdcm::PresentationContextGenerator Class Reference

[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.

```
#include <gdcmPresentationContextGenerator.h>
```

### Public Types

- `typedef std::vector`  
`< PresentationContext > PresentationContextArrayType`
- `typedef`  
`PresentationContextArrayType::size_type SizeType`

### Public Member Functions

- [PresentationContextGenerator](#) ( )
- `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 \*as, const char \*ts)
- const char \* [GetDefaultTransferSyntax](#) () const

### 27.215.1 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](#).

### 27.215.2 Member Typedef Documentation

27.215.2.1 `typedef std::vector<PresentationContext> gdcm::PresentationContextGenerator::PresentationContextArrayType`

27.215.2.2 `typedef PresentationContextArrayType::size_type gdcm::PresentationContextGenerator::SizeType`

### 27.215.3 Constructor & Destructor Documentation

27.215.3.1 `gdcm::PresentationContextGenerator::PresentationContextGenerator ( )`

### 27.215.4 Member Function Documentation

27.215.4.1 `bool gdcm::PresentationContextGenerator::AddPresentationContext ( const char * as, const char * ts )`  
`[protected]`

27.215.4.2 `bool gdcm::PresentationContextGenerator::GenerateFromFilenames ( const Directory::FilenamesType & files )`

Generate the [PresentationContext](#) array from a File-Set. [File](#) specified needs to be valid DICOM files. Used for C-ST↔ORE operations

Examples:

[CStoreQtProgress.cxx](#).

27.215.4.3 `bool gdcm::PresentationContextGenerator::GenerateFromUID ( UIDs::TSName asname )`

Generate the [PresentationContext](#) array from a UID (eg. VerificationSOPClass)

27.215.4.4 `const char* gdcm::PresentationContextGenerator::GetDefaultTransferSyntax ( ) const` `[protected]`

27.215.4.5 `PresentationContextArrayType const& gdcm::PresentationContextGenerator::GetPresentationContexts ( )`  
`[inline]`

Examples:

[CStoreQtProgress.cxx](#).

27.215.4.6 `void gdcm::PresentationContextGenerator::SetDefaultTransferSyntax ( const TransferSyntax & ts )`

Not implemented for now. GDCM internally uses Implicit Little Endian.

27.215.4.7 `void gdcm::PresentationContextGenerator::SetMergeModeToAbstractSyntax ( )`

27.215.4.8 `void gdcm::PresentationContextGenerator::SetMergeModeToTransferSyntax ( )`

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

- [gdcmPresentationContextGenerator.h](#)

## 27.216 gdcm::network::PresentationContextRQ Class Reference

[PresentationContextRQ](#) Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

```
#include <gdcmPresentationContextRQ.h>
```

### Public Types

- `typedef std::vector`  
`< TransferSyntaxSub >`  
`::size_type SizeType`

## Public Member Functions

- [PresentationContextRQ](#) ()
- [PresentationContextRQ](#) ([UIDs::TSName](#) asname, [UIDs::TSName](#) tsname=[UIDs::ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM](#))
- [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 &as)
- void [SetPresentationContextID](#) ([uint8\\_t](#) id)
- [size\\_t](#) [Size](#) () const
- const [std::ostream](#) & [Write](#) ([std::ostream](#) &os) const

### 27.216.1 Detailed Description

[PresentationContextRQ](#) Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

See also

[PresentationContextAC](#)

### 27.216.2 Member Typedef Documentation

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

### 27.216.3 Constructor & Destructor Documentation

27.216.3.1 `gdcm::network::PresentationContextRQ::PresentationContextRQ ( )`

27.216.3.2 `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 ).

27.216.3.3 `gdcm::network::PresentationContextRQ::PresentationContextRQ ( const PresentationContext & pc )`

### 27.216.4 Member Function Documentation

27.216.4.1 `void gdcm::network::PresentationContextRQ::AddTransferSyntax ( TransferSyntaxSub const & ts )`



- 27.216.4.2 **AbstractSyntax** const& gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) const [inline]
- 27.216.4.3 **AbstractSyntax&** gdcm::network::PresentationContextRQ::GetAbstractSyntax ( ) [inline]
- 27.216.4.4 **SizeType** gdcm::network::PresentationContextRQ::GetNumberOfTransferSyntaxes ( ) const [inline]
- 27.216.4.5 **uint8\_t** gdcm::network::PresentationContextRQ::GetPresentationContextID ( ) const
- 27.216.4.6 **TransferSyntaxSub** const& gdcm::network::PresentationContextRQ::GetTransferSyntax ( **SizeType** *i* ) const [inline]
- 27.216.4.7 **TransferSyntaxSub&** gdcm::network::PresentationContextRQ::GetTransferSyntax ( **SizeType** *i* ) [inline]
- 27.216.4.8 **std::vector<TransferSyntaxSub>** const& gdcm::network::PresentationContextRQ::GetTransferSyntaxes ( ) const [inline]
- 27.216.4.9 **bool** gdcm::network::PresentationContextRQ::operator== ( const **PresentationContextRQ** & *pc* ) const [inline]
- 27.216.4.10 **void** gdcm::network::PresentationContextRQ::Print ( std::ostream & *os* ) const
- 27.216.4.11 **std::istream&** gdcm::network::PresentationContextRQ::Read ( std::istream & *is* )
- 27.216.4.12 **void** gdcm::network::PresentationContextRQ::SetAbstractSyntax ( **AbstractSyntax** const & *as* )
- 27.216.4.13 **void** gdcm::network::PresentationContextRQ::SetPresentationContextID ( **uint8\_t** *id* )
- 27.216.4.14 **size\_t** gdcm::network::PresentationContextRQ::Size ( ) const
- 27.216.4.15 **const std::ostream&** gdcm::network::PresentationContextRQ::Write ( std::ostream & *os* ) const

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

- [gdcmPresentationContextRQ.h](#)

## 27.217 gdcm::network::PresentationDataValue Class Reference

[PresentationDataValue](#) Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

```
#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`)

### 27.217.1 Detailed Description

[PresentationDataValue Table](#) 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.

### 27.217.2 Constructor & Destructor Documentation

27.217.2.1 `gdcm::network::PresentationDataValue::PresentationDataValue ( )`

### 27.217.3 Member Function Documentation

27.217.3.1 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobs ( const std::vector< PresentationDataValue > & inPDVs ) [static]`

Warning

[DataSet](#) will be read as Implicit Little Endian TS

27.217.3.2 `static DataSet gdcm::network::PresentationDataValue::ConcatenatePDVBlobsAsExplicit ( const std::vector< PresentationDataValue > & inPDVs ) [static]`

27.217.3.3 `const std::string& gdcm::network::PresentationDataValue::GetBlob ( ) const`

27.217.3.4 `bool gdcm::network::PresentationDataValue::GetIsCommand ( ) const`

27.217.3.5 `bool gdcm::network::PresentationDataValue::GetIsLastFragment ( ) const`

27.217.3.6 `uint8_t gdcm::network::PresentationDataValue::GetMessageHeader ( ) const [inline]`

27.217.3.7 `uint8_t gdcm::network::PresentationDataValue::GetPresentationContextID ( ) const [inline]`

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

27.217.3.9 `std::istream& gdcm::network::PresentationDataValue::Read ( std::istream & is )`

27.217.3.10 `std::istream& gdcm::network::PresentationDataValue::ReadInto ( std::istream & is, std::ostream & os )`

27.217.3.11 `void gdcm::network::PresentationDataValue::SetBlob ( const std::string & partialblob )`

27.217.3.12 `void gdcm::network::PresentationDataValue::SetCommand ( bool inCommand )`

27.217.3.13 `void gdcm::network::PresentationDataValue::SetDataSet ( const DataSet & ds )`

Set [DataSet](#). Write [DataSet](#) in implicit.

#### Warning

size of dataset should be below maxpdusize

27.217.3.14 `void gdcm::network::PresentationDataValue::SetLastFragment ( bool inLast )`

27.217.3.15 `void gdcm::network::PresentationDataValue::SetMessageHeader ( uint8_t messageheader )` `[inline]`

27.217.3.16 `void gdcm::network::PresentationDataValue::SetPresentationContextID ( uint8_t id )` `[inline]`

27.217.3.17 `size_t gdcm::network::PresentationDataValue::Size ( )` `const`

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

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

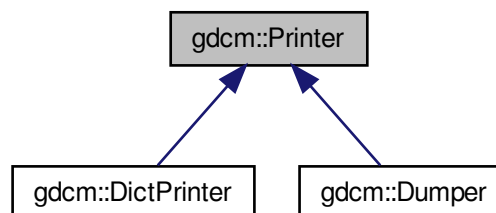
- [gdcmPresentationDataValue.h](#)

## 27.218 gdcm::Printer Class Reference

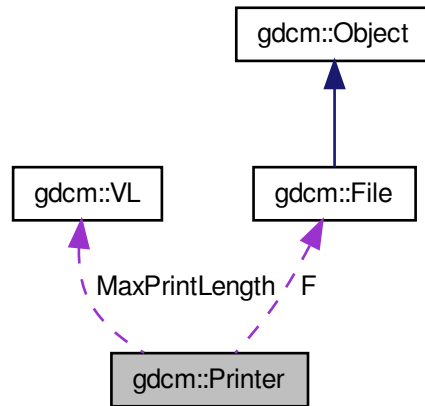
[Printer](#) class.

```
#include <gdcmPrinter.h>
```

Inheritance diagram for `gdcm::Printer`:



Collaboration diagram for `gdcM::Printer`:



## Public Types

- enum `PrintStyles` {  
`VERBOSE_STYLE` = 0,  
`CONDENSED_STYLE`,  
`XML` }

## 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

### 27.218.1 Detailed Description

[Printer](#) class.

### 27.218.2 Member Enumeration Documentation

#### 27.218.2.1 enum gdcm::Printer::PrintStyles

Enumerator

***VERBOSE\_STYLE***  
***CONDENSED\_STYLE***  
***XML***

### 27.218.3 Constructor & Destructor Documentation

#### 27.218.3.1 gdcm::Printer::Printer ( )

#### 27.218.3.2 gdcm::Printer::~~Printer ( )

### 27.218.4 Member Function Documentation

#### 27.218.4.1 [PrintStyles](#) gdcm::Printer::GetPrintStyle ( ) const [inline]

Get PrintStyle value.

#### 27.218.4.2 void gdcm::Printer::Print ( std::ostream & os )

Print.

#### 27.218.4.3 VR gdcm::Printer::PrintDataElement ( std::ostream & os, const [Dicts](#) & *dicts*, const [DataSet](#) & *ds*, const [DataElement](#) & *de*, std::ostream & *out*, std::string const & *indent* ) [protected]

#### 27.218.4.4 void gdcm::Printer::PrintDataSet ( const [DataSet](#) & *ds*, std::ostream & *os*, const std::string & *s* = " " )

Print an individual dataset.

27.218.4.5 void `gdcM::Printer::PrintSQ` ( const `SequenceOfItems` \* *sqi*, `std::ostream` & *os*, `std::string` const & *indent* )  
[protected]

27.218.4.6 void `gdcM::Printer::SetColor` ( bool *c* )

Set color mode or not.

27.218.4.7 void `gdcM::Printer::SetFile` ( `File` const & *f* ) [inline]

Set file.

27.218.4.8 void `gdcM::Printer::SetStyle` ( `PrintStyles` *ps* ) [inline]

Set `PrintStyle` value.

## 27.218.5 Member Data Documentation

27.218.5.1 const `File`\* `gdcM::Printer::F` [protected]

27.218.5.2 VL `gdcM::Printer::MaxPrintLength` [protected]

27.218.5.3 `PrintStyles` `gdcM::Printer::PrintStyle` [protected]

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

- [gdcMPrinter.h](#)

## 27.219 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)`

## 27.219.1 Detailed Description

Private [Dict](#).

## 27.219.2 Constructor & Destructor Documentation

27.219.2.1 `gdcm::PrivateDict::PrivateDict ( )` `[inline]`

27.219.2.2 `gdcm::PrivateDict::~~PrivateDict ( )` `[inline]`

## 27.219.3 Member Function Documentation

27.219.3.1 `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`.

27.219.3.2 `bool gdcm::PrivateDict::FindDictEntry ( const PrivateTag &tag ) const` `[inline]`

27.219.3.3 `const DictEntry& gdcm::PrivateDict::GetDictEntry ( const PrivateTag &tag ) const` `[inline]`

27.219.3.4 `bool gdcm::PrivateDict::IsEmpty ( ) const` `[inline]`

27.219.3.5 `void gdcm::PrivateDict::LoadDefault ( )` `[protected]`

27.219.3.6 `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()`.

27.219.3.7 `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.

## 27.219.4 Friends And Related Function Documentation

27.219.4.1 `friend class Dicts` `[friend]`

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

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

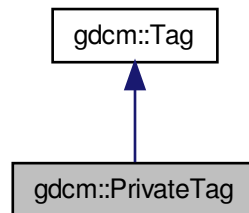
- [gdcmDict.h](#)

## 27.220 gdcm::PrivateTag Class Reference

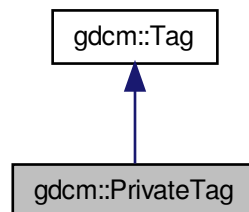
Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)

```
#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)



### 27.220.1 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:

[csa2img.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [ELSCINT1WaveToText.cxx](#), [FileStreaming.cs](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [MrProtocol.cxx](#), [pmsct\\_rgb1.cxx](#), [PublicDict.cxx](#), [ReadGEMSSDO.cxx](#), and [rle2img.cxx](#).

### 27.220.2 Constructor & Destructor Documentation

27.220.2.1 `gdcm::PrivateTag::PrivateTag ( uint16_t group = 0, uint16_t element = 0, const char * owner = " " )` `[inline]`

27.220.2.2 `gdcm::PrivateTag::PrivateTag ( Tag const & t, const char * owner = " " )` `[inline]`

References `gdcm::Tag::GetElement()`.

### 27.220.3 Member Function Documentation

27.220.3.1 `DataElement gdcm::PrivateTag::GetAsDataElement ( )` `const`

27.220.3.2 `const char* gdcm::PrivateTag::GetOwner ( )` `const` `[inline]`

#### Examples:

[PublicDict.cxx](#).

Referenced by `gdcm::PrivateDict::PrintXML()`.

27.220.3.3 `bool gdcm::PrivateTag::operator< ( const PrivateTag & _val )` `const`

27.220.3.4 `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")

27.220.3.5 `void gdcm::PrivateTag::SetOwner ( const char * owner )` `[inline]`

### 27.220.4 Friends And Related Function Documentation

27.220.4.1 `std::ostream& operator<< ( std::ostream & _os, const PrivateTag & _val )` `[friend]`

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

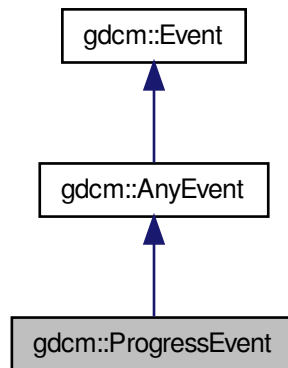
- [gdcmPrivateTag.h](#)

## 27.221 gdcM::ProgressEvent Class Reference

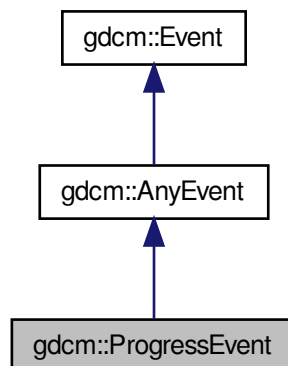
[ProgressEvent](#) Special type of event triggered during.

```
#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)

### 27.221.1 Detailed Description

[ProgressEvent](#) Special type of event triggered during.

See also

[AnyEvent](#)

### 27.221.2 Member Typedef Documentation

27.221.2.1 typedef [ProgressEvent](#) [gdcm::ProgressEvent::Self](#)

27.221.2.2 typedef [AnyEvent](#) [gdcm::ProgressEvent::Superclass](#)

### 27.221.3 Constructor & Destructor Documentation

27.221.3.1 [gdcm::ProgressEvent::ProgressEvent](#) ( double *p* = 0 ) [\[inline\]](#)

27.221.3.2 virtual [gdcm::ProgressEvent::~~ProgressEvent](#) ( ) [\[inline\]](#),[\[virtual\]](#)

27.221.3.3 [gdcm::ProgressEvent::ProgressEvent](#) ( const [Self](#) & *s* ) [\[inline\]](#)

### 27.221.4 Member Function Documentation

27.221.4.1 virtual bool [gdcm::ProgressEvent::CheckEvent](#) ( const [::gdcm::Event](#) \* *e* ) const [\[inline\]](#),[\[virtual\]](#)

27.221.4.2 virtual const char\* [gdcm::ProgressEvent::GetEventName](#) ( ) const [\[inline\]](#),[\[virtual\]](#)

Return the StringName associated with the event.

Implements [gdcm::Event](#).

27.221.4.3 double [gdcm::ProgressEvent::GetProgress](#) ( ) const [\[inline\]](#)

27.221.4.4 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](#).

27.221.4.5 void `gdcm::ProgressEvent::SetProgress ( double p )` `[inline]`

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

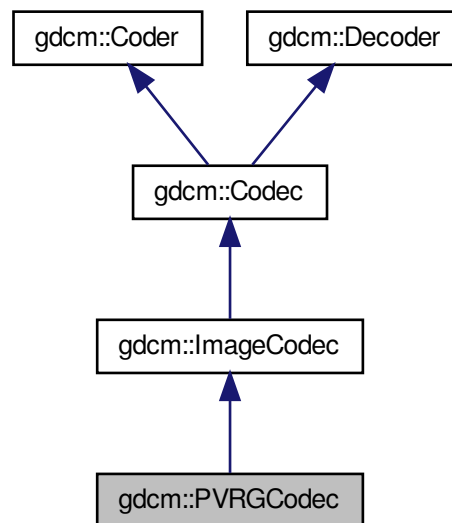
- [gdcmProgressEvent.h](#)

## 27.222 `gdcm::PVRGCodec` Class Reference

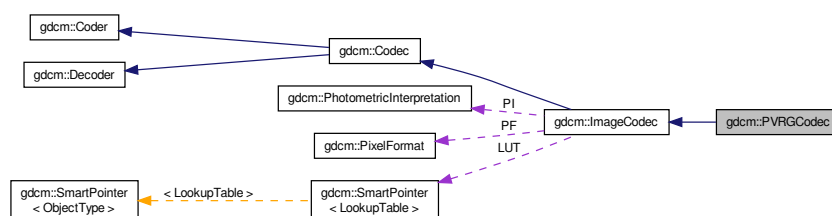
[PVRGCodec](#).

```
#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

### 27.222.1 Detailed Description

[PVRGCodec](#).

#### Note

pvr 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...

### 27.222.2 Constructor & Destructor Documentation

27.222.2.1 `gdcm::PVRGCodec::PVRGCodec ( )`

27.222.2.2 `gdcm::PVRGCodec::~~PVRGCodec ( )`

### 27.222.3 Member Function Documentation

27.222.3.1 `bool gdcm::PVRGCodec::CanCode ( TransferSyntax const & ) const` `[virtual]`

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.222.3.2 `bool gdcm::PVRGCodec::CanDecode ( TransferSyntax const & ) const` `[virtual]`

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.222.3.3 `virtual ImageCodec* gdcm::PVRGCodec::Clone ( ) const` [virtual]

Implements [gdcm::ImageCodec](#).

27.222.3.4 `bool gdcm::PVRGCodec::Code ( DataElement const & in_, DataElement & out_ )` [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.222.3.5 `bool gdcm::PVRGCodec::Decode ( DataElement const &, DataElement & )` [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.222.3.6 `void gdcm::PVRGCodec::SetLossyFlag ( bool / )`

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

- [gdcmPVRGCodec.h](#)

## 27.223 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 use)

### 27.223.1 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.

## 27.223.2 Constructor & Destructor Documentation

27.223.2.1 `gdcm::PythonFilter::PythonFilter ( )`

27.223.2.2 `gdcm::PythonFilter::~~PythonFilter ( )`

## 27.223.3 Member Function Documentation

27.223.3.1 `File& gdcm::PythonFilter::GetFile ( )` `[inline]`

27.223.3.2 `const File& gdcm::PythonFilter::GetFile ( ) const` `[inline]`

27.223.3.3 `void gdcm::PythonFilter::SetDicts ( const Dicts & dicts )`

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

27.223.3.5 `PyObject* gdcm::PythonFilter::ToPyObject ( const Tag & t ) const`

27.223.3.6 `void gdcm::PythonFilter::UseDictAlways ( bool use )` `[inline]`

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

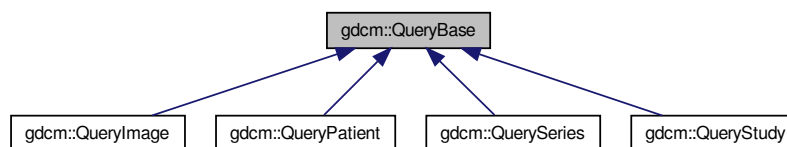
- [gdcmPythonFilter.h](#)

## 27.224 gdcm::QueryBase Class Reference

[QueryBase](#) contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

```
#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 > GetHierarchicalSearchTags (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

### 27.224.1 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.

### 27.224.2 Constructor & Destructor Documentation

27.224.2.1 virtual gdcmm::QueryBase::~~QueryBase ( ) [inline],[virtual]

### 27.224.3 Member Function Documentation

27.224.3.1 std::vector<[Tag](#)> gdcmm::QueryBase::GetAllRequiredTags ( const [ERootType](#) & *inRootType* ) const

In order to validate a query dataset we need to check that there exists at least one required (or unique) key

27.224.3.2 std::vector<[Tag](#)> gdcmm::QueryBase::GetAllTags ( const [ERootType](#) & *inRootType* ) const

In order to validate a query dataset, just check for the presence of a tag, not it's requirement level in the spec

27.224.3.3 virtual std::vector<[Tag](#)> gdcmm::QueryBase::GetHierarchicalSearchTags ( const [ERootType](#) & *inRootType* ) const  
[pure virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implemented in [gdcmm::QueryImage](#), [gdcmm::QueryPatient](#), [gdcmm::QuerySeries](#), and [gdcmm::QueryStudy](#).



27.224.3.4 `virtual const char* gdcm::QueryBase::GetName ( ) const` [pure virtual]

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.224.3.5 `virtual std::vector<Tag> gdcm::QueryBase::GetOptionalTags ( const ERootType & inRootType ) const` [pure virtual]

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.224.3.6 `virtual DataElement gdcm::QueryBase::GetQueryLevel ( ) const` [pure virtual]

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.224.3.7 `virtual std::vector<Tag> gdcm::QueryBase::GetRequiredTags ( const ERootType & inRootType ) const` [pure virtual]

Implemented in [gdcm::QueryImage](#), [gdcm::QueryPatient](#), [gdcm::QuerySeries](#), and [gdcm::QueryStudy](#).

27.224.3.8 `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](#)

## 27.225 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 [BaseRootQuery](#) \* [ProduceQuery](#) (ERootType inRootType, [EQueryType](#) inQueryType, [EQueryLevel](#) inQueryLevel)

### 27.225.1 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

**27.225.2 Member Function Documentation****27.225.2.1 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() ).

**27.225.2.2 static void gdcm::QueryFactory::ListCharSets ( std::ostream & os ) [static]**

List all possible CharSet.

**27.225.2.3 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

**27.225.2.4 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:

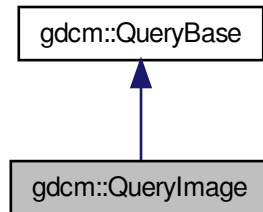
- [gdcmQueryFactory.h](#)

**27.226 gdcm::QueryImage Class Reference**

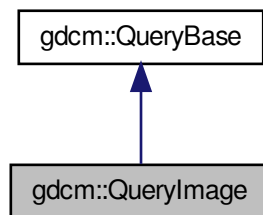
[QueryImage](#) contains: class to construct an image-based query for C-FIND and C-MOVE.

```
#include <gdcmQueryImage.h>
```

Inheritance diagram for gdcm::QueryImage:



Collaboration diagram for gdcm::QueryImage:



## 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`

### 27.226.1 Detailed Description

[QueryImage](#) contains: class to construct an image-based query for C-FIND and C-MOVE.

## 27.226.2 Member Function Documentation

27.226.2.1 `std::vector<Tag> gdcM::QueryImage::GetHierachicalSearchTags ( const ERootType & inRootType ) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcM::QueryBase](#).

27.226.2.2 `const char* gdcM::QueryImage::GetName ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.226.2.3 `std::vector<Tag> gdcM::QueryImage::GetOptionalTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.226.2.4 `DataElement gdcM::QueryImage::GetQueryLevel ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.226.2.5 `std::vector<Tag> gdcM::QueryImage::GetRequiredTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.226.2.6 `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:

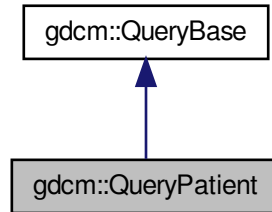
- [gdcMQueryImage.h](#)

## 27.227 gdcM::QueryPatient Class Reference

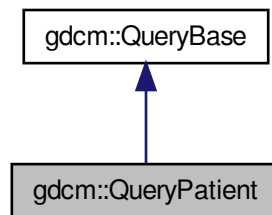
[QueryPatient](#) contains: class to construct a patient-based query for c-find and c-move.

```
#include <gdcMQueryPatient.h>
```

Inheritance diagram for gdcm::QueryPatient:



Collaboration diagram for gdcm::QueryPatient:



## 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

### 27.227.1 Detailed Description

`QueryPatient` contains: class to construct a patient-based query for c-find and c-move.

### 27.227.2 Member Function Documentation

**27.227.2.1** `std::vector<Tag> gdcm::QueryPatient::GetHierachicalSearchTags ( const ERootType & inRootType ) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcm::QueryBase](#).

**27.227.2.2** `const char* gdcm::QueryPatient::GetName ( ) const` [virtual]

Implements [gdcm::QueryBase](#).

**27.227.2.3** `std::vector<Tag> gdcm::QueryPatient::GetOptionalTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcm::QueryBase](#).

**27.227.2.4** `DataElement gdcm::QueryPatient::GetQueryLevel ( ) const` [virtual]

Implements [gdcm::QueryBase](#).

**27.227.2.5** `std::vector<Tag> gdcm::QueryPatient::GetRequiredTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcm::QueryBase](#).

**27.227.2.6** `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:

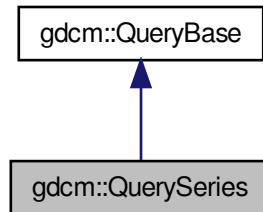
- [gdcmQueryPatient.h](#)

## 27.228 gdcm::QuerySeries Class Reference

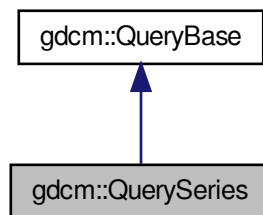
[QuerySeries](#) contains: class to construct a series-based query for c-find and c-move.

```
#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`

### 27.228.1 Detailed Description

[QuerySeries](#) contains: class to construct a series-based query for c-find and c-move.

## 27.228.2 Member Function Documentation

27.228.2.1 `std::vector<Tag> gdcM::QuerySeries::GetHierachicalSearchTags ( const ERootType & inRootType ) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcM::QueryBase](#).

27.228.2.2 `const char* gdcM::QuerySeries::GetName ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.228.2.3 `std::vector<Tag> gdcM::QuerySeries::GetOptionalTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.228.2.4 `DataElement gdcM::QuerySeries::GetQueryLevel ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.228.2.5 `std::vector<Tag> gdcM::QuerySeries::GetRequiredTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.228.2.6 `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](#)

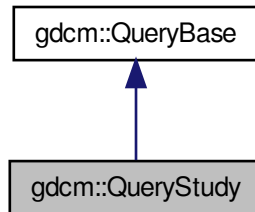
## 27.229 gdcM::QueryStudy Class Reference

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

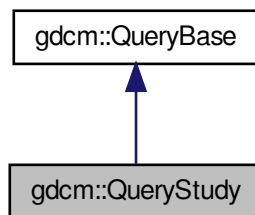
```
#include <gdcMQueryStudy.h>
```



Inheritance diagram for gdcm::QueryStudy:



Collaboration diagram for gdcm::QueryStudy:



## 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`

### 27.229.1 Detailed Description

QueryStudy.h contains: class to construct a study-based query for C-FIND and C-MOVE.

## 27.229.2 Member Function Documentation

27.229.2.1 `std::vector<Tag> gdcM::QueryStudy::GetHierachicalSearchTags ( const ERootType & inRootType ) const` [virtual]

Return all Unique Key for a particular Query Root type (from the same level and above).

Implements [gdcM::QueryBase](#).

27.229.2.2 `const char* gdcM::QueryStudy::GetName ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.229.2.3 `std::vector<Tag> gdcM::QueryStudy::GetOptionalTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.229.2.4 `DataElement gdcM::QueryStudy::GetQueryLevel ( ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.229.2.5 `std::vector<Tag> gdcM::QueryStudy::GetRequiredTags ( const ERootType & inRootType ) const` [virtual]

Implements [gdcM::QueryBase](#).

27.229.2.6 `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:

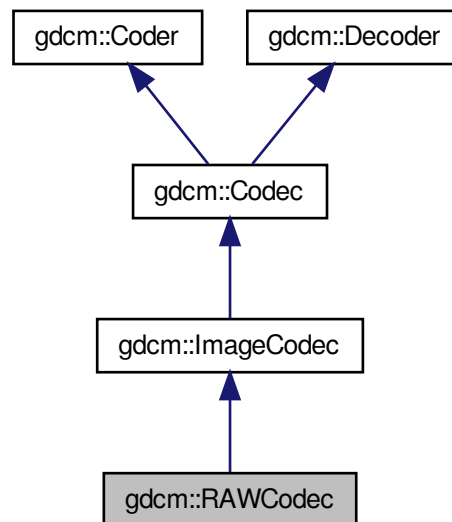
- [gdcMQueryStudy.h](#)

## 27.230 gdcM::RAWCodec Class Reference

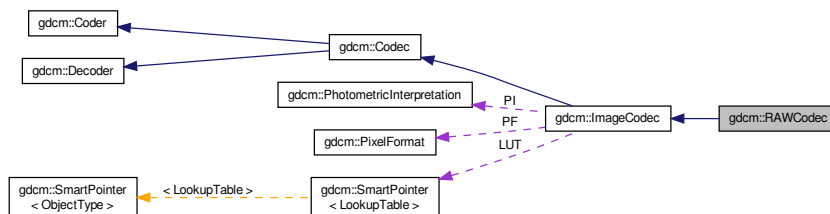
[RAWCodec](#) class.

```
#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

#### 27.230.1 Detailed Description

[RAWCodec](#) class.

#### 27.230.2 Constructor & Destructor Documentation

27.230.2.1 [gdcm::RAWCodec::RAWCodec](#) ( )

27.230.2.2 [gdcm::RAWCodec::~~RAWCodec](#) ( )

#### 27.230.3 Member Function Documentation

27.230.3.1 bool [gdcm::RAWCodec::CanCode](#) ( [TransferSyntax](#) const & ) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).

27.230.3.2 bool [gdcm::RAWCodec::CanDecode](#) ( [TransferSyntax](#) const & ) const [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcm::ImageCodec](#).

27.230.3.3 virtual [ImageCodec\\*](#) [gdcm::RAWCodec::Clone](#) ( ) const [virtual]

Implements [gdcm::ImageCodec](#).

27.230.3.4 bool [gdcm::RAWCodec::Code](#) ( [DataElement](#) const & *in\_*, [DataElement](#) & *out\_* ) [virtual]

Code.

Reimplemented from [gdcm::Coder](#).

27.230.3.5 bool [gdcm::RAWCodec::Decode](#) ( [DataElement](#) const &, [DataElement](#) & ) [virtual]

Decode.

Reimplemented from [gdcm::ImageCodec](#).

27.230.3.6 `bool gdcm::RAWCodec::DecodeByStreams ( std::istream & is, std::ostream & os )` `[protected]`, `[virtual]`

Reimplemented from [gdcm::ImageCodec](#).

27.230.3.7 `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.

27.230.3.8 `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:

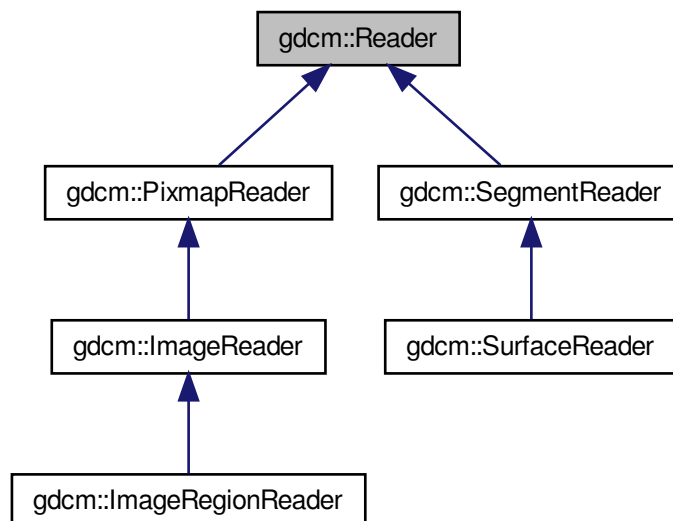
- [gdcmRAWCodec.h](#)

## 27.231 gdcm::Reader Class Reference

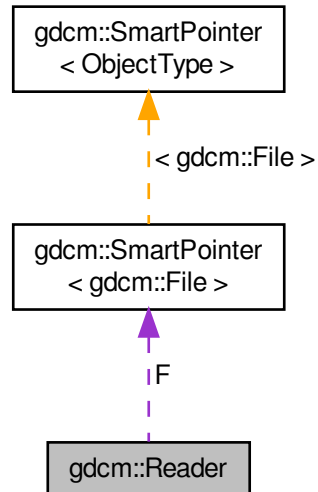
[Reader](#) ala DOM (Document [Object](#) Model)

```
#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.*
- `virtual bool Read ()`  
*Main function to read a file.*
- `bool ReadSelectedPrivateTags (std::set< PrivateTag > const &tags, 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](#)

### 27.231.1 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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [ScanDirectory.java](#), [SimplePrintPatientName.cs](#), and [TestReader.cxx](#).

## 27.231.2 Constructor & Destructor Documentation

27.231.2.1 `gdcm::Reader::Reader ( )` `[inline]`

27.231.2.2 `virtual gdcm::Reader::~~Reader ( )` `[virtual]`

## 27.231.3 Member Function Documentation

27.231.3.1 `bool gdcm::Reader::CanRead ( ) const`

Test whether this is a DICOM file

Warning

need to call either `SetFileName` or `SetStream` first

Examples:

[ReadUTF8QtDir.cxx](#).

27.231.3.2 `const File& gdcm::Reader::GetFile ( ) const` `[inline]`

Set/Get [File](#).

Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.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](#), and [TestReader.cxx](#).

27.231.3.3 `File& gdcm::Reader::GetFile ( )` `[inline]`

Set/Get [File](#).

27.231.3.4 `std::istream* gdcm::Reader::GetStreamPtr ( ) const` `[inline]`, `[protected]`

27.231.3.5 `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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpImageHeaderInfo.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveToText.cxx](#), [ExtractEncryptedContent.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.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](#), and [TestReader.cxx](#).



[SequenceUltrasound.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct\\_↵  
rgb1.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGE↵  
MSSDO.cxx](#), [rle2img.cxx](#), and [TestReader.cxx](#).

27.231.3.6 `bool gdcm::Reader::ReadDataSet ( )` [protected]

27.231.3.7 `bool gdcm::Reader::ReadMetaInformation ( )` [protected]

27.231.3.8 `bool gdcm::Reader::ReadPreamble ( )` [protected]

27.231.3.9 `bool gdcm::Reader::ReadSelectedPrivateTags ( std::set< PrivateTag > const & ptags, bool readvalues = true )`

Will only read the specified selected private tags.

27.231.3.10 `bool gdcm::Reader::ReadSelectedTags ( std::set< Tag > const & tags, bool readvalues = true )`

Will only read the specified selected tags.

27.231.3.11 `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>	

27.231.3.12 `void gdcm::Reader::SetFile ( File & file )` [inline]

Set/Get [File](#).

27.231.3.13 `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:

[ChangeSequenceUltrasound.cxx](#), [CheckBigEndianBug.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [ConvertToQImage.cxx](#), [csa2img.cxx](#), [DiffFile.cxx](#), [DumpADAC.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovie↵  
Group.cxx](#), [DumpImageHeaderInfo.cxx](#), [DumpPhilipsECHO.cxx](#), [DuplicatePCDE.cxx](#), [ELSCINT1WaveTo↵  
Text.cxx](#), [ExtractEncryptedContent.cxx](#), [ExtractIconFromFile.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [Get↵  
SequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [MrProtocol.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReadGEMSSDO.cxx](#), [ReadMultiTimesException.cxx](#), [ReadUTF8QtDir.cxx](#), [rle2img.cxx](#), [TestReader.cxx](#), and [threadgdcm.cxx](#).

27.231.3.14 `void gdcM::Reader::SetStream ( std::istream & input_stream )` [inline]

Set the open-ed stream directly.

Examples:

[ReadUTF8QtDir.cxx](#).

## 27.231.4 Friends And Related Function Documentation

27.231.4.1 `friend class StreamImageReader` [friend]

## 27.231.5 Member Data Documentation

27.231.5.1 `SmartPointer<File> gdcM::Reader::F` [protected]

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

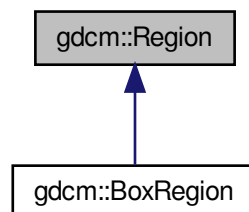
- [gdcMReader.h](#)

## 27.232 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.*

### 27.232.1 Detailed Description

Class for manipulation region.

### 27.232.2 Constructor & Destructor Documentation

27.232.2.1 `gdcm::Region::Region ( )`

27.232.2.2 `virtual gdcm::Region::~~Region ( )` [virtual]

### 27.232.3 Member Function Documentation

27.232.3.1 `virtual size_t gdcm::Region::Area ( ) const` [pure virtual]

compute the area

Implemented in [gdcm::BoxRegion](#).

27.232.3.2 `virtual Region* gdcm::Region::Clone ( ) const` [pure virtual]

Implemented in [gdcm::BoxRegion](#).

27.232.3.3 `virtual BoxRegion gdcm::Region::ComputeBoundingBox ( )` [pure virtual]

Return the Axis-Aligned minimum bounding box for all regions.

Implemented in [gdcm::BoxRegion](#).

27.232.3.4 `virtual bool gdcm::Region::Empty ( ) const` [pure virtual]

return whether this domain is empty:

Implemented in [gdcm::BoxRegion](#).

27.232.3.5 `virtual bool gdcm::Region::IsValid ( ) const` [pure virtual]

return whether this is valid domain

Implemented in [gdcm::BoxRegion](#).

27.232.3.6 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](#)

## 27.233 gdcm::Rescaler Class Reference

**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.

```
#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)`

### 27.233.1 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 [V←R: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](#)

### 27.233.2 Constructor & Destructor Documentation

27.233.2.1 `gdcm::Rescaler::Rescaler ( )` `[inline]`

27.233.2.2 `gdcm::Rescaler::~~Rescaler ( )` `[inline]`

### 27.233.3 Member Function Documentation

27.233.3.1 `PixelFormat::ScalarType gdcm::Rescaler::ComputeInterceptSlopePixelFormat ( )`

Compute the Pixel Format of the output data Used for direct transformation

### 27.233.3.2 PixelFormat gdcm::Rescaler::ComputePixelTypeFromMinMax ( )

Compute the Pixel Format of the output data Used for inverse transformation

### 27.233.3.3 double gdcm::Rescaler::GetIntercept ( ) const [inline]

### 27.233.3.4 double gdcm::Rescaler::GetSlope ( ) const [inline]

### 27.233.3.5 bool gdcm::Rescaler::InverseRescale ( char \* out, const char \* in, size\_t n )

Inverse transform.

### 27.233.3.6 template<typename TIn > void gdcm::Rescaler::InverseRescaleFunctionIntoBestFit ( char \* out, const TIn \* in, size\_t n ) [protected]

### 27.233.3.7 bool gdcm::Rescaler::Rescale ( char \* out, const char \* in, size\_t n )

Direct transform.

### 27.233.3.8 template<typename TIn > void gdcm::Rescaler::RescaleFunctionIntoBestFit ( char \* out, const TIn \* in, size\_t n ) [protected]

### 27.233.3.9 void gdcm::Rescaler::SetIntercept ( double i ) [inline]

Set Intercept: used for both direct&inverse transformation.

### 27.233.3.10 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

### 27.233.3.11 void gdcm::Rescaler::SetPixelFormat ( PixelFormat const & pf ) [inline]

Set Pixel Format of input data.

### 27.233.3.12 void gdcm::Rescaler::SetSlope ( double s ) [inline]

Set Slope: user for both direct&inverse transformation.

### 27.233.3.13 void gdcm::Rescaler::SetTargetPixelType ( PixelFormat const & targetst )

By default (when UseTargetPixelType is false), a best matching Target Pixel [Type](#) is computed. However user can override this auto selection by switching UseTargetPixelType:true and also specifying the specifix Target Pixel [Type](#)

### 27.233.3.14 void gdcm::Rescaler::SetUseTargetPixelType ( bool b )

Override default behavior of Rescale.

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

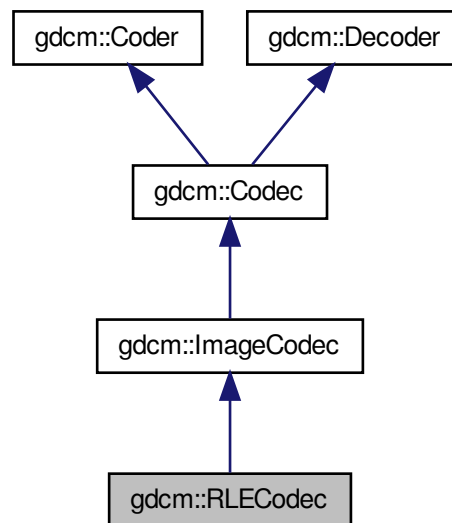
- [gdcmRescaler.h](#)

## 27.234 gdcm::RLECodec Class Reference

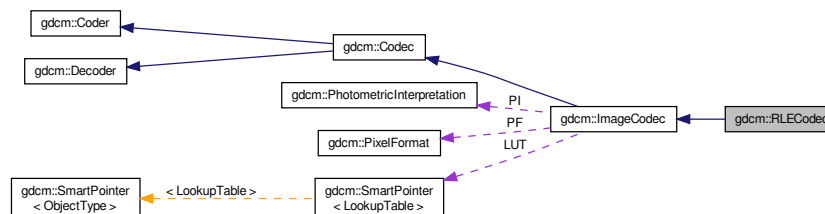
Class to do RLE.

```
#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 [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)

### Friends

- class [ImageRegionReader](#)

### Additional Inherited Members

#### 27.234.1 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).

#### 27.234.2 Constructor & Destructor Documentation

27.234.2.1 [gdcm::RLECodec::RLECodec](#) ( )

27.234.2.2 [gdcm::RLECodec::~~RLECodec](#) ( )

#### 27.234.3 Member Function Documentation

27.234.3.1 bool [gdcm::RLECodec::CanCode](#) ( [TransferSyntax](#) const & ) const [virtual]

Return whether this coder support this transfer syntax (can code it)

Reimplemented from [gdcm::ImageCodec](#).



27.234.3.2 `bool gdcmm::RLECodec::CanDecode ( TransferSyntax const & ) const` [virtual]

Return whether this decoder support this transfer syntax (can decode it)

Reimplemented from [gdcmm::ImageCodec](#).

27.234.3.3 `virtual ImageCodec* gdcmm::RLECodec::Clone ( ) const` [virtual]

Implements [gdcmm::ImageCodec](#).

27.234.3.4 `bool gdcmm::RLECodec::Code ( DataElement const & in_, DataElement & out_ )` [virtual]

Code.

Reimplemented from [gdcmm::Coder](#).

27.234.3.5 `bool gdcmm::RLECodec::Decode ( DataElement const &, DataElement & )` [virtual]

Decode.

Reimplemented from [gdcmm::ImageCodec](#).

27.234.3.6 `bool gdcmm::RLECodec::DecodeByStreams ( std::istream & is, std::ostream & os )` [protected],[virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.234.3.7 `bool gdcmm::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]

27.234.3.8 `unsigned long gdcmm::RLECodec::GetBufferLength ( ) const` [inline]

27.234.3.9 `bool gdcmm::RLECodec::GetHeaderInfo ( std::istream & is, TransferSyntax & ts )` [virtual]

Reimplemented from [gdcmm::ImageCodec](#).

27.234.3.10 `void gdcmm::RLECodec::SetBufferLength ( unsigned long l )` [inline]

27.234.3.11 `void gdcmm::RLECodec::SetLength ( unsigned long l )` [inline]

## 27.234.4 Friends And Related Function Documentation

27.234.4.1 `friend class ImageRegionReader` [friend]

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

- [gdcmmRLECodec.h](#)

## 27.235 gdcm::network::RoleSelectionSub Class Reference

[RoleSelectionSub](#) PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

```
#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

### 27.235.1 Detailed Description

[RoleSelectionSub](#) PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

### 27.235.2 Constructor & Destructor Documentation

27.235.2.1 `gdcm::network::RoleSelectionSub::RoleSelectionSub ( )`

### 27.235.3 Member Function Documentation

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

27.235.3.2 `std::istream& gdcm::network::RoleSelectionSub::Read ( std::istream & is )`

27.235.3.3 `void gdcm::network::RoleSelectionSub::SetTuple ( const char * uid, uint8_t scurole, uint8_t scprole )`

27.235.3.4 `size_t gdcm::network::RoleSelectionSub::Size ( ) const`

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

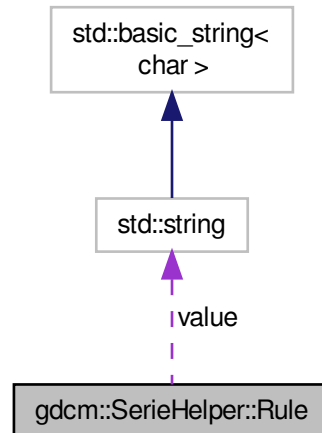
The documentation for this class was generated from the following file:

- [gdcmRoleSelectionSub.h](#)

## 27.236 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](#)

### 27.236.1 Member Data Documentation

27.236.1.1 `uint16_t` `gdcm::SerieHelper::Rule::elem`

27.236.1.2 `uint16_t` `gdcm::SerieHelper::Rule::group`

27.236.1.3 `int` `gdcm::SerieHelper::Rule::op`

27.236.1.4 `std::string` `gdcm::SerieHelper::Rule::value`

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

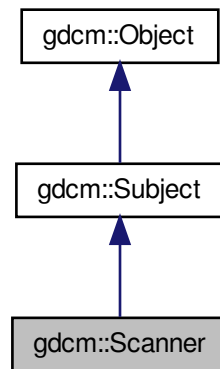
- [gdcmSerieHelper.h](#)

## 27.237 gdcm::Scanner Class Reference

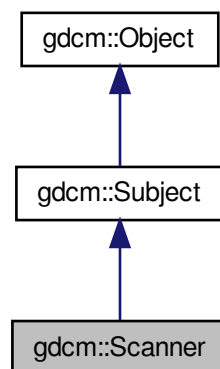
[Scanner](#) This filter is meant for quickly browsing a [FileSet](#) (a set of files on disk). Special consideration are taken so as to read the minimum amount of information in each file in order to retrieve the user specified set of DICOM [Attribute](#).

```
#include <gdcmScanner.h>
```

Inheritance diagram for `gdcm::Scanner`:



Collaboration diagram for `gdcm::Scanner`:



## Classes

- struct [ltstr](#)

## Public Types

- typedef `MappingType::const_iterator` [ConstIterator](#)

- typedef std::map< const char \*, [TagToValue](#), [ltstr](#) > [MappingType](#)
- typedef std::map< [Tag](#), const char \* > [TagToValue](#)
- typedef [TagToValue](#)::value\_type [TagToValueValueType](#)
- typedef std::set< std::string > [ValuesType](#)

## Public Member Functions

- [Scanner](#) ()
- [~Scanner](#) ()
- 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](#)< [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)

### 27.237.1 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 [gdcm::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](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 27.237.2 Member Typedef Documentation

27.237.2.1 typedef MappingType::const\_iterator [gdcm::Scanner::ConstIterator](#)

27.237.2.2 typedef std::map<const char \*,[TagToValue](#), Itstr> [gdcm::Scanner::MappingType](#)

27.237.2.3 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)

27.237.2.4 `typedef TagToValue::value_type gdcm::Scanner::TagToValueValueType`

27.237.2.5 `typedef std::set< std::string > gdcm::Scanner::ValuesType`

## 27.237.3 Constructor & Destructor Documentation

27.237.3.1 `gdcm::Scanner::Scanner ( )` `[inline]`

27.237.3.2 `gdcm::Scanner::~~Scanner ( )`

## 27.237.4 Member Function Documentation

27.237.4.1 `void gdcm::Scanner::AddPrivateTag ( PrivateTag const & t )`

27.237.4.2 `void gdcm::Scanner::AddSkipTag ( Tag const & t )`

Add a tag that will need to be skipped. Those are root level skip tags.

27.237.4.3 `void gdcm::Scanner::AddTag ( Tag const & t )`

Add a tag that will need to be read. Those are root level skip tags.

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.237.4.4 `ConstIterator gdcm::Scanner::Begin ( ) const` `[inline]`

27.237.4.5 `void gdcm::Scanner::ClearSkipTags ( )`

27.237.4.6 `void gdcm::Scanner::ClearTags ( )`

27.237.4.7 `ConstIterator gdcm::Scanner::End ( ) const` `[inline]`

27.237.4.8 `Directory::FileNamesType gdcm::Scanner::GetAllFileNamesFromTagToValue ( Tag const & t, const char * valuref ) const`

Will loop over all files and return a vector of std::strings of filenames where value match the reference value 'valuref'

27.237.4.9 `const char* gdcm::Scanner::GetFilenameFromTagToValue ( Tag const & t, const char * valuref ) const`

Will loop over all files and return the first file where value match the reference value 'valuref'

27.237.4.10 `Directory::FileNamesType const& gdcm::Scanner::GetFileNames ( ) const` `[inline]`

27.237.4.11 `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](#).

**27.237.4.12 TagToValue** const& gdcm::Scanner::GetMapping ( const char \* *filename* ) const

Get the std::map mapping filenames to value for file 'filename'.

Examples:

[DumpToSQLITE3.cxx](#), and [SimpleScanner.cxx](#).

**27.237.4.13 TagToValue** const& gdcm::Scanner::GetMappingFromTagToValue ( Tag const & *t*, const char \* *value* ) const

See [GetFilenameFromTagToValue\(\)](#). This is simply GetFilenameFromTagToValue followed.

**27.237.4.14 MappingType** const& gdcm::Scanner::GetMappings ( ) const [inline]

Mappings are the mapping from a particular tag to the map, mapping filename to value:

**27.237.4.15 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)

**27.237.4.16** 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 !

**27.237.4.17 ValuesType** const& gdcm::Scanner::GetValues ( ) const [inline]

Get all the values found (in lexicographic order)

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

**27.237.4.18 ValuesType** gdcm::Scanner::GetValues ( Tag const & *t* ) const

Get all the values found (in lexicographic order) associated with Tag 't'.



27.237.4.19 `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](#), and [SimpleScanner.cxx](#).

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

for wrapped language: instantiate a reference counted object

27.237.4.21 `void gdcm::Scanner::Print ( std::ostream & os ) const [virtual]`

Print result.

Reimplemented from [gdcm::Object](#).

Referenced by `gdcm::operator<<()`.

27.237.4.22 `void gdcm::Scanner::ProcessPublicTag ( StringFilter & sf, const char * filename ) [protected]`

27.237.4.23 `bool gdcm::Scanner::Scan ( Directory::FileNamesType const & filenames )`

Start the scan !

Examples:

[DiscriminateVolume.cxx](#), [DumpToSQLITE3.cxx](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

## 27.237.5 Friends And Related Function Documentation

27.237.5.1 `std::ostream& operator<< ( std::ostream & _os, const Scanner & s ) [friend]`

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

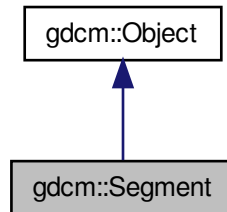
- [gdcmScanner.h](#)

## 27.238 gdcm::Segment Class Reference

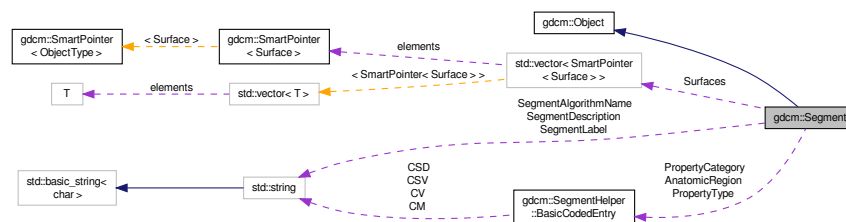
This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.

```
#include <gdcmSegment.h>
```

Inheritance diagram for `gdc::Segment`:



Collaboration diagram for `gdc::Segment`:



## Public Types

- enum `ALGOType` {  
`MANUAL` = 0,  
`AUTOMATIC`,  
`ALGOType_END` }
- 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()`
- `SegmentHelper::BasicCodedEntry`  
`const & GetPropertyCategory()` const
- `SegmentHelper::BasicCodedEntry & GetPropertyCategory()`

- [SegmentHelper::BasicCodedEntry](#)  
const & [GetPropertyType](#) () const
- [SegmentHelper::BasicCodedEntry](#) & [GetPropertyType](#) ()
- 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 [SetPropertyCategory](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- void [SetPropertyType](#) ([SegmentHelper::BasicCodedEntry](#) const &BSE)
- 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](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyCategory](#)
- [SegmentHelper::BasicCodedEntry](#) [PropertyType](#)
- std::string [SegmentAlgorithmName](#)
- [ALGOType](#) [SegmentAlgorithmType](#)
- std::string [SegmentDescription](#)
- std::string [SegmentLabel](#)
- unsigned short [SegmentNumber](#)
- unsigned long [SurfaceCount](#)
- [SurfaceVector](#) [Surfaces](#)

### Additional Inherited Members

#### 27.238.1 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

## 27.238.2 Member Typedef Documentation

27.238.2.1 `typedef std::vector< SmartPointer< Surface > > gdcm::Segment::SurfaceVector`

## 27.238.3 Member Enumeration Documentation

27.238.3.1 `enum gdcm::Segment::ALGOType`

Enumerator

***MANUAL***

***AUTOMATIC***

***ALGOType\_END***

## 27.238.4 Constructor & Destructor Documentation

27.238.4.1 `gdcm::Segment::Segment ( )`

27.238.4.2 `virtual gdcm::Segment::~~Segment ( ) [virtual]`

## 27.238.5 Member Function Documentation

27.238.5.1 `void gdcm::Segment::AddSurface ( SmartPointer< Surface > surface )`

27.238.5.2 `static ALGOType gdcm::Segment::GetALGOType ( const char * type ) [static]`

27.238.5.3 `static const char* gdcm::Segment::GetALGOTypeString ( ALGOType type ) [static]`

27.238.5.4 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetAnatomicRegion ( ) const`

27.238.5.5 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetAnatomicRegion ( )`

27.238.5.6 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyCategory ( ) const`

27.238.5.7 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyCategory ( )`

27.238.5.8 `SegmentHelper::BasicCodedEntry const& gdcm::Segment::GetPropertyType ( ) const`

27.238.5.9 `SegmentHelper::BasicCodedEntry& gdcm::Segment::GetPropertyType ( )`

27.238.5.10 `const char* gdcm::Segment::GetSegmentAlgorithmName ( ) const`

27.238.5.11 `ALGOType gdcm::Segment::GetSegmentAlgorithmType ( ) const`

27.238.5.12 `const char* gdcm::Segment::GetSegmentDescription ( ) const`

27.238.5.13 `const char* gdcm::Segment::GetSegmentLabel ( ) const`

27.238.5.14 `unsigned short gdcm::Segment::GetSegmentNumber ( ) const`

27.238.5.15 `SmartPointer< Surface > gdcm::Segment::GetSurface ( const unsigned int idx = 0 ) const`

- 27.238.5.16 unsigned long gdcm::Segment::GetSurfaceCount ( )
- 27.238.5.17 SurfaceVector const& gdcm::Segment::GetSurfaces ( ) const
- 27.238.5.18 SurfaceVector& gdcm::Segment::GetSurfaces ( )
- 27.238.5.19 void gdcm::Segment::SetAnatomicRegion ( SegmentHelper::BasicCodedEntry const & *BSE* )
- 27.238.5.20 void gdcm::Segment::SetPropertyCategory ( SegmentHelper::BasicCodedEntry const & *BSE* )
- 27.238.5.21 void gdcm::Segment::SetPropertyType ( SegmentHelper::BasicCodedEntry const & *BSE* )
- 27.238.5.22 void gdcm::Segment::SetSegmentAlgorithmName ( const char \* *name* )
- 27.238.5.23 void gdcm::Segment::SetSegmentAlgorithmType ( ALGOType *type* )
- 27.238.5.24 void gdcm::Segment::SetSegmentAlgorithmType ( const char \* *typeStr* )
- 27.238.5.25 void gdcm::Segment::SetSegmentDescription ( const char \* *description* )
- 27.238.5.26 void gdcm::Segment::SetSegmentLabel ( const char \* *label* )
- 27.238.5.27 void gdcm::Segment::SetSegmentNumber ( const unsigned short *num* )
- 27.238.5.28 void gdcm::Segment::SetSurfaceCount ( const unsigned long *nb* )

## 27.238.6 Member Data Documentation

- 27.238.6.1 SegmentHelper::BasicCodedEntry gdcm::Segment::AnatomicRegion [protected]
- 27.238.6.2 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyCategory [protected]
- 27.238.6.3 SegmentHelper::BasicCodedEntry gdcm::Segment::PropertyType [protected]
- 27.238.6.4 std::string gdcm::Segment::SegmentAlgorithmName [protected]
- 27.238.6.5 ALGOType gdcm::Segment::SegmentAlgorithmType [protected]
- 27.238.6.6 std::string gdcm::Segment::SegmentDescription [protected]
- 27.238.6.7 std::string gdcm::Segment::SegmentLabel [protected]
- 27.238.6.8 unsigned short gdcm::Segment::SegmentNumber [protected]
- 27.238.6.9 unsigned long gdcm::Segment::SurfaceCount [protected]
- 27.238.6.10 SurfaceVector gdcm::Segment::Surfaces [protected]

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

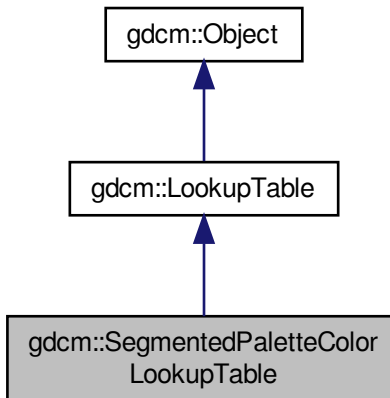
- [gdcmSegment.h](#)

## 27.239 gdcm::SegmentedPaletteColorLookupTable Class Reference

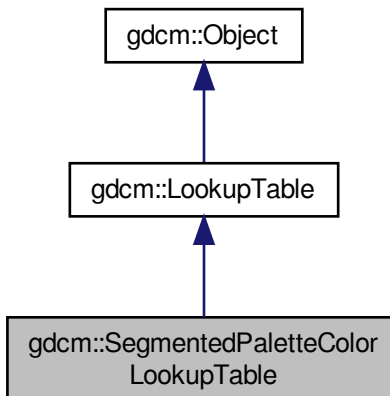
[SegmentedPaletteColorLookupTable](#) class.

```
#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

### 27.239.1 Detailed Description

[SegmentedPaletteColorLookupTable](#) class.

### 27.239.2 Constructor & Destructor Documentation

27.239.2.1 `gdcm::SegmentedPaletteColorLookupTable::SegmentedPaletteColorLookupTable ( )`

27.239.2.2 `gdcm::SegmentedPaletteColorLookupTable::~~SegmentedPaletteColorLookupTable ( )`

### 27.239.3 Member Function Documentation

27.239.3.1 `void gdcm::SegmentedPaletteColorLookupTable::Print ( std::ostream & ) const` `[inline]`, `[virtual]`

Reimplemented from [gdcm::LookupTable](#).

27.239.3.2 `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:

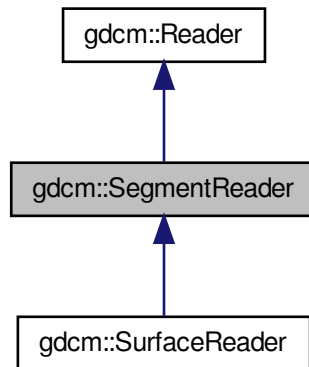
- [gdcmSegmentedPaletteColorLookupTable.h](#)

## 27.240 gdcm::SegmentReader Class Reference

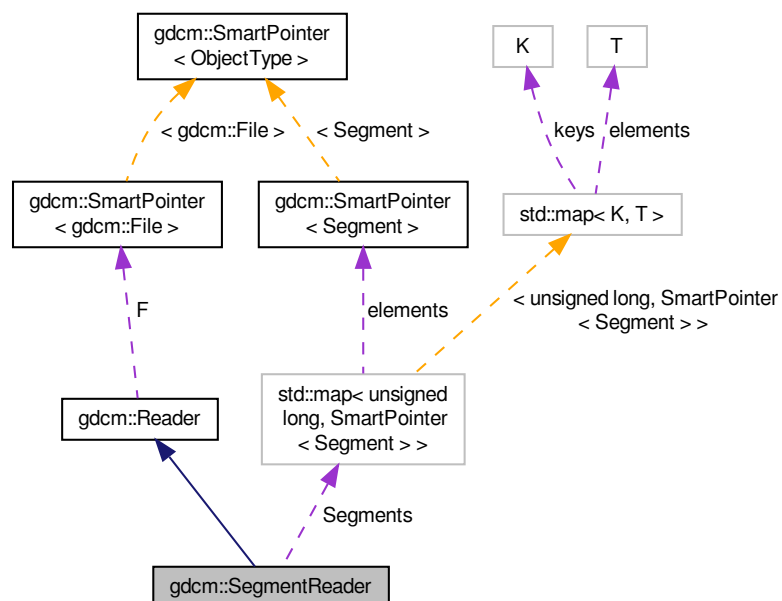
This class defines a segment reader. It reads attributes of group 0x0062.

```
#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](#)

### 27.240.1 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

### 27.240.2 Member Typedef Documentation

27.240.2.1 typedef std::map< unsigned long, [SmartPointer< Segment >](#) > [gdcm::SegmentReader::SegmentMap](#)  
[protected]

27.240.2.2 typedef std::vector< [SmartPointer< Segment >](#) > [gdcm::SegmentReader::SegmentVector](#)

### 27.240.3 Constructor & Destructor Documentation

27.240.3.1 [gdcm::SegmentReader::SegmentReader](#) ( )

27.240.3.2 virtual [gdcm::SegmentReader::~~SegmentReader](#) ( ) [virtual]

#### 27.240.4 Member Function Documentation

27.240.4.1 `const SegmentVector gdcM::SegmentReader::GetSegments ( ) const`

27.240.4.2 `SegmentVector gdcM::SegmentReader::GetSegments ( )`

27.240.4.3 `virtual bool gdcM::SegmentReader::Read ( ) [virtual]`

Read.

Reimplemented from [gdcM::Reader](#).

Reimplemented in [gdcM::SurfaceReader](#).

27.240.4.4 `bool gdcM::SegmentReader::ReadSegment ( const Item & segmentItem, const unsigned int idx ) [protected]`

27.240.4.5 `bool gdcM::SegmentReader::ReadSegments ( ) [protected]`

#### 27.240.5 Member Data Documentation

27.240.5.1 `SegmentMap gdcM::SegmentReader::Segments [protected]`

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

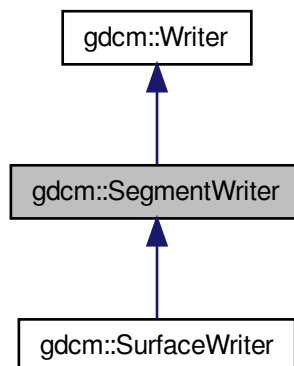
- [gdcMSegmentReader.h](#)

### 27.241 gdcM::SegmentWriter Class Reference

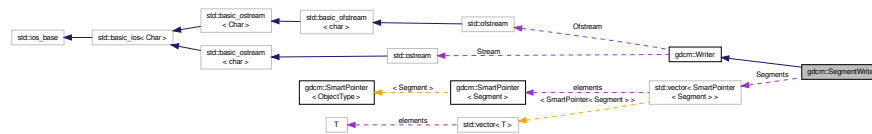
This class defines a segment writer. It writes attributes of group 0x0062.

```
#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](#)

### 27.241.1 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

### 27.241.2 Member Typedef Documentation

27.241.2.1 `typedef std::vector< SmartPointer< Segment > > gdcm::SegmentWriter::SegmentVector`

### 27.241.3 Constructor & Destructor Documentation

27.241.3.1 `gdcm::SegmentWriter::SegmentWriter ( )`

27.241.3.2 `virtual gdcm::SegmentWriter::~~SegmentWriter ( )` `[virtual]`

### 27.241.4 Member Function Documentation

27.241.4.1 `void gdcm::SegmentWriter::AddSegment ( SmartPointer< Segment > segment )`

27.241.4.2 `unsigned int gdcm::SegmentWriter::GetNumberOfSegments ( )` `const`

27.241.4.3 `SmartPointer< Segment > gdcm::SegmentWriter::GetSegment ( const unsigned int idx = 0 )` `const`

27.241.4.4 `const SegmentVector& gdcm::SegmentWriter::GetSegments ( )` `const`

27.241.4.5 `SegmentVector& gdcm::SegmentWriter::GetSegments ( )`

27.241.4.6 `bool gdcm::SegmentWriter::PrepareWrite ( )` `[protected]`

27.241.4.7 `void gdcm::SegmentWriter::SetNumberOfSegments ( const unsigned int size )`

27.241.4.8 `void gdcm::SegmentWriter::SetSegments ( SegmentVector & segments )`

27.241.4.9 `bool gdcm::SegmentWriter::Write ( )` `[virtual]`

Write.

Reimplemented from [gdcm::Writer](#).

Reimplemented in [gdcm::SurfaceWriter](#).

### 27.241.5 Member Data Documentation

27.241.5.1 `SegmentVector gdcm::SegmentWriter::Segments` `[protected]`

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

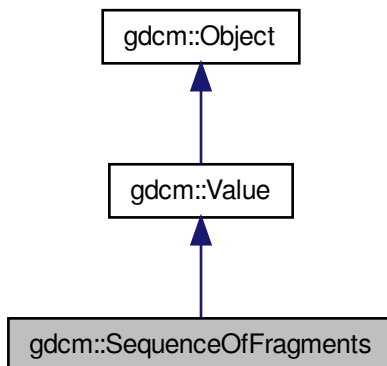
- [gdcmSegmentWriter.h](#)

## 27.242 [gdcm::SequenceOfFragments](#) Class Reference

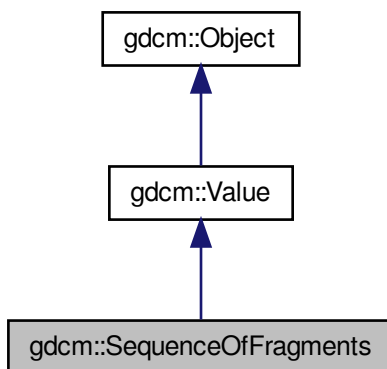
Class to represent a Sequence Of Fragments.

```
#include <gdcmSequenceOfFragments.h>
```

Inheritance diagram for gdcm::SequenceOfFragments:



Collaboration diagram for gdcm::SequenceOfFragments:



## Public Types

- typedef  
FragmentVector::const\_iterator [ConstIterator](#)
- typedef std::vector< [Fragment](#) > [FragmentVector](#)
- typedef FragmentVector::iterator [Iterator](#)
- typedef FragmentVector::size\_type [SizeType](#)

## Public Member Functions

- [SequenceOfFragments](#) ()  
*constructor (UndefinedLength by default)*
- void [AddFragment](#) ([Fragment](#) const &item)  
*Appends a [Fragment](#) to the already added ones.*
- [Iterator Begin](#) ()
- [ConstIterator Begin](#) () const
- void [Clear](#) ()  
*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

### 27.242.1 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](#).

## 27.242.2 Member Typedef Documentation

27.242.2.1 `typedef FragmentVector::const_iterator gdcmm::SequenceOfFragments::ConstIterator`

27.242.2.2 `typedef std::vector<Fragment> gdcmm::SequenceOfFragments::FragmentVector`

27.242.2.3 `typedef FragmentVector::iterator gdcmm::SequenceOfFragments::Iterator`

27.242.2.4 `typedef FragmentVector::size_type gdcmm::SequenceOfFragments::SizeType`

## 27.242.3 Constructor & Destructor Documentation

27.242.3.1 `gdcmm::SequenceOfFragments::SequenceOfFragments ( ) [inline]`

constructor (UndefinedLength by default)

## 27.242.4 Member Function Documentation

27.242.4.1 `void gdcmm::SequenceOfFragments::AddFragment ( Fragment const & item )`

Appends a [Fragment](#) to the already added ones.

Examples:

[FixBrokenJ2K.cxx](#).

27.242.4.2 `Iterator gdcmm::SequenceOfFragments::Begin ( ) [inline]`

27.242.4.3 `ConstIterator gdcmm::SequenceOfFragments::Begin ( ) const [inline]`

27.242.4.4 `void gdcmm::SequenceOfFragments::Clear ( ) [virtual]`

Clear.

Implements [gdcmm::Value](#).

27.242.4.5 `unsigned long gdcmm::SequenceOfFragments::ComputeByteLength ( ) const`

27.242.4.6 `VL gdcmm::SequenceOfFragments::ComputeLength ( ) const`

27.242.4.7 `Iterator gdcmm::SequenceOfFragments::End ( ) [inline]`

27.242.4.8 `ConstIterator gdcmm::SequenceOfFragments::End ( ) const [inline]`

27.242.4.9 `bool gdcmm::SequenceOfFragments::GetBuffer ( char * buffer, unsigned long length ) const`

27.242.4.10 `bool gdcmm::SequenceOfFragments::GetFragBuffer ( unsigned int fragNb, char * buffer, unsigned long & length ) const`

27.242.4.11 `const Fragment& gdcM::SequenceOfFragments::GetFragment ( SizeType num ) const`

Examples:

[FixBrokenJ2K.cxx](#), and [FixJAIBugJPEGSL.cxx](#).

27.242.4.12 `VL gdcM::SequenceOfFragments::GetLength ( ) const [inline],[virtual]`

Returns the SQ length, as read from disk.

Implements [gdcM::Value](#).

27.242.4.13 `SizeType gdcM::SequenceOfFragments::GetNumberOfFragments ( ) const`

Examples:

[FixJAIBugJPEGSL.cxx](#).

27.242.4.14 `const BasicOffsetTable& gdcM::SequenceOfFragments::GetTable ( ) const [inline]`

27.242.4.15 `BasicOffsetTable& gdcM::SequenceOfFragments::GetTable ( ) [inline]`

27.242.4.16 `static SmartPointer<SequenceOfFragments> gdcM::SequenceOfFragments::New ( ) [inline],  
[static]`

27.242.4.17 `bool gdcM::SequenceOfFragments::operator== ( const Value & val ) const [inline],[virtual]`

Implements [gdcM::Value](#).

27.242.4.18 `void gdcM::SequenceOfFragments::Print ( std::ostream & os ) const [inline],[virtual]`

Reimplemented from [gdcM::Object](#).

27.242.4.19 `template<typename TSwap > std::istream& gdcM::SequenceOfFragments::Read ( std::istream & is, bool readvalues =  
true ) [inline]`

27.242.4.20 `template<typename TSwap > std::istream& gdcM::SequenceOfFragments::ReadPreValue ( std::istream & is )  
[inline]`

References [gdcMDebugMacro](#).

27.242.4.21 `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\(\)](#).



27.242.4.22 void gdcm::SequenceOfFragments::SetLength ( VL *length* ) [inline],[virtual]

Sets the actual SQ length.

Implements [gdcm::Value](#).

27.242.4.23 template<typename TSwap > std::ostream const& gdcm::SequenceOfFragments::Write ( std::ostream & *os* ) const [inline]

References [gdcm::VL::Write\(\)](#), and [gdcm::Tag::Write\(\)](#).

27.242.4.24 bool gdcm::SequenceOfFragments::WriteBuffer ( std::ostream & *os* ) const

Examples:

[GetJPEGSamplePrecision.cxx](#).

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

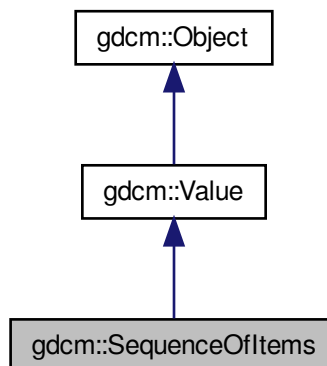
- [gdcmSequenceOfFragments.h](#)

## 27.243 gdcm::SequenceOfItems Class Reference

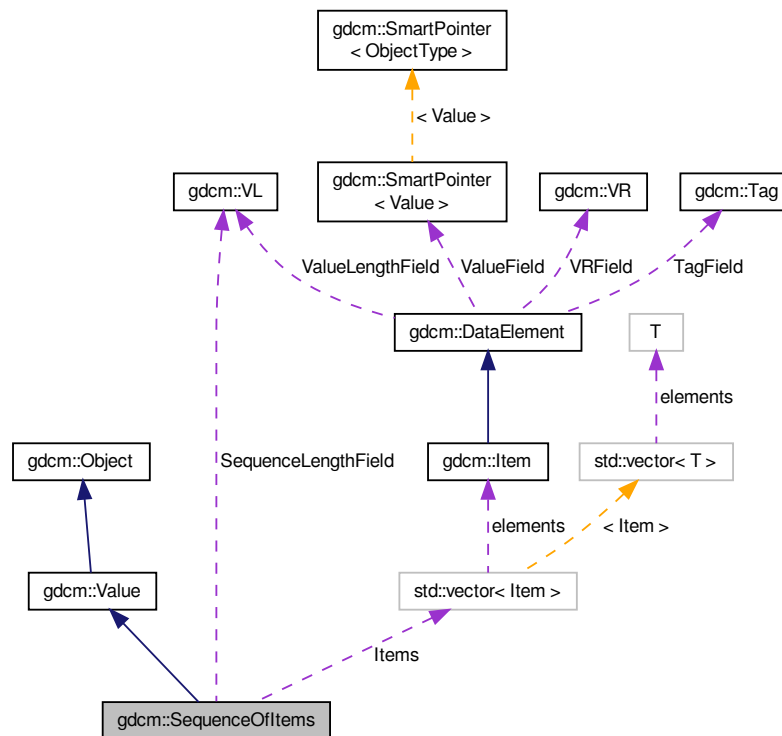
Class to represent a Sequence Of Items (value representation : SQ)

```
#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.*
- [Iterator](#) [Begin](#) ()
- [ConstIterator](#) [Begin](#) () const
- void [Clear](#) ()
- 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 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)`
- `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

### 27.243.1 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](#).

## 27.243.2 Member Typedef Documentation

27.243.2.1 `typedef ItemVector::const_iterator gdcmm::SequenceOfItems::ConstIterator`

27.243.2.2 `typedef std::vector< Item > gdcmm::SequenceOfItems::ItemVector`

27.243.2.3 `typedef ItemVector::iterator gdcmm::SequenceOfItems::Iterator`

27.243.2.4 `typedef ItemVector::size_type gdcmm::SequenceOfItems::SizeType`

## 27.243.3 Constructor &amp; Destructor Documentation

27.243.3.1 `gdcmm::SequenceOfItems::SequenceOfItems ( ) [inline]`

constructor (UndefinedLength by default)

## 27.243.4 Member Function Documentation

27.243.4.1 `void gdcmm::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](#).

27.243.4.2 `Iterator gdcmm::SequenceOfItems::Begin ( ) [inline]`

27.243.4.3 `ConstIterator gdcmm::SequenceOfItems::Begin ( ) const [inline]`

27.243.4.4 `void gdcmm::SequenceOfItems::Clear ( ) [inline],[virtual]`

Implements [gdcmm::Value](#).

27.243.4.5 `template<typename TDE > VL gdcmm::SequenceOfItems::ComputeLength ( ) const`

27.243.4.6 `Iterator gdcmm::SequenceOfItems::End ( ) [inline]`

27.243.4.7 **ConstIterator** gdcm::SequenceOfItems::End ( ) const [inline]

27.243.4.8 **bool** gdcm::SequenceOfItems::FindDataElement ( const Tag & t ) const

27.243.4.9 **const Item&** gdcm::SequenceOfItems::GetItem ( *SizeType position* ) const

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), [LargeVRDS↵Explicit.cxx](#), and [ReadAndDumpDICOMDIR.cxx](#).

27.243.4.10 **Item&** gdcm::SequenceOfItems::GetItem ( *SizeType position* )

27.243.4.11 **VL** gdcm::SequenceOfItems::GetLength ( ) const [inline],[virtual]

Returns the SQ length, as read from disk.

Implements [gdcm::Value](#).

27.243.4.12 **SizeType** gdcm::SequenceOfItems::GetNumberOfItems ( ) const [inline]

Examples:

[ChangeSequenceUltrasound.cxx](#), [DumpExamCard.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECHO.cxx](#), [ExtractEncryptedContent.cxx](#), [gdcmrtionplan.cxx](#), [gdcmrtplan.cxx](#), [GetSequenceUltrasound.cxx](#), and [LargeVRD↵SExplicit.cxx](#).

27.243.4.13 **bool** gdcm::SequenceOfItems::IsUndefinedLength ( ) const [inline]

return if [Value](#) Length if of undefined length

27.243.4.14 **static SmartPointer<SequenceOfItems>** gdcm::SequenceOfItems::New ( ) [inline],[static]

27.243.4.15 **SequenceOfItems&** gdcm::SequenceOfItems::operator= ( const **SequenceOfItems** & val ) [inline]

References Items, and SequenceLengthField.

27.243.4.16 **bool** gdcm::SequenceOfItems::operator== ( const **Value** & val ) const [inline],[virtual]

Implements [gdcm::Value](#).

References Items, and SequenceLengthField.

27.243.4.17 **void** gdcm::SequenceOfItems::Print ( std::ostream & os ) const [inline],[virtual]

Reimplemented from [gdcm::Object](#).

27.243.4.18 `template<typename TDE , typename TSwap > std::istream& gdcmm::SequenceOfItems::Read ( std::istream & is, bool readvalues = true ) [inline]`

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

References `gdcmm::Item::Clear()`, `gdcmmDebugMacro`, `gdcmmWarningMacro`, `gdcmm::Exception::GetDescription()`, `gdcmm::Item::GetNestedDataSet()`, `gdcmm::DataElement::GetTag()`, `gdcmm::DataElement::GetVL()`, `gdcmm::Item::Read()`, and `gdcmm::DataSet::Size()`.

27.243.4.19 `void gdcmm::SequenceOfItems::SetLength ( VL length ) [inline],[virtual]`

Sets the actual SQ length.

Implements [gdcmm::Value](#).

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

27.243.4.20 `void gdcmm::SequenceOfItems::SetLengthToUndefined ( )`

Properly set the Sequence of [Item](#) to be undefined length.

Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [GenAllIVR.cxx](#), [GenLongSeqs.cxx](#), and [GenSeqs.cxx](#).

27.243.4.21 `void gdcmm::SequenceOfItems::SetNumberOfItems ( SizeType n ) [inline]`

27.243.4.22 `template<typename TDE , typename TSwap > std::ostream const& gdcmm::SequenceOfItems::Write ( std::ostream & os ) const [inline]`

References `gdcmm::VL::Write()`, and `gdcmm::Tag::Write()`.

## 27.243.5 Member Data Documentation

### 27.243.5.1 ItemVector `gdcmm::SequenceOfItems::Items`

Vector of Sequence Items.

Referenced by `operator=()`, and `operator==()`.

### 27.243.5.2 VL `gdcmm::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:

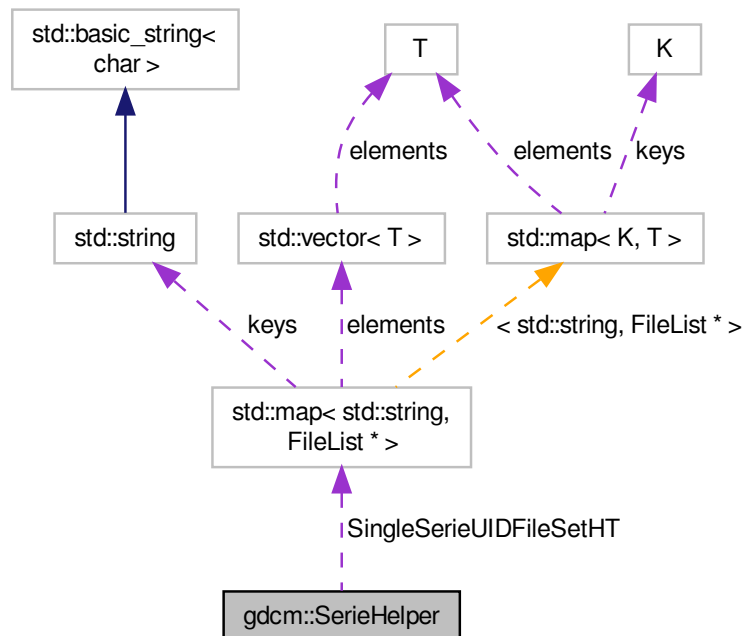
- [gdcmmSequenceOfItems.h](#)

## 27.244 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 disappear 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](#)

## 27.244.1 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 [gdcm::ImageHelper](#) or [gdcm::IPPSorter](#)

## 27.244.2 Member Typedef Documentation

27.244.2.1 typedef std::vector<[Rule](#)> [gdcm::SerieHelper::SerieRestrictions](#) [protected]

27.244.2.2 typedef std::map<std::string, [FileList](#) \*> [gdcm::SerieHelper::SingleSerieUIDFileSetmap](#) [protected]

## 27.244.3 Constructor & Destructor Documentation

27.244.3.1 [gdcm::SerieHelper::SerieHelper](#) ( )

27.244.3.2 [gdcm::SerieHelper::~~SerieHelper](#) ( )

## 27.244.4 Member Function Documentation

27.244.4.1 bool [gdcm::SerieHelper::AddFile](#) ( [FileWithName](#) & *header* ) [protected]



- 27.244.4.2 void gdcm::SerieHelper::AddFileName ( std::string const & *filename* ) [protected]
- 27.244.4.3 void gdcm::SerieHelper::AddRestriction ( const std::string & *tag* )
- 27.244.4.4 void gdcm::SerieHelper::AddRestriction ( uint16\_t *group*, uint16\_t *elem*, std::string const & *value*, int *op* )
- 27.244.4.5 void gdcm::SerieHelper::AddRestriction ( const Tag & *tag* ) [protected]
- 27.244.4.6 void gdcm::SerieHelper::Clear ( )
- 27.244.4.7 void gdcm::SerieHelper::CreateDefaultUniqueSeriesIdentifier ( )
- 27.244.4.8 std::string gdcm::SerieHelper::CreateUniqueSeriesIdentifier ( File \* *inFile* )
- 27.244.4.9 bool gdcm::SerieHelper::FileNameOrdering ( FileList \* *fileList* ) [protected]
- 27.244.4.10 FileList\* gdcm::SerieHelper::GetFirstSingleSerieUIDFileSet ( )
- 27.244.4.11 FileList\* gdcm::SerieHelper::GetNextSingleSerieUIDFileSet ( )
- 27.244.4.12 bool gdcm::SerieHelper::ImagePositionPatientOrdering ( FileList \* *fileSet* ) [protected]
- 27.244.4.13 void gdcm::SerieHelper::OrderFileList ( FileList \* *fileSet* )
- 27.244.4.14 void gdcm::SerieHelper::SetDirectory ( std::string const & *dir*, bool *recursive* = false )
- 27.244.4.15 void gdcm::SerieHelper::SetLoadMode ( int ) [inline]
- 27.244.4.16 void gdcm::SerieHelper::SetUseSeriesDetails ( bool *useSeriesDetails* )
- 27.244.4.17 bool gdcm::SerieHelper::UserOrdering ( FileList \* *fileSet* ) [protected]

## 27.244.5 Member Data Documentation

- 27.244.5.1 SingleSerieUIDFileSetmap::iterator gdcm::SerieHelper::ItFileSetHt [protected]
- 27.244.5.2 SingleSerieUIDFileSetmap gdcm::SerieHelper::SingleSerieUIDFileSetHT [protected]

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

- [gdcmSerieHelper.h](#)

## 27.245 gdcm::Series Class Reference

[Series](#).

```
#include <gdcmSeries.h>
```

### Public Member Functions

- [Series](#) ()

### 27.245.1 Detailed Description

[Series](#).

### 27.245.2 Constructor & Destructor Documentation

#### 27.245.2.1 `gdcmm::Series::Series ( ) [inline]`

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

- [gdcmmSeries.h](#)

## 27.246 `gdcmm::network::ServiceClassApplicationInformation` Class Reference

```
#include <gdcmmServiceClassApplicationInformation.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

### 27.246.1 Detailed Description

PS 3.4 [Table B.3-1](#) SERVICE-CLASS-APPLICATION-INFORMATION (A-ASSOCIATE-RQ)

### 27.246.2 Constructor & Destructor Documentation

#### 27.246.2.1 `gdcmm::network::ServiceClassApplicationInformation::ServiceClassApplicationInformation ( )`

### 27.246.3 Member Function Documentation

#### 27.246.3.1 `void gdcmm::network::ServiceClassApplicationInformation::Print ( std::ostream & os ) const`

#### 27.246.3.2 `std::istream& gdcmm::network::ServiceClassApplicationInformation::Read ( std::istream & is )`

#### 27.246.3.3 `void gdcmm::network::ServiceClassApplicationInformation::SetTuple ( uint8_t levelofsupport, uint8_t levelofdigitalsig, uint8_t elementcoercion )`

#### 27.246.3.4 `size_t gdcmm::network::ServiceClassApplicationInformation::Size ( ) const`

#### 27.246.3.5 `const std::ostream& gdcmm::network::ServiceClassApplicationInformation::Write ( std::ostream & os ) const`

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

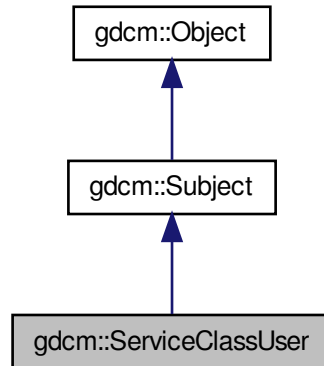
- [gdcmmServiceClassApplicationInformation.h](#)

## 27.247 gdcm::ServiceClassUser Class Reference

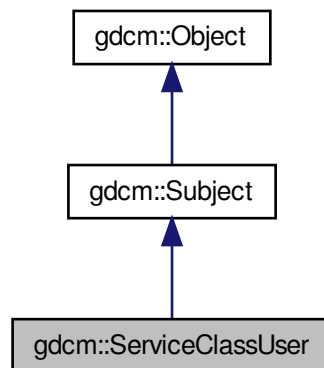
[ServiceClassUser](#).

```
#include <gdcmServiceClassUser.h>
```

Inheritance diagram for gdcm::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

### 27.247.1 Detailed Description

[ServiceClassUser](#).

Examples:

[CStoreQtProgress.cxx](#).

### 27.247.2 Constructor & Destructor Documentation

#### 27.247.2.1 gdcm::ServiceClassUser::ServiceClassUser ( )

Construct a SCU with default:

- hostname = localhost
- port = 104

#### 27.247.2.2 gdcm::ServiceClassUser::~~ServiceClassUser ( )

### 27.247.3 Member Function Documentation

#### 27.247.3.1 const char\* gdcm::ServiceClassUser::GetAETitle ( ) const

#### 27.247.3.2 const char\* gdcm::ServiceClassUser::GetCalledAETitle ( ) const

#### 27.247.3.3 double gdcm::ServiceClassUser::GetTimeout ( ) const

#### 27.247.3.4 bool gdcm::ServiceClassUser::InitializeConnection ( )

Will try to connect This will setup the actual timeout used during the whole connection time. Need to call SetTimeout first

Examples:

[CStoreQtProgress.cxx](#).

#### 27.247.3.5 bool gdcm::ServiceClassUser::IsPresentationContextAccepted ( const PresentationContext & pc ) const

Return if the passed in presentation was accepted during association negotiation.

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

for wrapped language: instantiate a reference counted object

#### 27.247.3.7 bool gdcm::ServiceClassUser::SendEcho ( )

C-ECHO.

27.247.3.8 `bool gdcm::ServiceClassUser::SendFind ( const BaseRootQuery * query, std::vector< DataSet > & retDatasets )`

C-FIND a query, return result are in retDatasets.

27.247.3.9 `bool gdcm::ServiceClassUser::SendMove ( const BaseRootQuery * query, const char * outputdir )`

Execute a C-MOVE, based on query, return files are written in outputdir.

27.247.3.10 `bool gdcm::ServiceClassUser::SendMove ( const BaseRootQuery * query, std::vector< DataSet > & retDatasets )`

Execute a C-MOVE, based on query, returned dataset are Implicit.

27.247.3.11 `bool gdcm::ServiceClassUser::SendMove ( const BaseRootQuery * query, std::vector< File > & retFile )`

Execute a C-MOVE, based on query, returned Files are stored in vector.

27.247.3.12 `bool gdcm::ServiceClassUser::SendStore ( const char * filename )`

Execute a C-STORE on file on disk, named filename.

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.13 `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.

27.247.3.14 `bool gdcm::ServiceClassUser::SendStore ( DataSet const & ds )`

Execute a C-STORE on a [DataSet](#), the transfer syntax used will be Implicit.

27.247.3.15 `void gdcm::ServiceClassUser::SetAETitle ( const char * aetitle )`

set calling ae title

27.247.3.16 `void gdcm::ServiceClassUser::SetCalledAETitle ( const char * aetitle )`

set called ae title

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.17 void gdcm::ServiceClassUser::SetHostname ( const char \* *hostname* )

Set the name of the called hostname (hostname or IP address)

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.18 void gdcm::ServiceClassUser::SetPort ( uint16\_t *port* )

Set port of remote host (called application)

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.19 void gdcm::ServiceClassUser::SetPortSCP ( uint16\_t *portscp* )

Set the port for any incoming C-STORE-SCP operation (typically in a return of C-MOVE)

27.247.3.20 void gdcm::ServiceClassUser::SetPresentationContexts ( std::vector< PresentationContext > const & *pcs* )

Set the Presentation Context used for the Association.

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.21 void gdcm::ServiceClassUser::SetTimeout ( double *t* )

set/get Timeout

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.22 bool gdcm::ServiceClassUser::StartAssociation ( )

Start the association. Need to call SetPresentationContexts before.

Examples:

[CStoreQtProgress.cxx](#).

27.247.3.23 `bool gdcM::ServiceClassUser::StopAssociation ( )`

Stop the running association.

Examples:

[CStoreQtProgress.cxx](#).

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

- [gdcMServiceClassUser.h](#)

## 27.248 gdcM::SHA1 Class Reference

Class for [SHA1](#).

```
#include <gdcMSHA1.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])

### 27.248.1 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

### 27.248.2 Constructor & Destructor Documentation

27.248.2.1 `gdcM::SHA1::SHA1 ( )`

27.248.2.2 `gdcM::SHA1::~~SHA1 ( )`

### 27.248.3 Member Function Documentation



27.248.3.1 static bool gdcM::SHA1::Compute ( const char \* *buffer*, unsigned long *buf\_len*, char *digest\_str*[20\*2+1] )  
[static]

27.248.3.2 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:

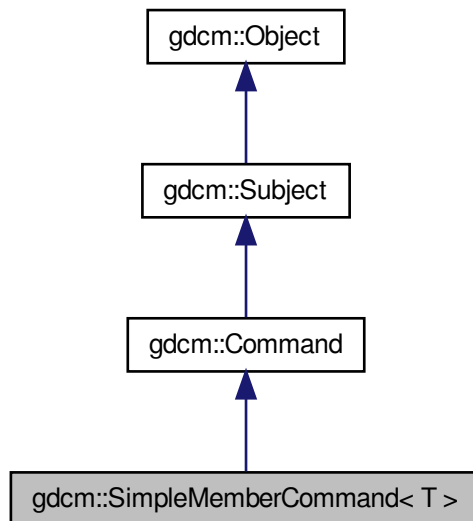
- [gdcMSHA1.h](#)

## 27.249 gdcM::SimpleMemberCommand< T > Class Template Reference

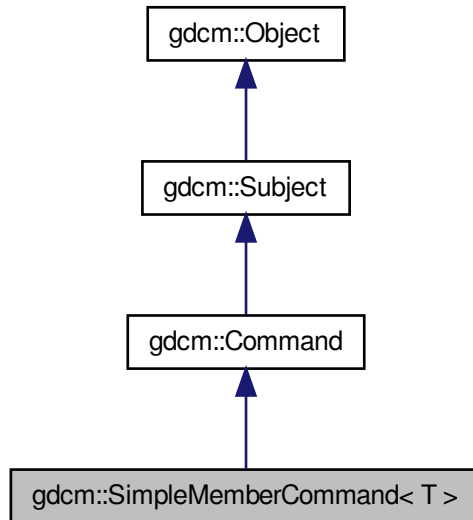
[Command](#) subclass that calls a pointer to a member function.

```
#include <gdcMCommand.h>
```

Inheritance diagram for gdcM::SimpleMemberCommand< T >:



Collaboration diagram for `gdc::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](#)

### 27.249.1 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.

### 27.249.2 Member Typedef Documentation

27.249.2.1 `template<typename T > typedef SimpleMemberCommand gdcM::SimpleMemberCommand< T >::Self`

Standard class typedefs.

27.249.2.2 `template<typename T > typedef void(T::* gdcM::SimpleMemberCommand< T >::TMemberFunctionPointer)()`

A method callback.

### 27.249.3 Constructor & Destructor Documentation

27.249.3.1 `template<typename T > gdcM::SimpleMemberCommand< T >::SimpleMemberCommand ( )`  
`[inline], [protected]`

Referenced by `gdcM::SimpleMemberCommand< T >::New()`.

27.249.3.2 `template<typename T > virtual gdcM::SimpleMemberCommand< T >::~~SimpleMemberCommand ( )`  
`[inline], [protected], [virtual]`

### 27.249.4 Member Function Documentation

27.249.4.1 `template<typename T > virtual void gdcM::SimpleMemberCommand< T >::Execute ( Subject *, const Event & )` `[inline], [virtual]`

Invoke the callback function.

Implements [gdcM::Command](#).

References `gdcM::SimpleMemberCommand< T >::m_MemberFunction`.

27.249.4.2 `template<typename T > virtual void gdcM::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 [gdcM::Command](#).

References `gdcmm::SimpleMemberCommand< T >::m_MemberFunction`.

**27.249.4.3** `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.

References `gdcmm::SimpleMemberCommand< T >::SimpleMemberCommand()`.

**27.249.4.4** `template<typename T > void gdcmm::SimpleMemberCommand< T >::SetCallbackFunction ( T * object, TMemberFunctionPointer memberFunction ) [inline]`

Specify the callback function.

References `gdcmm::SimpleMemberCommand< T >::m_MemberFunction`, and `gdcmm::SimpleMemberCommand< T >::m_This`.

## 27.249.5 Member Data Documentation

**27.249.5.1** `template<typename T > TMemberFunctionPointer gdcmm::SimpleMemberCommand< T >::m_MemberFunction [protected]`

Referenced by `gdcmm::SimpleMemberCommand< T >::Execute()`, and `gdcmm::SimpleMemberCommand< T >::SetCallbackFunction()`.

**27.249.5.2** `template<typename T > T* gdcmm::SimpleMemberCommand< T >::m_This [protected]`

Referenced by `gdcmm::SimpleMemberCommand< T >::SetCallbackFunction()`.

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

- [gdcmmCommand.h](#)

## 27.250 gdcmm::SimpleSubjectWatcher Class Reference

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

```
#include <gdcmmSimpleSubjectWatcher.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](#) ()

### 27.250.1 Detailed Description

[SimpleSubjectWatcher](#) This is a typical [Subject](#) Watcher class. It will observe all events.

Examples:

[SimpleScanner.cxx](#).

### 27.250.2 Constructor & Destructor Documentation

27.250.2.1 `gdcmm::SimpleSubjectWatcher::SimpleSubjectWatcher ( Subject * s, const char * comment = " " )`

27.250.2.2 `virtual gdcmm::SimpleSubjectWatcher::~SimpleSubjectWatcher ( )` [virtual]

### 27.250.3 Member Function Documentation

27.250.3.1 `virtual void gdcmm::SimpleSubjectWatcher::EndFilter ( )` [protected], [virtual]

27.250.3.2 `virtual void gdcmm::SimpleSubjectWatcher::ShowAbort ( )` [protected], [virtual]

27.250.3.3 `virtual void gdcmm::SimpleSubjectWatcher::ShowAnonymization ( Subject * caller, const Event & evt )` [protected], [virtual]

27.250.3.4 `virtual void gdcmm::SimpleSubjectWatcher::ShowData ( Subject * caller, const Event & evt )` [protected], [virtual]

27.250.3.5 `virtual void gdcmm::SimpleSubjectWatcher::ShowDataSet ( Subject * caller, const Event & evt )` [protected], [virtual]

27.250.3.6 `virtual void gdcmm::SimpleSubjectWatcher::ShowFileName ( Subject * caller, const Event & evt )` [protected], [virtual]

Examples:

[SimpleScanner.cxx](#).

27.250.3.7 `virtual void gdcmm::SimpleSubjectWatcher::ShowIteration ( )` [protected], [virtual]

27.250.3.8 `virtual void gdcmm::SimpleSubjectWatcher::ShowProgress ( Subject * caller, const Event & evt )` [protected], [virtual]

27.250.3.9    `virtual void gdcM::SimpleSubjectWatcher::StartFilter ( ) [protected],[virtual]`

27.250.3.10   `void gdcM::SimpleSubjectWatcher::TestAbortOff ( ) [protected]`

27.250.3.11   `void gdcM::SimpleSubjectWatcher::TestAbortOn ( ) [protected]`

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

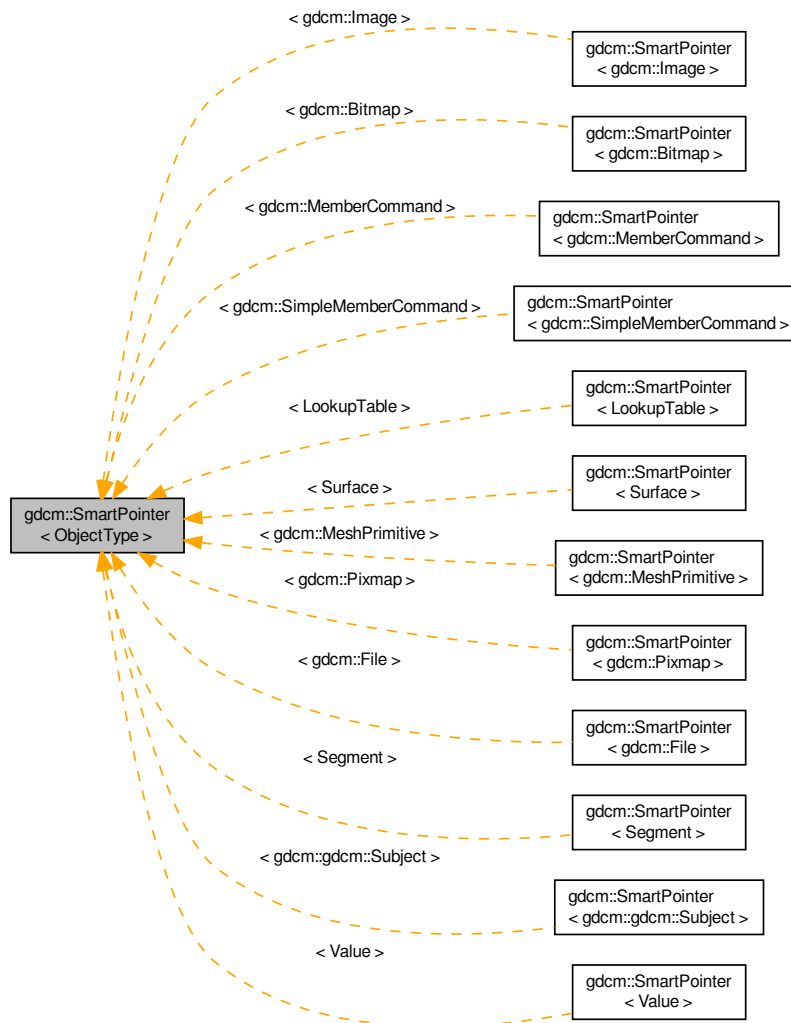
- [gdcMSimpleSubjectWatcher.h](#)

## 27.251   `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)`

### 27.251.1 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/smarterp.htm>

and `itk::SmartPointer`

#### Examples:

[ChangeSequenceUltrasound.cxx](#), [CStoreQtProgress.cxx](#), [DumpGEMSMovieGroup.cxx](#), [DumpPhilipsECGHO.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetSubSequenceData.cxx](#), [LargeVRDSExplicit.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), and [SimpleScanner.cxx](#).

### 27.251.2 Constructor & Destructor Documentation

27.251.2.1 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer ( ) [inline]`

27.251.2.2 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer ( const SmartPointer< ObjectType > & p ) [inline]`

27.251.2.3 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer ( ObjectType * p ) [inline]`

27.251.2.4 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::SmartPointer ( ObjectType const & p ) [inline]`

27.251.2.5 `template<class ObjectType> gdcmm::SmartPointer< ObjectType >::~~SmartPointer ( ) [inline]`

### 27.251.3 Member Function Documentation



27.251.3.1 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::GetPointer ( ) const`  
`[inline]`

Explicit function to retrieve the pointer.

27.251.3.2 `template<class ObjectType> gdcm::SmartPointer< ObjectType >::operator ObjectType * ( ) const`  
`[inline]`

Return pointer to object.

27.251.3.3 `template<class ObjectType> ObjectType& gdcm::SmartPointer< ObjectType >::operator* ( ) const`  
`[inline]`

27.251.3.4 `template<class ObjectType> ObjectType* gdcm::SmartPointer< ObjectType >::operator-> ( ) const`  
`[inline]`

Overload operator ->

27.251.3.5 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= ( SmartPointer< ObjectType > const & r )` `[inline]`

Overload operator assignment.

Referenced by `gdcm::SmartPointer< Value >::operator=()`.

27.251.3.6 `template<class ObjectType> SmartPointer& gdcm::SmartPointer< ObjectType >::operator= ( ObjectType * r )`  
`[inline]`

Overload operator assignment.

27.251.3.7 `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](#)

## 27.252 gdcm::network::SOPClassExtendedNegociationSub Class Reference

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

```
#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

### 27.252.1 Detailed Description

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

### 27.252.2 Constructor & Destructor Documentation

27.252.2.1 `gdcm::network::SOPClassExtendedNegociationSub::SOPClassExtendedNegociationSub ( )`

### 27.252.3 Member Function Documentation

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

27.252.3.2 `std::istream& gdcm::network::SOPClassExtendedNegociationSub::Read ( std::istream & is )`

27.252.3.3 `void gdcm::network::SOPClassExtendedNegociationSub::SetTuple ( const char * uid, uint8_t levelofsupport = 3, uint8_t levelofdigitalsig = 0, uint8_t elementcoercion = 2 )`

27.252.3.4 `size_t gdcm::network::SOPClassExtendedNegociationSub::Size ( ) const`

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

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

- [gdcmSOPClassExtendedNegociationSub.h](#)

## 27.253 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](#) ()

### 27.253.1 Detailed Description

Class convert a class SOP Class UID into [IOD](#).

Reference PS 3.4 [Table](#) B.5-1 STANDARD SOP CLASSES

### 27.253.2 Member Typedef Documentation

27.253.2.1 `typedef const char* gdcm::SOPClassUIDToIOD::const(SOPClassUIDToIODType)[2]`

### 27.253.3 Member Function Documentation

27.253.3.1 `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](#).

27.253.3.2 `static const char* gdcm::SOPClassUIDToIOD::GetIODFromSOPClassUID ( const char * sopclassuid ) [static]`

27.253.3.3 `static unsigned int gdcm::SOPClassUIDToIOD::GetNumberOfSOPClassToIOD ( ) [static]`

Return the number of SOP Class UID listed internally.

27.253.3.4 `static const char* gdcm::SOPClassUIDToIOD::GetSOPClassUIDFromIOD ( const char * iod ) [static]`

27.253.3.5 `static SOPClassUIDToIODType& gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIOD ( unsigned int i ) [static]`

27.253.3.6 `static SOPClassUIDToIODType* gdcm::SOPClassUIDToIOD::GetSOPClassUIDToIODs ( ) [static]`

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

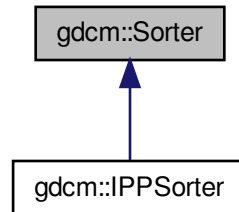
- [gdcmSOPClassUIDToIOD.h](#)

## 27.254 gdcm::Sorter Class Reference

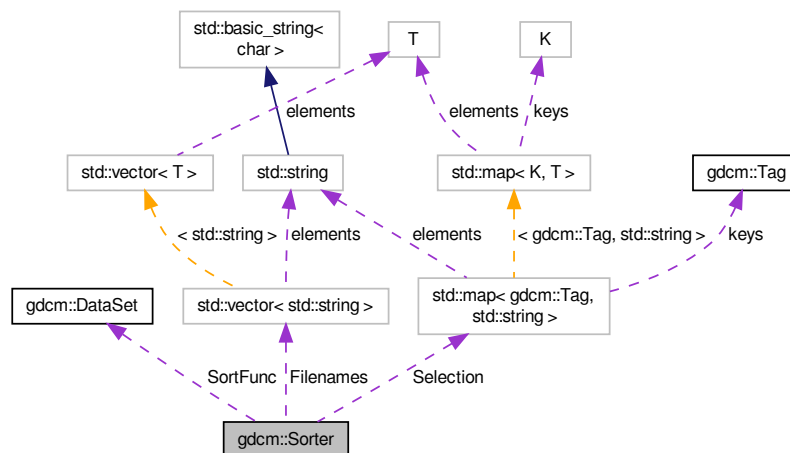
[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::SortFunction](#).

```
#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`)
- virtual bool `Sort` (`std::vector< std::string > const &filenames`)  
*Typically the output of `gdcmm::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`

## Friends

- `std::ostream & operator<<` (`std::ostream &_os, const Sorter &s`)

### 27.254.1 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`.

### 27.254.2 Member Typedef Documentation

27.254.2.1 typedef `std::map<Tag,std::string>` `gdcmm::Sorter::SelectionMap` [protected]

27.254.2.2 typedef `bool(* gdcmm::Sorter::SortFunction)(DataSet const &, DataSet const &)`

Set the sort function which compares one dataset to the other.

### 27.254.3 Constructor & Destructor Documentation

27.254.3.1 `gdcm::Sorter::Sorter ( )`

27.254.3.2 `virtual gdcm::Sorter::~~Sorter ( )` `[virtual]`

### 27.254.4 Member Function Documentation

27.254.4.1 `bool gdcm::Sorter::AddSelect ( Tag const & tag, const char * value )`

UNSUPPORTED FOR NOW.

27.254.4.2 `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:

[gdcmorthoplanes.cxx](#), [reslicesphere.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.254.4.3 `void gdcm::Sorter::Print ( std::ostream & os ) const`

Print.

Examples:

[gdcmorthoplanes.cxx](#), [SortImage.cxx](#), and [VolumeSorter.cxx](#).

Referenced by `gdcm::operator<<()`.

27.254.4.4 `void gdcm::Sorter::SetSortFunction ( SortFunction f )`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

27.254.4.5 `virtual bool gdcm::Sorter::Sort ( std::vector< std::string > const & filenames )` `[virtual]`

Typically the output of `gdcm::Directory::GetFileNames()`

Reimplemented in `gdcm::IPPSorter`.

Examples:

[SortImage.cxx](#).

27.254.4.6 `virtual bool gdcm::Sorter::StableSort ( std::vector< std::string > const & filenames )` `[virtual]`

Examples:

[SortImage.cxx](#), and [VolumeSorter.cxx](#).

### 27.254.5 Friends And Related Function Documentation

27.254.5.1 `std::ostream& operator<< ( std::ostream & _os, const Sorter & s )` `[friend]`

### 27.254.6 Member Data Documentation

27.254.6.1 `std::vector<std::string> gdcm::Sorter::FileNames` `[protected]`

27.254.6.2 `std::map<Tag,std::string> gdcm::Sorter::Selection` `[protected]`

27.254.6.3 **SortFunction** `gdcm::Sorter::SortFunc` `[protected]`

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

- [gdcmSorter.h](#)

## 27.255 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)

### 27.255.1 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)

## 27.255.2 Member Enumeration Documentation

### 27.255.2.1 enum gdcm::Spacing::SpacingType

Enumerator

**DETECTOR**  
**MAGNIFIED**  
**CALIBRATED**  
**UNKNOWN**

## 27.255.3 Constructor & Destructor Documentation

### 27.255.3.1 gdcm::Spacing::Spacing ( )

### 27.255.3.2 gdcm::Spacing::~~Spacing ( )

## 27.255.4 Member Function Documentation

### 27.255.4.1 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](#)

## 27.256 gdcm::Spectroscopy Class Reference

[Spectroscopy](#) class.

```
#include <gdcmSpectroscopy.h>
```

### Public Member Functions

- [Spectroscopy](#) ()

#### 27.256.1 Detailed Description

[Spectroscopy](#) class.

#### 27.256.2 Constructor & Destructor Documentation

27.256.2.1 [gdcm::Spectroscopy::Spectroscopy](#) ( ) [inline]

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

- [gdcmSpectroscopy.h](#)

## 27.257 gdcm::SplitMosaicFilter Class Reference

[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.

```
#include <gdcmSplitMosaicFilter.h>
```

### Public Member Functions

- [SplitMosaicFilter](#) ()
- [~SplitMosaicFilter](#) ()
- bool [ComputeMOSAICDimensions](#) (unsigned int dims[3])
- [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.*

### 27.257.1 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.

### 27.257.2 Constructor & Destructor Documentation

27.257.2.1 `gdcm::SplitMosaicFilter::SplitMosaicFilter ( )`

27.257.2.2 `gdcm::SplitMosaicFilter::~~SplitMosaicFilter ( )`

### 27.257.3 Member Function Documentation

27.257.3.1 `bool gdcm::SplitMosaicFilter::ComputeMOSAICDimensions ( unsigned int dims[3] )`

Compute the new dimensions according to private information stored in the MOSAIC header.

27.257.3.2 `File& gdcm::SplitMosaicFilter::GetFile ( )` `[inline]`

27.257.3.3 `const File& gdcm::SplitMosaicFilter::GetFile ( ) const` `[inline]`

27.257.3.4 `const Image& gdcm::SplitMosaicFilter::GetImage ( ) const` `[inline]`

27.257.3.5 `Image& gdcm::SplitMosaicFilter::GetImage ( )` `[inline]`

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

27.257.3.7 `void gdcm::SplitMosaicFilter::SetImage ( const Image & image )`

27.257.3.8 `bool gdcm::SplitMosaicFilter::Split ( )`

Split the SIEMENS MOSAIC image.

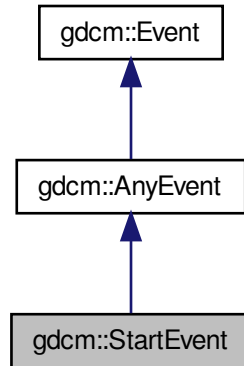
The documentation for this class was generated from the following file:

- [gdcmSplitMosaicFilter.h](#)

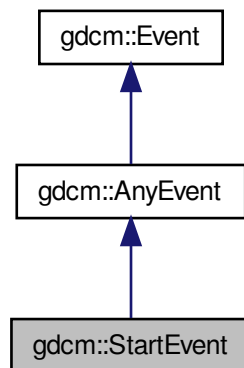
## 27.258 gdcm::StartEvent Class Reference

```
#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](#)

## 27.259 `gdcm::static_assert_test< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

## 27.260 `gdcm::STATIC_ASSERTION_FAILURE< x >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

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

- [gdcmStaticAssert.h](#)

## 27.261 `gdcm::STATIC_ASSERTION_FAILURE< true >` Struct Template Reference

```
#include <gdcmStaticAssert.h>
```

### Public Types

- enum { [value](#) = 1 }

### 27.261.1 Member Enumeration Documentation

#### 27.261.1.1 anonymous enum

Enumerator

***value***

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

- [gdcmStaticAssert.h](#)

## 27.262 `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)

### 27.262.1 Detailed Description

[StreamImageReader](#).

#### Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::Image](#) representation via an ITK streaming (ie, multithreaded) interface [Image](#) is different from [Pixmap](#) has it has a position and a direction in Space. Currently, this class is thread safe in that it can read a single extent in a single thread. Multiple versions can be used for multiple extents/threads.

#### See also

[Image](#)

#### Examples:

[ExtractOneFrame.cs](#), and [StreamImageReaderTest.cxx](#).

### 27.262.2 Constructor & Destructor Documentation

27.262.2.1 [gdcm::StreamImageReader::StreamImageReader \( \)](#)

27.262.2.2 [virtual gdcm::StreamImageReader::~~StreamImageReader \( \)](#) `[virtual]`

### 27.262.3 Member Function Documentation

27.262.3.1 [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](#).

27.262.3.2 [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](#).

**27.262.3.3** `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](#).

**27.262.3.4** `std::vector<unsigned int> gdcM::StreamImageReader::GetDimensionsValueForResolution ( unsigned int )`

**27.262.3.5** `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](#).

**27.262.3.6** `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](#).

**27.262.3.7** `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](#).

27.262.3.8 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:

[StreamImageReaderTest.cxx](#).

27.262.3.9 void gdcm::StreamImageReader::SetStream ( std::istream & *inStream* )

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

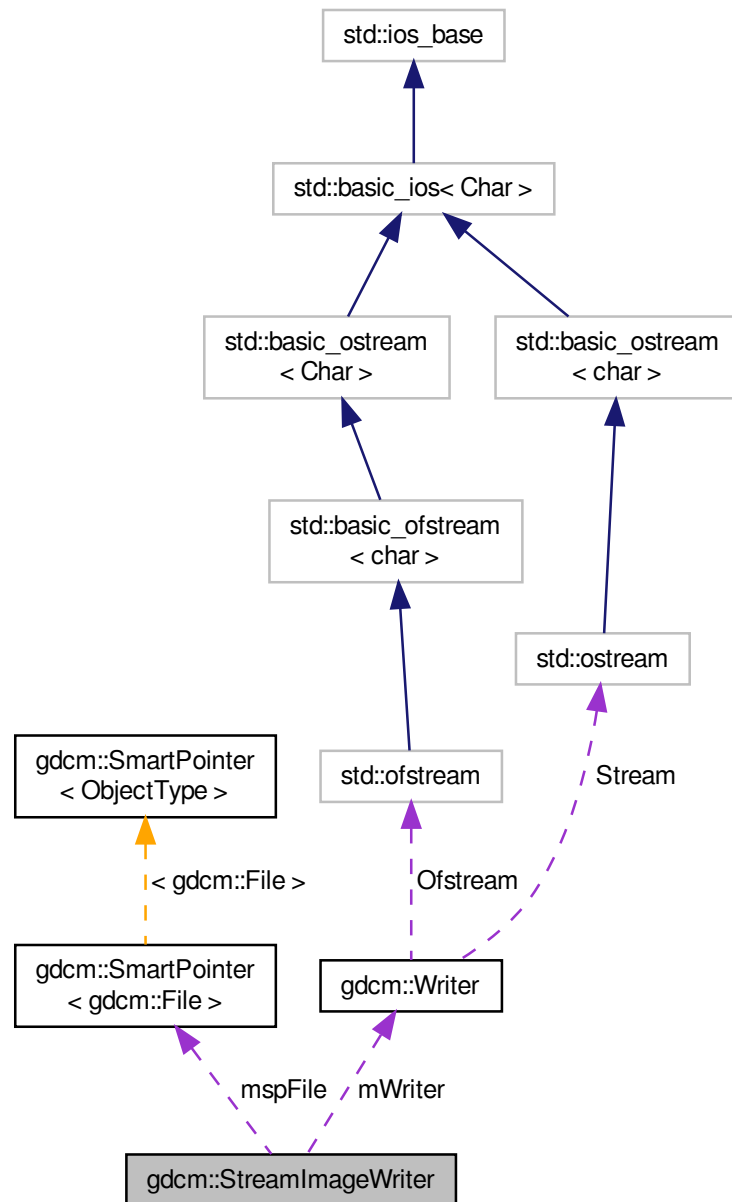
- [gdcmStreamImageReader.h](#)

## 27.263 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](#)

## 27.263.1 Detailed Description

[StreamImageReader](#).

### Note

its role is to convert the DICOM [DataSet](#) into a [gdcm::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](#).

## 27.263.2 Constructor & Destructor Documentation

27.263.2.1 `gdcm::StreamImageWriter::StreamImageWriter ( )`

27.263.2.2 `virtual gdcm::StreamImageWriter::~~StreamImageWriter ( )` [virtual]

## 27.263.3 Member Function Documentation

27.263.3.1 `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](#).

27.263.3.2 `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](#).

27.263.3.3 `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](#).

27.263.3.4 `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](#).

27.263.3.5 `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.

27.263.3.6 void gdcm::StreamImageWriter::SetStream ( std::ostream & *inStream* )

Examples:

[Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

27.263.3.7 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](#).

27.263.3.8 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](#).

27.263.3.9 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

27.263.3.10 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.

### 27.263.4 Member Data Documentation

27.263.4.1 `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.

27.263.4.2 `int gdcM::StreamImageWriter::mElementOffsets1` `[protected]`

27.263.4.3 `SmartPointer<File> gdcM::StreamImageWriter::mspFile` `[protected]`

27.263.4.4 `Writer gdcM::StreamImageWriter::mWriter` `[protected]`

27.263.4.5 `uint16_t gdcM::StreamImageWriter::mXMax` `[protected]`

27.263.4.6 `uint16_t gdcM::StreamImageWriter::mXMin` `[protected]`

27.263.4.7 `uint16_t gdcM::StreamImageWriter::mYMax` `[protected]`

27.263.4.8 `uint16_t gdcM::StreamImageWriter::mYMin` `[protected]`

27.263.4.9 `uint16_t gdcM::StreamImageWriter::mZMax` `[protected]`

27.263.4.10 `uint16_t gdcM::StreamImageWriter::mZMin` `[protected]`

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

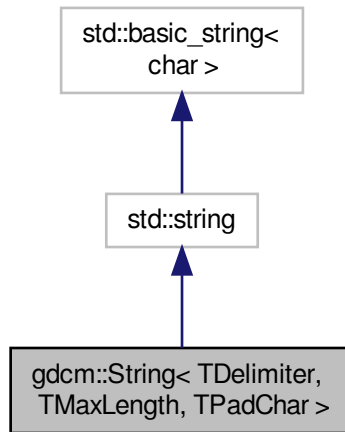
- [gdcMStreamImageWriter.h](#)

## 27.264 `gdcM::String< TDelimiter, TMaxLength, TPadChar >` Class Template Reference

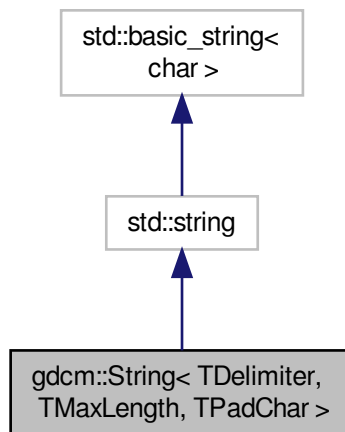
[String.](#)

```
#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
- [gdcmm::String](#)< TDelimiter,  
TMaxLength, TPadChar > [Truncate](#) () const

## Static Public Member Functions

- static std::string [Trim](#) (const char \*input)

### 27.264.1 Detailed Description

```
template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '>class gdcmm::String< TDelimiter, TMaxLength, TPadChar >
```

[String](#).

#### Note

TDelimiter template parameter is used to separate multiple [String](#) (VM1 >) TMaxLength is only a hint. Noone actually respect the max length TPadChar is the string padding (0 or space)

### 27.264.2 Member Typedef Documentation

27.264.2.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_iterator gdcmm::String< TDelimiter, TMaxLength, TPadChar >::const_iterator`

27.264.2.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_reference`

27.264.2.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::const_reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::const_reverse_iterator`

27.264.2.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::difference_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::difference_type`

27.264.2.5 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::iterator`

27.264.2.6 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::pointer gdcm::String< TDelimiter, TMaxLength, TPadChar >::pointer`

27.264.2.7 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reference gdcm::String< TDelimiter, TMaxLength, TPadChar >::reference`

27.264.2.8 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::reverse_iterator gdcm::String< TDelimiter, TMaxLength, TPadChar >::reverse_iterator`

27.264.2.9 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::size_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::size_type`

27.264.2.10 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> typedef std::string::value_type gdcm::String< TDelimiter, TMaxLength, TPadChar >::value_type`

### 27.264.3 Constructor & Destructor Documentation

27.264.3.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( ) [inline]`

[String](#) constructors.

27.264.3.2 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( const value_type * s ) [inline]`

27.264.3.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( const value_type * s, size_type n ) [inline]`

27.264.3.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> gdcm::String< TDelimiter, TMaxLength, TPadChar >::String ( const std::string & s, size_type pos = 0, size_type n = npos ) [inline]`

### 27.264.4 Member Function Documentation

27.264.4.1 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = ' '> bool gdcm::String< TDelimiter, TMaxLength, TPadChar >::IsValid ( ) const [inline]`

return if string is valid

Referenced by `gdcm::String< TDelimiter, TMaxLength, TPadChar >::Truncate()`.

27.264.4.2 `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:

27.264.4.3 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> std::string gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Trim ( ) const [inline]`

Trim function is required to return a std::string object, otherwise we could not create a [gdcmm::String](#) object with an odd number of bytes...

27.264.4.4 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> static std::string gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Trim ( const char * input ) [inline], [static]`

27.264.4.5 `template<char TDelimiter = '\\', unsigned int TMaxLength = 64, char TPadChar = '> gdcmm::String< TDelimiter, TMaxLength, TPadChar > gdcmm::String< TDelimiter, TMaxLength, TPadChar >::Truncate ( ) const [inline]`

References `gdcmm::String< TDelimiter, TMaxLength, TPadChar >::IsValid()`.

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

- [gdcmmString.h](#)

## 27.265 gdcmm::StringFilter Class Reference

[StringFilter](#) [StringFilter](#) 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.

```
#include <gdcmmStringFilter.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, [VL](#) const &vl)  
*DEPRECATED: NEVER USE IT.*
- std::string [FromString](#) (const [Tag](#) &t, const char \*value, size\_t 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 [Tag](#) &t) const  
*Convert to string the ByteValue contained in a DataElement.*
- std::pair< std::string, std::string > [ToStringPair](#) (const [Tag](#) &t) const
- 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

### 27.265.1 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](#), and [SimplePrintPatientName.cs](#).

### 27.265.2 Constructor & Destructor Documentation

27.265.2.1 `gdcm::StringFilter::StringFilter ( )`

27.265.2.2 `gdcm::StringFilter::~~StringFilter ( )`

### 27.265.3 Member Function Documentation

27.265.3.1 `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

27.265.3.2 `bool gdcm::StringFilter::ExecuteQuery ( std::string const & query, DataSet const & ds, std::string & value ) const`  
[protected]

27.265.3.3 `std::string gdcm::StringFilter::FromString ( const Tag & t, const char * value, VL const & vl )`

DEPRECATED: NEVER USE IT.

27.265.3.4 `std::string gdcm::StringFilter::FromString ( const Tag & t, const char * value, size_t len )`

27.265.3.5 `File& gdcm::StringFilter::GetFile ( )` [inline]

27.265.3.6 `const File& gdcm::StringFilter::GetFile ( ) const` [inline]

27.265.3.7 `void gdcm::StringFilter::SetDicts ( const Dicts & dicts )`

Allow user to pass in there own dicts.

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

Set/Get [File](#).

Examples:

[ReadAndPrintAttributes.cxx](#).

27.265.3.9 `std::string gdcM::StringFilter::ToString ( const Tag & t ) const`

Convert to string the [ByteValue](#) contained in a [DataElement](#).

Examples:

[ReadAndPrintAttributes.cxx](#).

27.265.3.10 `std::pair<std::string, std::string> gdcM::StringFilter::ToStringPair ( const Tag & t ) 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](#).

27.265.3.11 `std::pair<std::string, std::string> gdcM::StringFilter::ToStringPair ( const Tag & t, DataSet const & ds ) const`  
[protected]

27.265.3.12 `void gdcM::StringFilter::UseDictAlways ( bool )` [inline]

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

- [gdcMStringFilter.h](#)

## 27.266 gdcM::Study Class Reference

[Study](#).

```
#include <gdcMStudy.h>
```

### Public Member Functions

- [Study](#) ()

### 27.266.1 Detailed Description

[Study](#).

## 27.266.2 Constructor & Destructor Documentation

### 27.266.2.1 gdcmm::Study::Study ( ) [inline]

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

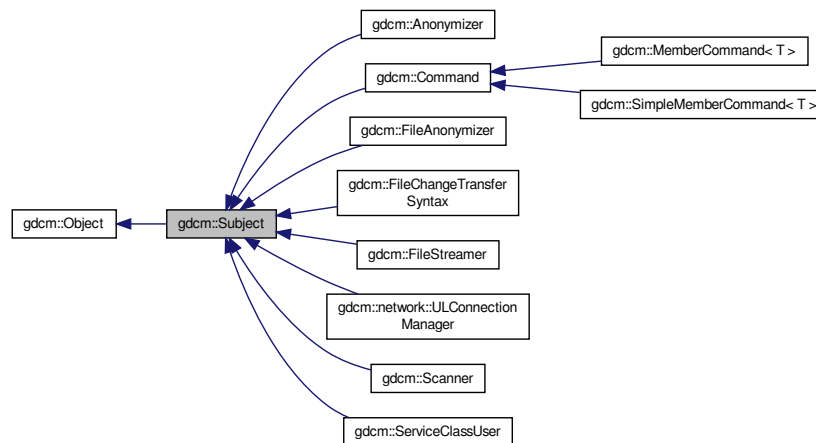
- [gdcmmStudy.h](#)

## 27.267 gdcmm::Subject Class Reference

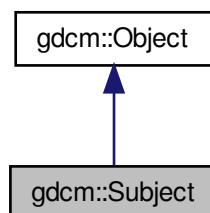
[Subject.](#)

```
#include <gdcmmSubject.h>
```

Inheritance diagram for gdcmm::Subject:



Collaboration diagram for gdcmm::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

### 27.267.1 Detailed Description

[Subject](#).

See also

[Command Event](#)

Examples:

[SimpleScanner.cxx](#).

### 27.267.2 Constructor & Destructor Documentation

27.267.2.1 [gdcmm::Subject::Subject](#) ( )

27.267.2.2 [gdcmm::Subject::~~Subject](#) ( )

### 27.267.3 Member Function Documentation

27.267.3.1 unsigned long [gdcmm::Subject::AddObserver](#) ( const [Event](#) & *event*, [Command](#) \* )

Allow people to add/remove/invoke observers (callbacks) to any GDCM object. This is an implementation of the subject/observer design pattern. An observer is added by specifying an event to respond to and an [gdcmm::Command](#) to execute. It returns an unsigned long tag which can be used later to remove the event or retrieve the command. The memory for the [Command](#) becomes the responsibility of this object, so don't pass the same instance of a command to two different objects

27.267.3.2 unsigned long [gdcmm::Subject::AddObserver](#) ( const [Event](#) & *event*, [Command](#) \* ) const

27.267.3.3 [Command](#)\* [gdcmm::Subject::GetCommand](#) ( unsigned long *tag* )

Get the command associated with the given tag. NOTE: This returns a pointer to a [Command](#), but it is safe to assign this to a [Command::Pointer](#). Since [Command](#) inherits from [LightObject](#), at this point in the code, only a pointer or a reference to the [Command](#) can be used.

27.267.3.4 `bool gdcm::Subject::HasObserver ( const Event & event ) const`

Return true if an observer is registered for this event.

27.267.3.5 `void gdcm::Subject::InvokeEvent ( const Event & )`

Call Execute on all the Commands observing this event id.

27.267.3.6 `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.

27.267.3.7 `void gdcm::Subject::RemoveAllObservers ( )`

Remove all observers .

27.267.3.8 `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:

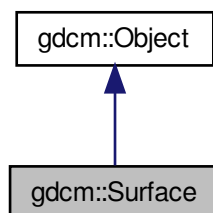
- [gdcmSubject.h](#)

## 27.268 gdcm::Surface Class Reference

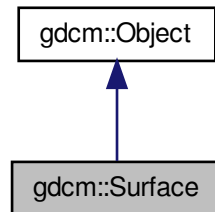
This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.

```
#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

### 27.268.1 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

### 27.268.2 Member Enumeration Documentation

#### 27.268.2.1 enum `gdcm::Surface::STATES`

Enumerator

***NO***  
***YES***  
***UNKNOWN***  
***STATES\_END***

#### 27.268.2.2 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***

### 27.268.3 Constructor & Destructor Documentation

#### 27.268.3.1 `gdcm::Surface::Surface ( )`

#### 27.268.3.2 `virtual gdcm::Surface::~~Surface ( )` [`virtual`]



## 27.268.4 Member Function Documentation

27.268.4.1 **SegmentHelper::BasicCodedEntry** const& gdcm::Surface::GetAlgorithmFamily ( ) const

27.268.4.2 **SegmentHelper::BasicCodedEntry**& gdcm::Surface::GetAlgorithmFamily ( )

27.268.4.3 const char\* gdcm::Surface::GetAlgorithmName ( ) const

27.268.4.4 const char\* gdcm::Surface::GetAlgorithmVersion ( ) const

27.268.4.5 const float\* gdcm::Surface::GetAxisOfRotation ( ) const

Note

Pointer is null if undefined

27.268.4.6 const float\* gdcm::Surface::GetCenterOfRotation ( ) const

Note

Pointer is null if undefined

27.268.4.7 **STATES** gdcm::Surface::GetFiniteVolume ( ) const

27.268.4.8 **STATES** gdcm::Surface::GetManifold ( ) const

27.268.4.9 float gdcm::Surface::GetMaximumPointDistance ( ) const

27.268.4.10 float gdcm::Surface::GetMeanPointDistance ( ) const

27.268.4.11 **MeshPrimitive** const& gdcm::Surface::GetMeshPrimitive ( ) const

27.268.4.12 **MeshPrimitive**& gdcm::Surface::GetMeshPrimitive ( )

27.268.4.13 unsigned long gdcm::Surface::GetNumberOfSurfacePoints ( ) const

27.268.4.14 unsigned long gdcm::Surface::GetNumberOfVectors ( ) const

27.268.4.15 const **DataElement**& gdcm::Surface::GetPointCoordinatesData ( ) const

27.268.4.16 **DataElement**& gdcm::Surface::GetPointCoordinatesData ( )

27.268.4.17 const float\* gdcm::Surface::GetPointPositionAccuracy ( ) const

Note

Pointer is null if undefined

27.268.4.18 const float\* gdcm::Surface::GetPointsBoundingBoxCoordinates ( ) const

## Note

Pointer is null if undefined

- 27.268.4.19 **SegmentHelper::BasicCodedEntry** const& gdcmm::Surface::GetProcessingAlgorithm ( ) const
- 27.268.4.20 **SegmentHelper::BasicCodedEntry**& gdcmm::Surface::GetProcessingAlgorithm ( )
- 27.268.4.21 const unsigned short\* gdcmm::Surface::GetRecommendedDisplayCIELabValue ( ) const
- 27.268.4.22 unsigned short gdcmm::Surface::GetRecommendedDisplayCIELabValue ( const unsigned int *idx* ) const
- 27.268.4.23 unsigned short gdcmm::Surface::GetRecommendedDisplayGrayscaleValue ( ) const
- 27.268.4.24 float gdcmm::Surface::GetRecommendedPresentationOpacity ( ) const
- 27.268.4.25 **VIEWType** gdcmm::Surface::GetRecommendedPresentationType ( ) const
- 27.268.4.26 static **STATES** gdcmm::Surface::GetSTATES ( const char \* *state* ) [static]
- 27.268.4.27 static const char\* gdcmm::Surface::GetSTATESString ( **STATES** *state* ) [static]
- 27.268.4.28 const char\* gdcmm::Surface::GetSurfaceComments ( ) const
- 27.268.4.29 unsigned long gdcmm::Surface::GetSurfaceNumber ( ) const
- 27.268.4.30 bool gdcmm::Surface::GetSurfaceProcessing ( ) const
- 27.268.4.31 const char\* gdcmm::Surface::GetSurfaceProcessingDescription ( ) const
- 27.268.4.32 float gdcmm::Surface::GetSurfaceProcessingRatio ( ) const
- 27.268.4.33 const float\* gdcmm::Surface::GetVectorAccuracy ( ) const
- 27.268.4.34 const **DataElement**& gdcmm::Surface::GetVectorCoordinateData ( ) const
- 27.268.4.35 **DataElement**& gdcmm::Surface::GetVectorCoordinateData ( )
- 27.268.4.36 unsigned short gdcmm::Surface::GetVectorDimensionality ( ) const
- 27.268.4.37 static **VIEWType** gdcmm::Surface::GetVIEWType ( const char \* *type* ) [static]
- 27.268.4.38 static const char\* gdcmm::Surface::GetVIEWTypeString ( **VIEWType** *type* ) [static]
- 27.268.4.39 void gdcmm::Surface::SetAlgorithmFamily ( **SegmentHelper::BasicCodedEntry** const & *BSE* )
- 27.268.4.40 void gdcmm::Surface::SetAlgorithmName ( const char \* *str* )
- 27.268.4.41 void gdcmm::Surface::SetAlgorithmVersion ( const char \* *str* )
- 27.268.4.42 void gdcmm::Surface::SetAxisOfRotation ( const float \* *axis* )

- 27.268.4.43 void gdcm::Surface::SetCenterOfRotation ( const float \* *center* )
- 27.268.4.44 void gdcm::Surface::SetFiniteVolume ( STATES *state* )
- 27.268.4.45 void gdcm::Surface::SetManifold ( STATES *state* )
- 27.268.4.46 void gdcm::Surface::SetMaximumPointDistance ( float *maximum* )
- 27.268.4.47 void gdcm::Surface::SetMeanPointDistance ( float *average* )
- 27.268.4.48 void gdcm::Surface::SetMeshPrimitive ( MeshPrimitive & *mp* )
- 27.268.4.49 void gdcm::Surface::SetNumberOfSurfacePoints ( const unsigned long *nb* )
- 27.268.4.50 void gdcm::Surface::SetNumberOfVectors ( const unsigned long *nb* )
- 27.268.4.51 void gdcm::Surface::SetPointCoordinatesData ( DataElement const & *de* )
- 27.268.4.52 void gdcm::Surface::SetPointPositionAccuracy ( const float \* *accuracies* )
- 27.268.4.53 void gdcm::Surface::SetPointsBoundingBoxCoordinates ( const float \* *coordinates* )
- 27.268.4.54 void gdcm::Surface::SetProcessingAlgorithm ( SegmentHelper::BasicCodedEntry const & *BSE* )
- 27.268.4.55 void gdcm::Surface::SetRecommendedDisplayCIELabValue ( const unsigned short *vl[3]* )
- 27.268.4.56 void gdcm::Surface::SetRecommendedDisplayCIELabValue ( const unsigned short *vl*, const unsigned int *idx* = 0 )
- 27.268.4.57 void gdcm::Surface::SetRecommendedDisplayCIELabValue ( const std::vector< unsigned short > & *vl* )
- 27.268.4.58 void gdcm::Surface::SetRecommendedDisplayGrayscaleValue ( const unsigned short *vl* )
- 27.268.4.59 void gdcm::Surface::SetRecommendedPresentationOpacity ( const float *opacity* )
- 27.268.4.60 void gdcm::Surface::SetRecommendedPresentationType ( VIEWType *type* )
- 27.268.4.61 void gdcm::Surface::SetSurfaceComments ( const char \* *comment* )
- 27.268.4.62 void gdcm::Surface::SetSurfaceNumber ( const unsigned long *nb* )
- 27.268.4.63 void gdcm::Surface::SetSurfaceProcessing ( bool *b* )
- 27.268.4.64 void gdcm::Surface::SetSurfaceProcessingDescription ( const char \* *description* )
- 27.268.4.65 void gdcm::Surface::SetSurfaceProcessingRatio ( const float *ratio* )
- 27.268.4.66 void gdcm::Surface::SetVectorAccuracy ( const float \* *accuracy* )
- 27.268.4.67 void gdcm::Surface::SetVectorCoordinateData ( DataElement const & *de* )

27.268.4.68 void gdcmm::Surface::SetVectorDimensionality ( const unsigned short *dim* )

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

- [gdcmmSurface.h](#)

## 27.269 gdcmm::SurfaceHelper Class Reference

[SurfaceHelper](#) Helper class for [Surface](#) object.

```
#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).*

#### 27.269.1 Detailed Description

[SurfaceHelper](#) Helper class for [Surface](#) object.

#### 27.269.2 Member Typedef Documentation

27.269.2.1 typedef std::vector< unsigned short > gdcmm::SurfaceHelper::ColorArray

#### 27.269.3 Member Function Documentation

27.269.3.1 template<typename T , typename U > std::vector< T > gdcmm::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>	Type of CIELab components.
<i>U</i>	Type of rangeMax value.

**27.269.3.2** `template<typename U > std::vector< float > gdcm::SurfaceHelper::RecommendedDisplayCIELabToRGB ( const ColorArray & CIELab, const U rangeMax = 255 ) [static]`

Convert a DICOM CIE-Lab (after reading) color into RGB.

See also

PS 3.3 C.10.7.1.1

Parameters

<i>CIELab</i>	DICOM CIE-Lab array.
<i>rangeMax</i>	Max value of the RGB range.

Template Parameters

<i>U</i>	Type of rangeMax value.
----------	-------------------------

**27.269.3.3** `template<typename T , typename U > SurfaceHelper::ColorArray gdcm::SurfaceHelper::RGBToRecommendedDisplayCIELab ( const std::vector< T > & RGB, const U rangeMax = 255 ) [static]`

Convert a RGB color into DICOM CIE-Lab (ready to write).

See also

PS 3.3 C.10.7.1.1

Parameters

<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.
----------	-------------------------

27.269.3.4 `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:

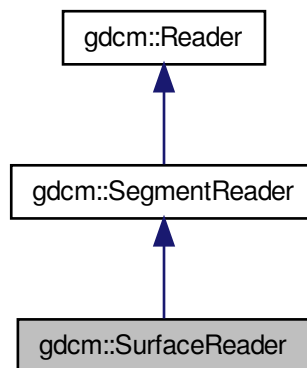
- [gdcmSurfaceHelper.h](#)

## 27.270 gdcm::SurfaceReader Class Reference

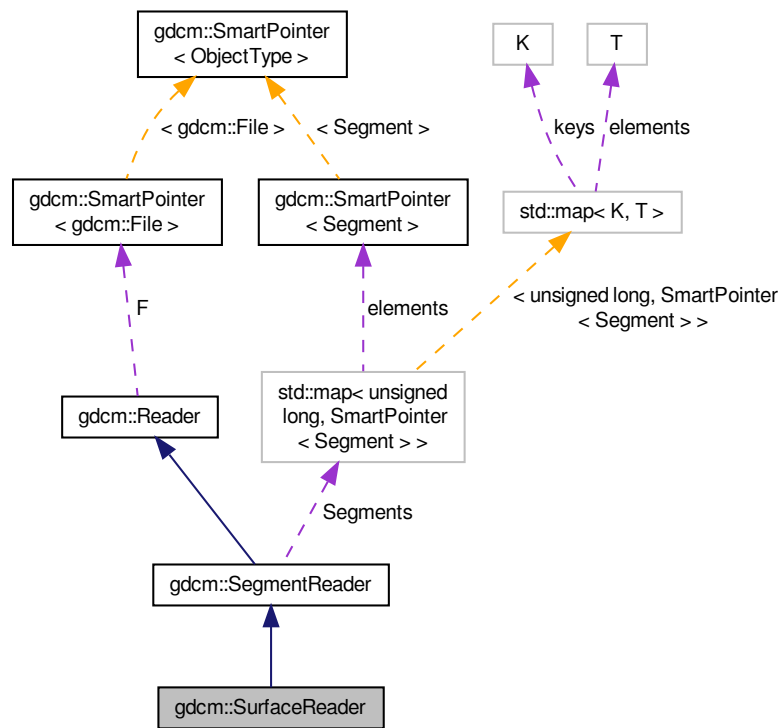
This class defines a SURFACE IE reader. It reads surface mesh module attributes.

```
#include <gdcmSurfaceReader.h>
```

Inheritance diagram for gdcm::SurfaceReader:



Collaboration diagram for gdcmm::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

### 27.270.1 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

## 27.270.2 Constructor & Destructor Documentation

27.270.2.1 `gdcm::SurfaceReader::SurfaceReader ( )`

27.270.2.2 `virtual gdcm::SurfaceReader::~~SurfaceReader ( )` [virtual]

## 27.270.3 Member Function Documentation

27.270.3.1 `unsigned long gdcm::SurfaceReader::GetNumberOfSurfaces ( )` const

27.270.3.2 `virtual bool gdcm::SurfaceReader::Read ( )` [virtual]

Read.

Reimplemented from [gdcm::SegmentReader](#).

27.270.3.3 `bool gdcm::SurfaceReader::ReadPointMacro ( SmartPointer< Surface > surface, const DataSet & surfaceDS )`  
[protected]

27.270.3.4 `bool gdcm::SurfaceReader::ReadSurface ( const Item & surfaceItem, const unsigned long idx )` [protected]

27.270.3.5 `bool gdcm::SurfaceReader::ReadSurfaces ( )` [protected]

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

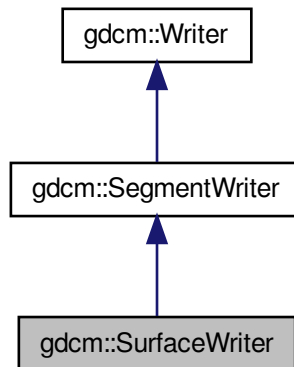
- [gdcmSurfaceReader.h](#)

## 27.271 gdcm::SurfaceWriter Class Reference

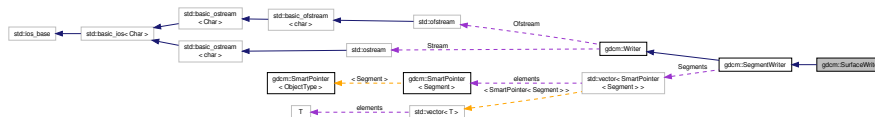
This class defines a SURFACE IE writer. It writes surface mesh module attributes.

```
#include <gdcmSurfaceWriter.h>
```





```
graph LR; A["std::basic_ostream<Char>"] --> B["std::basic_ostream<char>"]; B --> C["std::basic_ios<Char>"]; C --> D["std::ios_base"];
```



- virtual `CSurfaceWriter`

- Write.*

- bool **PrepareWrite** ()

- &ts)

unsigned long NumberOfChannels

## Additional Inherited Members

### 27.271.1 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

### 27.271.2 Constructor & Destructor Documentation

27.271.2.1 `gdcm::SurfaceWriter::SurfaceWriter ( )`

27.271.2.2 `virtual gdcm::SurfaceWriter::~~SurfaceWriter ( ) [virtual]`

### 27.271.3 Member Function Documentation

27.271.3.1 `void gdcm::SurfaceWriter::ComputeNumberOfSurfaces ( ) [protected]`

27.271.3.2 `unsigned long gdcm::SurfaceWriter::GetNumberOfSurfaces ( )`

27.271.3.3 `bool gdcm::SurfaceWriter::PrepareWrite ( ) [protected]`

27.271.3.4 `bool gdcm::SurfaceWriter::PrepareWritePointMacro ( SmartPointer< Surface > surface, DataSet & surfaceDS, const TransferSyntax & ts ) [protected]`

27.271.3.5 `void gdcm::SurfaceWriter::SetNumberOfSurfaces ( const unsigned long nb )`

27.271.3.6 `bool gdcm::SurfaceWriter::Write ( ) [virtual]`

Write.

Reimplemented from [gdcm::SegmentWriter](#).

### 27.271.4 Member Data Documentation

27.271.4.1 `unsigned long gdcm::SurfaceWriter::NumberOfSurfaces [protected]`

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

- [gdcmSurfaceWriter.h](#)

## 27.272 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)

### 27.272.1 Detailed Description

[SwapCode](#) representation.

Examples:

[TestByteSwap.cxx](#).

### 27.272.2 Member Enumeration Documentation

#### 27.272.2.1 enum gdcm::SwapCode::SwapCodeType

Enumerator

***Unknown***

***LittleEndian***

***BigEndian***

***BadLittleEndian***

***BadBigEndian***

### 27.272.3 Constructor & Destructor Documentation

27.272.3.1 `gdcM::SwapCode::SwapCode ( SwapCodeType sc = Unknown )` `[inline]`

### 27.272.4 Member Function Documentation

27.272.4.1 `static int gdcM::SwapCode::GetIndex ( SwapCode const & sc )` `[static]`, `[protected]`

27.272.4.2 `static const char* gdcM::SwapCode::GetSwapCodeString ( SwapCode const & sc )` `[static]`

Referenced by `gdcM::operator<<()`.

27.272.4.3 `gdcM::SwapCode::operator SwapCode::SwapCodeType ( ) const` `[inline]`

### 27.272.5 Friends And Related Function Documentation

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

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

- [gdcMSwapCode.h](#)

## 27.273 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 )`

### 27.273.1 Member Function Documentation

27.273.1.1 `template<typename T> static T gdcM::SwapperDoOp::Swap ( T val )` `[static]`

Referenced by `gdcM::Item::Read()`.

27.273.1.2 `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](#)

## 27.274 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)

### 27.274.1 Detailed Description

Examples:

[ReadExplicitLengthSQIVR.cxx](#).

### 27.274.2 Member Function Documentation

**27.274.2.1** template<typename T > static T gdcm::SwapperNoOp::Swap ( T val ) [inline], [static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Write().

**27.274.2.2** template<typename T > static void gdcm::SwapperNoOp::SwapArray ( T \*, size\_t ) [inline], [static]

Referenced by gdcm::EncodingImplementation< VR::VRBINARY >::Read().

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

- [gdcmSwapper.h](#)

## 27.275 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)
- *strsep*
- 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)

### 27.275.1 Detailed Description

Class to do system operation.

OS independent functionalities

### 27.275.2 Member Function Documentation

**27.275.2.1** static bool [gdcm::System::DeleteDirectory](#) ( const char \* *source* ) [static]

remove a directory named source

27.275.2.2 `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

27.275.2.3 `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](#).

27.275.2.4 `static bool gdcm::System::FileIsDirectory ( const char * name ) [static]`

Check whether the file specified is a directory:

Examples:

[gdcmorthoplanes.cxx](#), and [threadgdcm.cxx](#).

27.275.2.5 `static bool gdcm::System::FileIsSymlink ( const char * name ) [static]`

Check whether name is a symlink.

27.275.2.6 `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](#).

27.275.2.7 `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

27.275.2.8 `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]

**27.275.2.9** `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

**27.275.2.10** `static const char* gdcM::System::GetCurrentModuleFileName ( ) [static]`

Return the directory the current module is located: NOT THREAD SAFE

**27.275.2.11** `static const char* gdcM::System::GetCurrentProcessFileName ( ) [static]`

Return the directory the current process (executable) is located: NOT THREAD SAFE

**27.275.2.12** `static const char* gdcM::System::GetCurrentResourcesDirectory ( ) [static]`

On some system (Apple) return the path to the current bundled 'Resources' directory NOT THREAD SAFE

**27.275.2.13** `static const char* gdcM::System::GetCurrentWorkingDirectory ( ) [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

**27.275.2.14** `static bool gdcM::System::GetCurrentHostName ( char hostname[255] ) [static]`

Retrieve the hostname, only the first 255 byte are copied. This may come handy to specify the Station Name

**27.275.2.15** `static const char* gdcM::System::GetCurrentLastError ( ) [static]`

Return the last error.

**27.275.2.16** `static const char* gdcM::System::GetCurrentLocaleCharSet ( ) [static]`

return locale charmap

**27.275.2.17** `static bool gdcM::System::GetCurrentPermissions ( const char * file, unsigned short & mode ) [static],  
[protected]`

NOT THREAD SAFE.

**27.275.2.18** `static const char* gdcM::System::GetCurrentTimezoneOffsetFromUTC ( ) [static]`

Return the value for Timezone Offset From UTC as string.

Warning

not thread safe



27.275.2.19 static bool gdcmm::System::MakeDirectory ( const char \* *path* ) [static]

Create a directory name path.

27.275.2.20 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)

27.275.2.21 static bool gdcmm::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](#)

27.275.2.22 static bool gdcmm::System::RemoveFile ( const char \* *source* ) [static]

remove a file named source

27.275.2.23 static bool gdcmm::System::SetPermissions ( const char \* *file*, unsigned short *mode* ) [static],  
[protected]

27.275.2.24 static int gdcmm::System::StrCaseCmp ( const char \* *s1*, const char \* *s2* ) [static]

consistent func for C99 spec of strcasecmp/strncasecmp

27.275.2.25 static int gdcmm::System::StrNCaseCmp ( const char \* *s1*, const char \* *s2*, size\_t *n* ) [static]

Precondition

$n \neq 0$

27.275.2.26 static char\* gdcmm::System::StrSep ( char \*\* *stringp*, const char \* *delim* ) [static]

strsep

27.275.2.27 static char\* gdcmm::System::StrTokR ( char \* *ptr*, const char \* *sep*, char \*\* *end* ) [static]

strtok\_r

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

- [gdcmmSystem.h](#)

## 27.276 gdcmm::Table Class Reference

[Table.](#)

```
#include <gdcmmTable.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)

### 27.276.1 Detailed Description

[Table.](#)

### 27.276.2 Member Typedef Documentation

27.276.2.1 typedef std::map<[Tag](#), [TableEntry](#)> [gdcmm::Table::MapTableEntry](#)

### 27.276.3 Constructor & Destructor Documentation

27.276.3.1 [gdcmm::Table::Table](#) ( ) [\[inline\]](#)

27.276.3.2 [gdcmm::Table::~~Table](#) ( ) [\[inline\]](#)

### 27.276.4 Member Function Documentation

27.276.4.1 const [TableEntry](#)& [gdcmm::Table::GetTableEntry](#) ( const [Tag](#) & *tag* ) const [\[inline\]](#)

27.276.4.2 void [gdcmm::Table::InsertEntry](#) ( [Tag](#) const & *tag*, [TableEntry](#) const & *te* ) [\[inline\]](#)

### 27.276.5 Friends And Related Function Documentation

27.276.5.1 std::ostream& [operator<<](#) ( std::ostream &\_os, const [Table](#) &\_val ) [\[friend\]](#)

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

- [gdcmmTable.h](#)

## 27.277 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](#) ()

### 27.277.1 Detailed Description

[TableEntry](#).

### 27.277.2 Constructor & Destructor Documentation

**27.277.2.1** `gdcm::TableEntry::TableEntry ( const char * attribute = 0, Type const & type = Type () , const char * des = 0 )`  
[inline]

**27.277.2.2** `gdcm::TableEntry::~~TableEntry ( )` [inline]

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

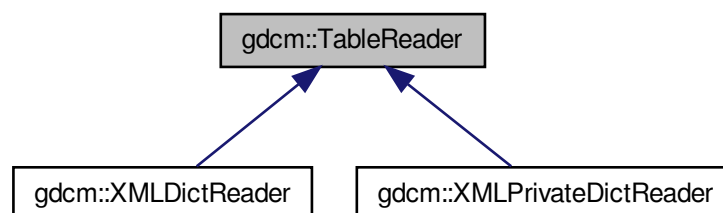
- [gdcmTableEntry.h](#)

## 27.278 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)

### 27.278.1 Detailed Description

Class for representing a [TableReader](#).

#### Note

This class is an empty shell meant to be derived

### 27.278.2 Constructor & Destructor Documentation

27.278.2.1 `gdcm::TableReader::TableReader ( Defs & defs )` `[inline]`

27.278.2.2 `virtual gdcm::TableReader::~~TableReader ( )` `[inline]`, `[virtual]`

### 27.278.3 Member Function Documentation

27.278.3.1 `virtual void gdcm::TableReader::CharacterDataHandler ( const char * data, int length )` `[virtual]`

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

27.278.3.2 `virtual void gdcm::TableReader::EndElement ( const char * name )` `[virtual]`

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

- 27.278.3.3 `const Defs& gdcm::TableReader::GetDefs ( ) const` `[inline]`
- 27.278.3.4 `const char* gdcm::TableReader::GetFilename ( )` `[inline]`
- 27.278.3.5 `void gdcm::TableReader::HandleIOD ( const char ** atts )`
- 27.278.3.6 `void gdcm::TableReader::HandleIOEntry ( const char ** atts )`
- 27.278.3.7 `void gdcm::TableReader::HandleMacro ( const char ** atts )`
- 27.278.3.8 `void gdcm::TableReader::HandleMacroEntry ( const char ** atts )`
- 27.278.3.9 `void gdcm::TableReader::HandleMacroEntryDescription ( const char ** atts )`
- 27.278.3.10 `void gdcm::TableReader::HandleModule ( const char ** atts )`
- 27.278.3.11 `void gdcm::TableReader::HandleModuleEntry ( const char ** atts )`
- 27.278.3.12 `void gdcm::TableReader::HandleModuleEntryDescription ( const char ** atts )`
- 27.278.3.13 `void gdcm::TableReader::HandleModuleInclude ( const char ** atts )`
- 27.278.3.14 `int gdcm::TableReader::Read ( )`
- 27.278.3.15 `void gdcm::TableReader::SetFilename ( const char * filename )` `[inline]`
- 27.278.3.16 `virtual void gdcm::TableReader::StartElement ( const char * name, const char ** atts )` `[virtual]`

Reimplemented in [gdcm::XMLDictReader](#), and [gdcm::XMLPrivateDictReader](#).

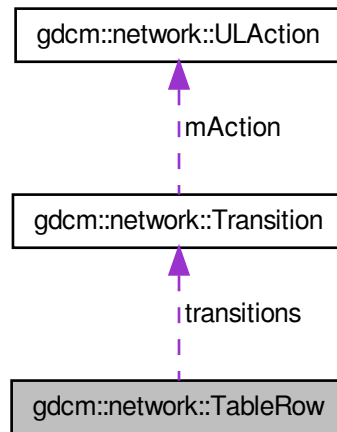
The documentation for this class was generated from the following file:

- [gdcmTableReader.h](#)

## 27.279 gdcm::network::TableRow Class Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for `gdcM::network::TableRow`:



## Public Member Functions

- [TableRow](#) ()
- [~TableRow](#) ()

## Public Attributes

- [Transition](#) \* [transitions](#) [[cMaxStateID](#)]

## 27.279.1 Constructor & Destructor Documentation

27.279.1.1 `gdcM::network::TableRow::TableRow ( )` [[inline](#)]

References `gdcM::network::cMaxStateID`, and `transitions`.

27.279.1.2 `gdcM::network::TableRow::~~TableRow ( )` [[inline](#)]

References `gdcM::network::cMaxStateID`, and `transitions`.

## 27.279.2 Member Data Documentation

27.279.2.1 `Transition* gdcM::network::TableRow::transitions[cMaxStateID]`

Referenced by `TableRow()`, and `~TableRow()`.

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

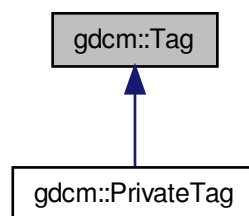
- [gdcmULTransitionTable.h](#)

## 27.280 gdcm::Tag Class Reference

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)

```
#include <gdcmTag.h>
```

Inheritance diagram for gdcm::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)

### 27.280.1 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](#), [ClinicalTrialIdentificationWorkflow.cs](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateJPIPDataSet.cxx](#), [DumpToSQLITE3.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [ExtractEncryptedContent.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FileChangeTS.cs](#), [FileStreaming.cs](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [gdcmmrtionplan.cxx](#), [gdcmmrtplan.cxx](#), [GenAllIVR.cxx](#), [GenFakeIdentifyFile.cxx](#), [GenFakeImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [GetJPEGSamplePrecision.cxx](#), [GetSequenceUltrasound.cxx](#), [GetSubSequenceData.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [PublicDict.cxx](#), [ReadAndDumpDICOMDIR.cxx](#), [ReadAndPrintAttributes.cxx](#), [ReadExplicitLengthSQIVR.cxx](#), [ReformatFile.cs](#), [rle2img.cxx](#), [SimplePrintPatientName.cs](#), [SimpleScanner.cxx](#), [SortImage.cxx](#), [StandardizeFiles.cs](#), [StreamImageReaderTest.cxx](#), [TraverseModules.cxx](#), and [VolumeSorter.cxx](#).

## 27.280.2 Constructor &amp; Destructor Documentation

27.280.2.1 `gdcmm::Tag ( uint16_t group, uint16_t element ) [inline]`

Constructor with 2\*uint16\_t.

27.280.2.2 `gdcmm::Tag ( uint32_t tag = 0 ) [inline]`

Constructor with 1\*uint32\_t Prefer the ctor that takes two uint16\_t.

27.280.2.3 `gdcmm::Tag ( const Tag &_val ) [inline]`

References tag.

## 27.280.3 Member Function Documentation

27.280.3.1 `uint16_t gdcmm::Tag::GetElement ( ) const [inline]`

Returns the 'Element number' of the given [Tag](#).

## Examples:

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by `gdcmm::DataSet::ComputeGroupLength()`, `IsGroupXX()`, `gdcmm::PrivateDict::PrintXML()`, `gdcmm::PrivateTag::PrivateTag()`, `gdcmm::SequenceOfFragments::ReadValue()`, and `SetPrivateCreator()`.

27.280.3.2 `uint32_t gdcmm::Tag::GetElementTag ( ) const [inline]`

Returns the full tag value of the given [Tag](#).

**27.280.3.3** `uint16_t gdcmm::Tag::GetGroup ( ) const [inline]`

Returns the 'Group number' of the given [Tag](#).

Examples:

[DuplicatePCDE.cxx](#), and [GenAllVR.cxx](#).

Referenced by `gdcmm::DataSet::ComputeGroupLength()`, `gdcmm::CommandDataSet::Insert()`, `gdcmm::FileMetaInformation::Insert()`, `gdcmm::DataSet::Insert()`, `IsGroupXX()`, `gdcmm::PrivateDict::PrintXML()`, `gdcmm::SequenceOfFragments::ReadValue()`, `gdcmm::Attribute< Group, Element, TVR, TVM >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1 >::SetFromDataElement()`, `gdcmm::Attribute< Group, Element, TVR, VM::VM1_n >::SetFromDataElement()`, and `SetPrivateCreator()`.

**27.280.3.4** `uint32_t gdcmm::Tag::GetLength ( ) const [inline]`

return the length of tag (read: size on disk)

**27.280.3.5** `Tag gdcmm::Tag::GetPrivateCreator ( ) const [inline]`

Return the Private Creator Data [Element](#) tag of a private data element.

References `SetElement()`.

**27.280.3.6** `bool gdcmm::Tag::IsGroupLength ( ) const [inline]`

return whether the tag correspond to a group length tag:

**27.280.3.7** `bool gdcmm::Tag::IsGroupXX ( const Tag & t ) const [inline]`

e.g 6002,3000 belong to groupXX: 6000,3000

References `GetElement()`, `GetGroup()`, and `IsPrivate()`.

**27.280.3.8** `bool gdcmm::Tag::IsIllegal ( ) const [inline]`

return if the tag is considered to be an illegal tag

**27.280.3.9** `bool gdcmm::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()`.

27.280.3.10 `bool gdcmm::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](#).

27.280.3.11 `bool gdcmm::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...

27.280.3.12 `bool gdcmm::Tag::operator!= ( const Tag & _val ) const [inline]`

References tag.

27.280.3.13 `bool gdcmm::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.

27.280.3.14 `bool gdcmm::Tag::operator<= ( const Tag & t2 ) const [inline]`

27.280.3.15 `Tag& gdcmm::Tag::operator= ( const Tag & _val ) [inline]`

References tag.

27.280.3.16 `bool gdcmm::Tag::operator== ( const Tag & _val ) const [inline]`

References tag.

27.280.3.17 `const uint16_t& gdcmm::Tag::operator[] ( const unsigned int & _id ) const [inline]`

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

27.280.3.18 `uint16_t& gdcmm::Tag::operator[] ( const unsigned int & _id ) [inline]`

Returns the Group or [Element](#) of the given [Tag](#), depending on id (0/1)

27.280.3.19 `std::string gdcmm::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)

27.280.3.20 `std::string gdcmm::Tag::PrintAsContinuousUpperCaseString ( ) const`

Same as `PrintAsContinuousString`, but hexadecimal [a-f] are printed using upper case.

27.280.3.21 `std::string gdcmm::Tag::PrintAsPipeSeparatedString ( ) const`

Print as a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromPipeSeparatedString](#)

27.280.3.22 `template<typename TSwap> std::istream& gdcmm::Tag::Read ( std::istream & is ) [inline]`

Read a tag from binary representation.

27.280.3.23 `bool gdcmm::Tag::ReadFromCommaSeparatedString ( const char * str )`

Read from a comma separated string. This is a highly user oriented function, the string should be formatted as↵: 1234,5678 to specify the tag (0x1234,0x5678) The notation comes from the DICOM standard, and is handy to use from a command line program

27.280.3.24 `bool gdcmm::Tag::ReadFromContinuousString ( const char * str )`

Read From XML formatted tag value eg. tag = "12345678" It comes in useful when reading tag values from XML file(in NativeDICOMModel)

27.280.3.25 `bool gdcmm::Tag::ReadFromPipeSeparatedString ( const char * str )`

Read from a pipe separated string (GDCM 1.x compat only). Do not use in newer code

See also

[ReadFromCommaSeparatedString](#)

27.280.3.26 `void gdcmm::Tag::SetElement ( uint16_t element ) [inline]`

Sets the '[Element](#) number' of the given [Tag](#).

Examples:

[DuplicatePCDE.cxx](#), and [PublicDict.cxx](#).

Referenced by `GetPrivateCreator()`, and `gdcmm::operator>>()`.

27.280.3.27 `void gdcmm::Tag::SetElementTag ( uint16_t group, uint16_t element ) [inline]`

Sets the 'Group number' & '[Element](#) number' of the given [Tag](#).

27.280.3.28 void gdcmm::Tag::SetElementTag ( uint32\_t tag ) [inline]

Sets the full tag value of the given [Tag](#).

27.280.3.29 void gdcmm::Tag::SetGroup ( uint16\_t group ) [inline]

Sets the 'Group number' of the given [Tag](#).

Referenced by gdcmm::operator>>().

27.280.3.30 void gdcmm::Tag::SetPrivateCreator ( Tag const & t ) [inline]

Set private creator:

Examples:

[DuplicatePCDE.cxx](#).

References GetElement(), GetGroup(), and IsPrivate().

27.280.3.31 template<typename TSwap > const std::ostream& gdcmm::Tag::Write ( std::ostream & os ) const [inline]

Write a tag in binary rep.

Referenced by gdcmm::SequenceOfItems::Write(), gdcmm::Item::Write(), and gdcmm::SequenceOfFragments::Write().

## 27.280.4 Friends And Related Function Documentation

27.280.4.1 std::ostream& operator<< ( std::ostream & \_os, const Tag & \_val ) [friend]

27.280.4.2 std::istream& operator>> ( std::istream & \_is, Tag & \_val ) [friend]

## 27.280.5 Member Data Documentation

27.280.5.1 char gdcmm::Tag::bytes[4]

27.280.5.2 uint32\_t gdcmm::Tag::tag

Referenced by operator!==( ), operator<(), operator=(), operator==( ), and Tag().

27.280.5.3 uint16\_t gdcmm::Tag::tags[2]

Referenced by operator<().

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

- [gdcmmTag.h](#)

## 27.281 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.*

#### 27.281.1 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)

#### 27.281.2 Constructor & Destructor Documentation

27.281.2.1 `gdcm::TagPath::TagPath ( )`

27.281.2.2 `gdcm::TagPath::~~TagPath ( )`

#### 27.281.3 Member Function Documentation

27.281.3.1 `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

27.281.3.2 `bool gdcm::TagPath::ConstructFromTagList ( Tag const * l, unsigned int n )`

Construct from a list of tags.

27.281.3.3 `static bool gdcm::TagPath::IsValid ( const char * path )` `[static]`

Return if path is valid or not.

27.281.3.4 void gdcm::TagPath::Print ( std::ostream & ) const

27.281.3.5 bool gdcm::TagPath::Push ( Tag const & t )

27.281.3.6 bool gdcm::TagPath::Push ( unsigned int *itemnum* )

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

- [gdcmTagPath.h](#)

## 27.282 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, unsigned long 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 [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.*

### 27.282.1 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

### 27.282.2 Member Typedef Documentation

#### 27.282.2.1 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

#### 27.282.2.2 typedef const char\* const(\* gdcm::Testing::MediaStorageDataFilesType)[2]

return the table that map the media storage (as string) of a filename (gdcmData)

### 27.282.3 Constructor & Destructor Documentation

#### 27.282.3.1 gdcm::Testing::Testing ( ) [inline]

#### 27.282.3.2 gdcm::Testing::~~Testing ( ) [inline]

### 27.282.4 Member Function Documentation

#### 27.282.4.1 static bool gdcm::Testing::ComputeFileMD5 ( const char \* filename, char digest\_str[33] ) [static]



27.282.4.2 `static bool gdcm::Testing::ComputeMD5 ( const char * buffer, unsigned long 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.

27.282.4.3 `static const char* gdcm::Testing::GetDataExtraRoot ( )` `[static]`

Return the GDCM DATA EXTRA ROOT.

Examples:

[DiscriminateVolume.cxx](#), [reslicesphere.cxx](#), and [VolumeSorter.cxx](#).

27.282.4.4 `static const char* gdcm::Testing::GetDataRoot ( )` `[static]`

Return the GDCM DATA ROOT.

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), and [MagnifyFile.cxx](#).

27.282.4.5 `static const char* gdcm::Testing::GetFileName ( unsigned int file )` `[static]`

27.282.4.6 `static const char* const* gdcm::Testing::GetFileNames ( )` `[static]`

return the table of fullpath to gdcmData DICOM files:

Examples:

[TestReader.cxx](#).

27.282.4.7 `static int gdcm::Testing::GetLossyFlagFromFile ( const char * filepath )` `[static]`

Return the lossy flag of the given filename -1 -> Error 0 -> Lossless 1 -> Lossy

27.282.4.8 `static const char* const* gdcm::Testing::GetMD5DataImage ( unsigned int file )` `[static]`

27.282.4.9 `static MD5DataImagesType gdcm::Testing::GetMD5DataImages ( )` `[static]`

27.282.4.10 `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.

27.282.4.11 `static const char* gdcm::Testing::GetMD5FromFile ( const char * filepath ) [static]`

27.282.4.12 `static const char* const* gdcm::Testing::GetMediaStorageDataFile ( unsigned int file ) [static]`

27.282.4.13 `static MediaStorageDataFileType gdcm::Testing::GetMediaStorageDataFiles ( ) [static]`

27.282.4.14 `static const char* gdcm::Testing::GetMediaStorageFromFile ( const char * filepath ) [static]`

Examples:

[TestReader.cxx](#).

27.282.4.15 `static unsigned int gdcm::Testing::GetNumberOfFileNames ( ) [static]`

27.282.4.16 `static unsigned int gdcm::Testing::GetNumberOfMD5DataImages ( ) [static]`

27.282.4.17 `static unsigned int gdcm::Testing::GetNumberOfMediaStorageDataFiles ( ) [static]`

27.282.4.18 `static const char* gdcm::Testing::GetPixelSpacingDataRoot ( ) [static]`

Return the GDCM PIXEL SPACING DATA ROOT (See David Clunie website for dataset)

27.282.4.19 `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

27.282.4.20 `static const char* gdcm::Testing::GetSourceDirectory ( ) [static]`

27.282.4.21 `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

27.282.4.22 `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:

27.282.4.23 `static const wchar_t* gdcm::Testing::GetTempDirectoryW ( const wchar_t * subdir = 0 ) [static]`

NOT THREAD SAFE.

27.282.4.24 `static const char* gdcm::Testing::GetTempFilename ( const char * filename, const char * subdir = 0 ) [static]`

NOT THREAD SAFE.

27.282.4.25 `static const wchar_t* gdcm::Testing::GetTempFilenameW ( const wchar_t * filename, const wchar_t * subdir = 0 ) [static]`

NOT THREAD SAFE.

27.282.4.26 void gdcM::Testing::Print ( std::ostream & os = std::cout )

Print.

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

- [gdcMTesting.h](#)

## 27.283 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](#) ()

### 27.283.1 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.

### 27.283.2 Constructor & Destructor Documentation

27.283.2.1 `gdcm::Trace::Trace ( )`

27.283.2.2 `gdcm::Trace::~~Trace ( )`

### 27.283.3 Member Function Documentation

27.283.3.1 `static void gdcm::Trace::DebugOff ( )` `[static]`

Examples:

[TestReader.cxx](#).

27.283.3.2 `static void gdcm::Trace::DebugOn ( )` `[static]`

Examples:

[Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), and [StreamImageReaderTest.cxx](#).

27.283.3.3 `static void gdcm::Trace::ErrorOff ( )` `[static]`

27.283.3.4 `static void gdcm::Trace::ErrorOn ( )` `[static]`

27.283.3.5 `static bool gdcm::Trace::GetDebugFlag ( )` `[static]`

27.283.3.6 `static std::ostream& gdcm::Trace::GetDebugStream ( )` `[static]`

27.283.3.7 `static bool gdcm::Trace::GetErrorFlag ( )` `[static]`

27.283.3.8 `static std::ostream& gdcm::Trace::GetErrorStream ( )` `[static]`

27.283.3.9 `static std::ostream& gdcm::Trace::GetStream ( )` `[static]`

27.283.3.10 `static bool gdcmm::Trace::GetWarningFlag ( ) [static]`

27.283.3.11 `static std::ostream& gdcmm::Trace::GetWarningStream ( ) [static]`

27.283.3.12 `static void gdcmm::Trace::SetDebug ( bool debug ) [static]`

Turn debug messages on (default: false)

Examples:

[DumpToSQLITE3.cxx](#).

27.283.3.13 `static void gdcmm::Trace::SetDebugStream ( std::ostream & os ) [static]`

Explicitely set the stream which receive Debug messages:

27.283.3.14 `static void gdcmm::Trace::SetError ( bool debug ) [static]`

Turn error messages on (default: true)

27.283.3.15 `static void gdcmm::Trace::SetErrorStream ( std::ostream & os ) [static]`

Explicitely set the stream which receive Error messages:

Examples:

[CStoreQtProgress.cxx](#).

27.283.3.16 `static void gdcmm::Trace::SetStream ( std::ostream & os ) [static]`

Explicitely set the ostream for [gdcmm::Trace](#) to report to This will set the DebugStream, WarningStream and ErrorStream at once:

27.283.3.17 `static void gdcmm::Trace::SetStreamToFile ( const char * filename ) [static]`

Explicitely set the filename for [gdcmm::Trace](#) to report to The file will be created (it will not append to existing file)

27.283.3.18 `static void gdcmm::Trace::SetWarning ( bool debug ) [static]`

Turn warning messages on (default: true)

Examples:

[DumpToSQLITE3.cxx](#).

27.283.3.19 `static void gdcmm::Trace::SetWarningStream ( std::ostream & os ) [static]`

Explicitely set the stream which receive Warning messages:

27.283.3.20 `static void gdcm::Trace::WarningOff ( ) [static]`

Examples:

[TestReader.cxx](#).

27.283.3.21 `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](#)

## 27.284 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](#),  
[CT\\_private\\_ELE](#),  
[JPIPReferenced](#),

[TS\\_END](#) }

## Public Member Functions

- [TransferSyntax](#) (TType type=[ImplicitVRLittleEndian](#))
- bool [CanStoreLossy](#) () const
- [NegociatedType](#) [GetNegociatedType](#) () const
- const char \* [GetString](#) () const
- [SwapCode](#) [GetSwapCode](#) () const
- bool [IsEncapsulated](#) () const
- bool [IsEncoded](#) () const
- bool [IsExplicit](#) () const
- bool [IsImplicit](#) () const
- bool [IsLossless](#) () const
- bool [IsLossy](#) () const
- bool [IsValid](#) () const
- [operator](#) TType () const

## Static Public Member Functions

- static const char \* [GetTSString](#) (TType ts)
- static TType [GetTSType](#) (const char \*str)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [TransferSyntax](#) &ts)

## 27.284.1 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 [gdcm::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:

[FileChangeTS.cs](#), [GetJPEGSamplePrecision.cxx](#), and [LargeVRDSExplicit.cxx](#).

## 27.284.2 Member Enumeration Documentation

### 27.284.2.1 enum gdcm::TransferSyntax::NegociatedType

Enumerator

*Unknown*

*Explicit*

*Implicit*

### 27.284.2.2 enum gdcm::TransferSyntax::TSType

Enumerator

*ImplicitVRLittleEndian*

*ImplicitVRBigEndianPrivateGE*

*ExplicitVRLittleEndian*

*DeflatedExplicitVRLittleEndian*

*ExplicitVRBigEndian*

*JPEGBaselineProcess1*

*JPEGExtendedProcess2\_4*

*JPEGExtendedProcess3\_5*

*JPEGSpectralSelectionProcess6\_8*

*JPEGFullProgressionProcess10\_12*

*JPEGLosslessProcess14*

*JPEGLosslessProcess14\_1*

*JPEGLSLossless*

*JPEGLSNearLossless*

*JPEG2000Lossless*

*JPEG2000*

*JPEG2000Part2Lossless*

*JPEG2000Part2*

*RLELossless*

*MPEG2MainProfile*

*ImplicitVRBigEndianACRNEMA*

*CT\_private\_ELE*

*JPIPReferenced*

*TS\_END*

## 27.284.3 Constructor & Destructor Documentation

### 27.284.3.1 gdcm::TransferSyntax::TransferSyntax ( TSType type = ImplicitVRLittleEndian ) [inline]

## 27.284.4 Member Function Documentation

### 27.284.4.1 bool gdcm::TransferSyntax::CanStoreLossy ( ) const

return if TransFer Syntax Allow storing of Lossy Pixel Data



27.284.4.2 **NegotiatedType** gdcm::TransferSyntax::GetNegociatedType ( ) const

27.284.4.3 const char\* gdcm::TransferSyntax::GetString ( ) const [inline]

References GetTSString().

27.284.4.4 **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.

27.284.4.5 static const char\* gdcm::TransferSyntax::GetTSString ( TSType ts ) [static]

Examples:

[LargeVRDSExplicit.cxx](#).

Referenced by GetString(), and gdcm::operator<<().

27.284.4.6 static TSType gdcm::TransferSyntax::GetTSType ( const char \* str ) [static]

27.284.4.7 bool gdcm::TransferSyntax::IsEncapsulated ( ) const

Examples:

[ExtractIconFromFile.cxx](#).

27.284.4.8 bool gdcm::TransferSyntax::IsEncoded ( ) const

27.284.4.9 bool gdcm::TransferSyntax::IsExplicit ( ) const

27.284.4.10 bool gdcm::TransferSyntax::IsImplicit ( ) const

27.284.4.11 bool gdcm::TransferSyntax::IsLossless ( ) const

Return if the transfer syntax algorithm is a lossless algorithm

27.284.4.12 bool gdcm::TransferSyntax::IsLossy ( ) const

Return if the transfer syntax algorithm is a lossy algorithm

27.284.4.13 bool gdcm::TransferSyntax::IsValid ( ) const [inline]

27.284.4.14 gdcm::TransferSyntax::operator TSType ( ) const [inline]

## 27.284.5 Friends And Related Function Documentation

27.284.5.1 `std::ostream& operator<< ( std::ostream & os, const TransferSyntax & ts )` [friend]

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

- [gdcmTransferSyntax.h](#)

## 27.285 gdcm::network::TransferSyntaxSub Class Reference

[TransferSyntaxSub](#) Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

```
#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

### 27.285.1 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

### 27.285.2 Constructor & Destructor Documentation

27.285.2.1 `gdcm::network::TransferSyntaxSub::TransferSyntaxSub ( )`

### 27.285.3 Member Function Documentation

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

27.285.3.2 `bool gdcm::network::TransferSyntaxSub::operator== ( const TransferSyntaxSub & ts ) const` [inline]

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

27.285.3.4 `std::istream& gdcm::network::TransferSyntaxSub::Read ( std::istream & is )`

27.285.3.5 `void gdcm::network::TransferSyntaxSub::SetName ( const char * name )`

27.285.3.6 `void gdcm::network::TransferSyntaxSub::SetNameFromUID ( UIDs::TSName tsname )`

27.285.3.7 `size_t gdcm::network::TransferSyntaxSub::Size ( ) const`

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

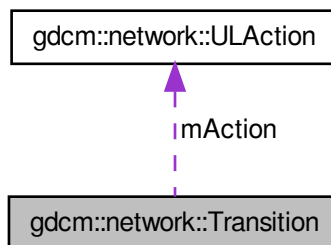
The documentation for this class was generated from the following file:

- [gdcmTransferSyntaxSub.h](#)

## 27.286 gdcm::network::Transition Struct Reference

```
#include <gdcmULTransitionTable.h>
```

Collaboration diagram for gdcm::network::Transition:



### Public Member Functions

- [Transition](#) ( )
- [Transition](#) (int inEndState, [ULAction](#) \*inAction)
- [~Transition](#) ( )

### Static Public Member Functions

- static [Transition](#) \* [MakeNew](#) (int inEndState, [ULAction](#) \*inAction)

### Public Attributes

- [ULAction](#) \* [mAction](#)
- int [mEnd](#)

### 27.286.1 Constructor & Destructor Documentation

**27.286.1.1** `gdcm::network::Transition::Transition ( ) [inline]`

References `gdcm::network::eStaDoesNotExist`, `mAction`, and `mEnd`.

Referenced by `MakeNew()`.

**27.286.1.2** `gdcm::network::Transition::~~Transition ( ) [inline]`

References `mAction`.

**27.286.1.3** `gdcm::network::Transition::Transition ( int inEndState, ULAction * inAction ) [inline]`

References `mAction`, and `mEnd`.

## 27.286.2 Member Function Documentation

**27.286.2.1** `static Transition* gdcm::network::Transition::MakeNew ( int inEndState, ULAction * inAction ) [inline], [static]`

References `Transition()`.

## 27.286.3 Member Data Documentation

**27.286.3.1** `ULAction* gdcm::network::Transition::mAction`

Referenced by `Transition()`, and `~Transition()`.

**27.286.3.2** `int gdcm::network::Transition::mEnd`

Referenced by `Transition()`.

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

- [gdcmULTransitionTable.h](#)

## 27.287 gdcm::Type Class Reference

Type.

```
#include <gdcmType.h>
```

### Public Types

- enum [TypeType](#) {  
`T1 = 0,`  
`T1C,`  
`T2,`  
`T2C,`  
`T3,`

UNKNOWN }

## Public Member Functions

- [Type](#) ([TypeType](#) type=[UNKNOWN](#))
- [operator TypeType](#) () const

## Static Public Member Functions

- static const char \* [GetTypeString](#) ([TypeType](#) type)
- static [TypeType](#) [GetTypeType](#) (const char \*type)

## Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Type](#) &vr)

## 27.287.1 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](#).

## 27.287.2 Member Enumeration Documentation

### 27.287.2.1 enum gdcm::Type::TypeType

#### Enumerator

**T1**

**T1C**

**T2**

**T2C**

**T3**

**UNKNOWN**

### 27.287.3 Constructor & Destructor Documentation

27.287.3.1 `gdcmm::Type::Type ( TypeType type = UNKNOWN )` `[inline]`

### 27.287.4 Member Function Documentation

27.287.4.1 `static const char* gdcmm::Type::GetTypeString ( TypeType type )` `[static]`

Referenced by `gdcmm::operator<<()`.

27.287.4.2 `static TypeType gdcmm::Type::GetTypeType ( const char * type )` `[static]`

Referenced by `gdcmm::ModuleEntry::ModuleEntry()`.

27.287.4.3 `gdcmm::Type::operator TypeType ( ) const` `[inline]`

### 27.287.5 Friends And Related Function Documentation

27.287.5.1 `std::ostream& operator<< ( std::ostream & os, const Type & vr )` `[friend]`

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

- [gdcmmType.h](#)

## 27.288 gdcmm::UI Struct Reference

```
#include <gdcmmVR.h>
```

### Public Attributes

- char [Internal](#) [64+1]

### Friends

- std::ostream & [operator<<](#) (std::ostream &\_os, const [UI](#) &\_val)

### 27.288.1 Friends And Related Function Documentation

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

### 27.288.2 Member Data Documentation

27.288.2.1 `char gdcmm::UI::Internal[64+1]`

Referenced by `gdcmm::operator<<()`.

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

- [gdcmmVR.h](#)

## 27.289 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)

### 27.289.1 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](#), and [uid\\_unique.cxx](#).

### 27.289.2 Constructor & Destructor Documentation

#### 27.289.2.1 gdcm::UIDGenerator::UIDGenerator ( ) [inline]

By default the root of a UID is a GDCM Root...

### 27.289.3 Member Function Documentation

#### 27.289.3.1 `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](#), [StreamImageReaderTest.cxx](#), and [uid\\_unique.cxx](#).

#### 27.289.3.2 `static bool gdcm::UIDGenerator::GenerateUUID ( unsigned char * uid_data ) [static],[protected]`

#### 27.289.3.3 `static const char* gdcm::UIDGenerator::GetGDCMUID ( ) [static]`

Return the default (GDCM) root UID:

#### 27.289.3.4 `static const char* gdcm::UIDGenerator::GetRoot ( ) [static]`

#### 27.289.3.5 `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`)

#### 27.289.3.6 `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 responsability for guaranteeing uniqueness of [UIDs](#) when the root is longer than 26 bytes.

Examples:

[uid\\_unique.cxx](#).

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

- [gdcmUIDGenerator.h](#)

## 27.290 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](#)↵



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_5_1_4_1_1_13_1_3 }
```

## 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)

### 27.290.1 Detailed Description

all known uids

Examples:

[GenerateStandardSOPClasses.cxx](#).

### 27.290.2 Member Typedef Documentation

27.290.2.1 `typedef const char* const(* gdcm::UIDs::TransferSyntaxStringsType)[2]`

### 27.290.3 Member Enumeration Documentation

27.290.3.1 `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*  
*JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImage*

*JPEGLSLosslessImageCompression*  
*JPEGLSLossyNearLosslessImageCompression*  
*JPEG2000ImageCompressionLosslessOnly*  
*JPEG2000ImageCompression*  
*JPEG2000Part2MulticomponentImageCompressionLosslessOnly*  
*JPEG2000Part2MulticomponentImageCompression*  
*JPIPReferenced*  
*JPIPReferencedDeflate*  
*MPEG2MainProfileMainLevel*  
*RLELossless*  
*RFC2557MIMEencapsulation*  
*XMLEncoding*  
*MediaStorageDirectoryStorage*  
*TalairachBrainAtlasFrameofReference*  
*SPM2T1FrameofReference*  
*SPM2T2FrameofReference*  
*SPM2PDFFrameofReference*  
*SPM2EPIFrameofReference*  
*SPM2FIL T1FrameofReference*  
*SPM2PETFrameofReference*  
*SPM2TRANSMFrameofReference*  
*SPM2SPECTFrameofReference*  
*SPM2GRAYFrameofReference*  
*SPM2WHITEFrameofReference*  
*SPM2CSFFFrameofReference*  
*SPM2BRAINMASKFrameofReference*  
*SPM2AVG305T1FrameofReference*  
*SPM2AVG152T1FrameofReference*  
*SPM2AVG152T2FrameofReference*  
*SPM2AVG152PDFFrameofReference*

*SPM2SINGLESUBJT1FrameofReference*  
*ICBM452T1FrameofReference*  
*ICBMSingleSubjectMRIFrameofReference*  
*BasicStudyContentNotificationSOPClassRetired*  
*StorageCommitmentPushModelSOPClass*  
*StorageCommitmentPushModelSOPInstance*  
*StorageCommitmentPullModelSOPClassRetired*  
*StorageCommitmentPullModelSOPInstanceRetired*  
*ProceduralEventLoggingSOPClass*  
*ProceduralEventLoggingSOPInstance*  
*SubstanceAdministrationLoggingSOPClass*  
*SubstanceAdministrationLoggingSOPInstance*  
*DICOMUIDRegistry*  
*DICOMControlledTerminology*  
*DICOMApplicationContextName*  
*DetachedPatientManagementSOPClassRetired*  
*DetachedPatientManagementMetaSOPClassRetired*  
*DetachedVisitManagementSOPClassRetired*  
*DetachedStudyManagementSOPClassRetired*  
*StudyComponentManagementSOPClassRetired*  
*ModalityPerformedProcedureStepSOPClass*  
*ModalityPerformedProcedureStepRetrieveSOPClass*  
*ModalityPerformedProcedureStepNotificationSOPClass*  
*DetachedResultsManagementSOPClassRetired*  
*DetachedResultsManagementMetaSOPClassRetired*  
*DetachedStudyManagementMetaSOPClassRetired*  
*DetachedInterpretationManagementSOPClassRetired*  
*StorageServiceClass*  
*BasicFilmSessionSOPClass*  
*BasicFilmBoxSOPClass*  
*BasicGrayscaleImageBoxSOPClass*  
*BasicColorImageBoxSOPClass*  
*ReferencedImageBoxSOPClassRetired*  
*BasicGrayscalePrintManagementMetaSOPClass*  
*ReferencedGrayscalePrintManagementMetaSOPClassRetired*  
*PrintJobSOPClass*  
*BasicAnnotationBoxSOPClass*  
*PrinterSOPClass*  
*PrinterConfigurationRetrievalSOPClass*  
*PrinterSOPInstance*  
*PrinterConfigurationRetrievalSOPInstance*  
*BasicColorPrintManagementMetaSOPClass*

*ReferencedColorPrintManagementMetaSOPClassRetired*  
*VOILUTBoxSOPClass*  
*PresentationLUTSOPClass*  
*ImageOverlayBoxSOPClassRetired*  
*BasicPrintImageOverlayBoxSOPClassRetired*  
*PrintQueueSOPInstanceRetired*  
*PrintQueueManagementSOPClassRetired*  
*StoredPrintStorageSOPClassRetired*  
*HardcopyGrayscaleImageStorageSOPClassRetired*  
*HardcopyColorImageStorageSOPClassRetired*  
*PullPrintRequestSOPClassRetired*  
*PullStoredPrintManagementMetaSOPClassRetired*  
*MediaCreationManagementSOPClassUID*  
*ComputedRadiographyImageStorage*  
*DigitalXRayImageStorageForPresentation*  
*DigitalXRayImageStorageForProcessing*  
*DigitalMammographyXRayImageStorageForPresentation*  
*DigitalMammographyXRayImageStorageForProcessing*  
*DigitalIntraoralXRayImageStorageForPresentation*  
*DigitalIntraoralXRayImageStorageForProcessing*  
*CTImageStorage*  
*EnhancedCTImageStorage*  
*UltrasoundMultiframeImageStorageRetired*  
*UltrasoundMultiframeImageStorage*  
*MRIImageStorage*  
*EnhancedMRIImageStorage*  
*MRSpectroscopyStorage*  
*NuclearMedicineImageStorageRetired*  
*UltrasoundImageStorageRetired*  
*UltrasoundImageStorage*  
*SecondaryCaptureImageStorage*  
*MultiframeSingleBitSecondaryCaptureImageStorage*  
*MultiframeGrayscaleByteSecondaryCaptureImageStorage*  
*MultiframeGrayscaleWordSecondaryCaptureImageStorage*  
*MultiframeTrueColorSecondaryCaptureImageStorage*  
*StandaloneOverlayStorageRetired*  
*StandaloneCurveStorageRetired*  
*WaveformStorageTrialRetired*  
*GeneralECGWaveformStorage*  
*AmbulatoryECGWaveformStorage*  
*HemodynamicWaveformStorage*  
*CardiacElectrophysiologyWaveformStorage*



*BasicVoiceAudioWaveformStorage*  
*StandaloneModalityLUTStorageRetired*  
*StandaloneVOILUTStorageRetired*  
*GrayscaleSoftcopyPresentationStateStorageSOPClass*  
*ColorSoftcopyPresentationStateStorageSOPClass*  
*PseudoColorSoftcopyPresentationStateStorageSOPClass*  
*BlendingSoftcopyPresentationStateStorageSOPClass*  
*XRayAngiographicImageStorage*  
*EnhancedXAImageStorage*  
*XRayRadiofluoroscopicImageStorage*  
*EnhancedXRImageStorage*  
*XRay3DAngiographicImageStorage*  
*XRay3DCraniofacialImageStorage*  
*XRayAngiographicBiPlaneImageStorageRetired*  
*NuclearMedicineImageStorage*  
*RawDataStorage*  
*SpatialRegistrationStorage*  
*SpatialFiducialsStorage*  
*DeformableSpatialRegistrationStorage*  
*SegmentationStorage*  
*RealWorldValueMappingStorage*  
*VImageStorageTrialRetired*  
*VMultiframeImageStorageTrialRetired*  
*VLEndoscopicImageStorage*  
*VideoEndoscopicImageStorage*  
*VMicroscopicImageStorage*  
*VideoMicroscopicImageStorage*  
*VLSlideCoordinatesMicroscopicImageStorage*  
*VLPhotographicImageStorage*  
*VideoPhotographicImageStorage*  
*OphthalmicPhotography8BitImageStorage*  
*OphthalmicPhotography16BitImageStorage*  
*StereometricRelationshipStorage*  
*OphthalmicTomographyImageStorage*  
*TextSRStorageTrialRetired*  
*AudioSRStorageTrialRetired*  
*DetailSRStorageTrialRetired*  
*ComprehensiveSRStorageTrialRetired*  
*BasicTextSRStorage*  
*EnhancedSRStorage*  
*ComprehensiveSRStorage*  
*ProcedureLogStorage*

*MammographyCADSRStorage*  
*KeyObjectSelectionDocumentStorage*  
*ChestCADSRStorage*  
*XRayRadiationDoseSRStorage*  
*EncapsulatedPDFStorage*  
*EncapsulatedCDASStorage*  
*PositronEmissionTomographyImageStorage*  
*StandalonePETCurveStorageRetired*  
*RTImageStorage*  
*RTDoseStorage*  
*RTStructureSetStorage*  
*RTBeamsTreatmentRecordStorage*  
*RTPlanStorage*  
*RTBrachyTreatmentRecordStorage*  
*RTTreatmentSummaryRecordStorage*  
*RTIonPlanStorage*  
*RTIonBeamsTreatmentRecordStorage*  
*PatientRootQueryRetrieveInformationModelFIND*  
*PatientRootQueryRetrieveInformationModelMOVE*  
*PatientRootQueryRetrieveInformationModelGET*  
*StudyRootQueryRetrieveInformationModelFIND*  
*StudyRootQueryRetrieveInformationModelMOVE*  
*StudyRootQueryRetrieveInformationModelGET*  
*PatientStudyOnlyQueryRetrieveInformationModelFINDRetired*  
*PatientStudyOnlyQueryRetrieveInformationModelMOVERetired*  
*PatientStudyOnlyQueryRetrieveInformationModelGETRetired*  
*ModalityWorklistInformationModelFIND*  
*GeneralPurposeWorklistInformationModelFIND*  
*GeneralPurposeScheduledProcedureStepSOPClass*  
*GeneralPurposePerformedProcedureStepSOPClass*  
*GeneralPurposeWorklistManagementMetaSOPClass*  
*InstanceAvailabilityNotificationSOPClass*  
*RTBeamsDeliveryInstructionStorageSupplement74FrozenDraft*  
*RTConventionalMachineVerificationSupplement74FrozenDraft*  
*RTIonMachineVerificationSupplement74FrozenDraft*  
*UnifiedWorklistandProcedureStepServiceClass*  
*UnifiedProcedureStepPushSOPClass*  
*UnifiedProcedureStepWatchSOPClass*  
*UnifiedProcedureStepPullSOPClass*  
*UnifiedProcedureStepEventSOPClass*  
*UnifiedWorklistandProcedureStepSOPInstance*  
*GeneralRelevantPatientInformationQuery*

*BreastImagingRelevantPatientInformationQuery*  
*CardiacRelevantPatientInformationQuery*  
*HangingProtocolStorage*  
*HangingProtocolInformationModelFIND*  
*HangingProtocolInformationModelMOVE*  
*ProductCharacteristicsQuerySOPClass*  
*SubstanceApprovalQuerySOPClass*  
*dicomDeviceName*  
*dicomDescription*  
*dicomManufacturer*  
*dicomManufacturerModelName*  
*dicomSoftwareVersion*  
*dicomVendorData*  
*dicomAETitle*  
*dicomNetworkConnectionReference*  
*dicomApplicationCluster*  
*dicomAssociationInitiator*  
*dicomAssociationAcceptor*  
*dicomHostname*  
*dicomPort*  
*dicomSOPClass*  
*dicomTransferRole*  
*dicomTransferSyntax*  
*dicomPrimaryDeviceType*  
*dicomRelatedDeviceReference*  
*dicomPreferredCalledAETitle*  
*dicomTLSCyphersuite*  
*dicomAuthorizedNodeCertificateReference*  
*dicomThisNodeCertificateReference*  
*dicomInstalled*  
*dicomStationName*  
*dicomDeviceSerialNumber*  
*dicomInstitutionName*  
*dicomInstitutionAddress*  
*dicomInstitutionDepartmentName*  
*dicomIssuerOfPatientID*  
*dicomPreferredCallingAETitle*  
*dicomSupportedCharacterSet*  
*dicomConfigurationRoot*  
*dicomDevicesRoot*  
*dicomUniqueAETitlesRegistryRoot*  
*dicomDevice*

*dicomNetworkAE*  
*dicomNetworkConnection*  
*dicomUniqueAETitle*  
*dicomTransferCapability*  
*VLWholeSlideMicroscopyImageStorage*  
*EnhancedUSVolumeStorage*  
*SurfaceSegmentationStorage*  
*BreastTomosynthesisImageStorage*

27.290.3.2 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*  
*uid\_1\_2\_840\_10008\_1\_2\_4\_92*  
*uid\_1\_2\_840\_10008\_1\_2\_4\_93*

*uid\_1\_2\_840\_10008\_1\_2\_4\_94*  
*uid\_1\_2\_840\_10008\_1\_2\_4\_95*  
*uid\_1\_2\_840\_10008\_1\_2\_4\_100*  
*uid\_1\_2\_840\_10008\_1\_2\_5*  
*uid\_1\_2\_840\_10008\_1\_2\_6\_1*  
*uid\_1\_2\_840\_10008\_1\_2\_6\_2*  
*uid\_1\_2\_840\_10008\_1\_3\_10*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_1*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_2*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_3*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_4*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_5*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_6*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_7*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_8*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_9*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_10*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_11*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_12*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_13*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_14*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_15*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_16*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_17*  
*uid\_1\_2\_840\_10008\_1\_4\_1\_18*  
*uid\_1\_2\_840\_10008\_1\_4\_2\_1*  
*uid\_1\_2\_840\_10008\_1\_4\_2\_2*  
*uid\_1\_2\_840\_10008\_1\_9*  
*uid\_1\_2\_840\_10008\_1\_20\_1*  
*uid\_1\_2\_840\_10008\_1\_20\_1\_1*  
*uid\_1\_2\_840\_10008\_1\_20\_2*  
*uid\_1\_2\_840\_10008\_1\_20\_2\_1*  
*uid\_1\_2\_840\_10008\_1\_40*  
*uid\_1\_2\_840\_10008\_1\_40\_1*  
*uid\_1\_2\_840\_10008\_1\_42*  
*uid\_1\_2\_840\_10008\_1\_42\_1*  
*uid\_1\_2\_840\_10008\_2\_6\_1*  
*uid\_1\_2\_840\_10008\_2\_16\_4*  
*uid\_1\_2\_840\_10008\_3\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1*

*uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5*  
*uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1*  
*uid\_1\_2\_840\_10008\_4\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_9*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_14*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_15*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_16*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_17*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_18*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_22*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_23*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_24*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_25*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_26*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_27*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_29*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_30*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_31*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_32*  
*uid\_1\_2\_840\_10008\_5\_1\_1\_33*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1*

*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_8*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_2\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_3\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_1*

*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_67*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_11*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_22*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_33*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_40*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_50*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_59*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_65*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_67*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9*



*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_31*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_32\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_32\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_32\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_32*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_33*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_41*  
*uid\_1\_2\_840\_10008\_5\_1\_4\_42*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_1*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_2*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_3*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_4*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_5*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_6*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_7*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_8*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_9*  
*uid\_1\_2\_840\_10008\_15\_0\_3\_10*

```

uid_1_2_840_10008_15_0_3_11
uid_1_2_840_10008_15_0_3_12
uid_1_2_840_10008_15_0_3_13
uid_1_2_840_10008_15_0_3_14
uid_1_2_840_10008_15_0_3_15
uid_1_2_840_10008_15_0_3_16
uid_1_2_840_10008_15_0_3_17
uid_1_2_840_10008_15_0_3_18
uid_1_2_840_10008_15_0_3_19
uid_1_2_840_10008_15_0_3_20
uid_1_2_840_10008_15_0_3_21
uid_1_2_840_10008_15_0_3_22
uid_1_2_840_10008_15_0_3_23
uid_1_2_840_10008_15_0_3_24
uid_1_2_840_10008_15_0_3_25
uid_1_2_840_10008_15_0_3_26
uid_1_2_840_10008_15_0_3_27
uid_1_2_840_10008_15_0_3_28
uid_1_2_840_10008_15_0_3_29
uid_1_2_840_10008_15_0_3_30
uid_1_2_840_10008_15_0_3_31
uid_1_2_840_10008_15_0_4_1
uid_1_2_840_10008_15_0_4_2
uid_1_2_840_10008_15_0_4_3
uid_1_2_840_10008_15_0_4_4
uid_1_2_840_10008_15_0_4_5
uid_1_2_840_10008_15_0_4_6
uid_1_2_840_10008_15_0_4_7
uid_1_2_840_10008_15_0_4_8
uid_1_2_840_10008_5_1_4_1_1_77_1_6
uid_1_2_840_10008_5_1_4_1_1_6_2
uid_1_2_840_10008_5_1_4_1_1_66_5
uid_1_2_840_10008_5_1_4_1_1_13_1_3

```

#### 27.290.4 Member Function Documentation

##### 27.290.4.1 `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<<()`.

27.290.4.2 static unsigned int gdcm::UIDs::GetNumberOfTransferSyntaxStrings ( ) [static]

27.290.4.3 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<<().

27.290.4.4 static const char\* const\* gdcm::UIDs::GetTransferSyntaxString ( unsigned int *ts* ) [static]

27.290.4.5 static TransferSyntaxStringsType gdcm::UIDs::GetTransferSyntaxStrings ( ) [static]

27.290.4.6 static const char\* gdcm::UIDs::GetUIDName ( unsigned int *ts* ) [static]

27.290.4.7 static const char\* gdcm::UIDs::GetUIDString ( unsigned int *ts* ) [static]

27.290.4.8 gdcm::UIDs::operator TSType ( ) const [inline]

27.290.4.9 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:

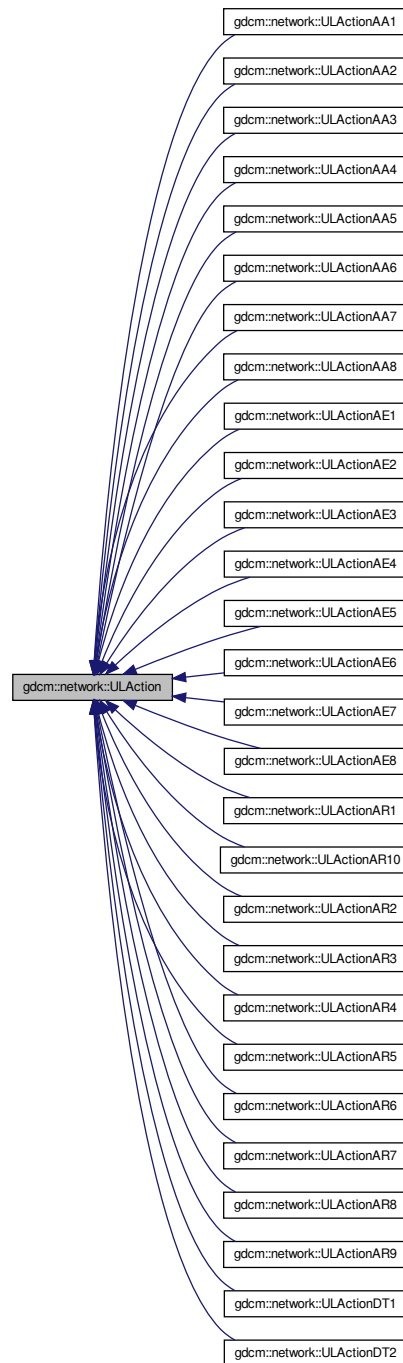
- [gdcmUIDs.h](#)

## 27.291 gdcm::network::ULAction Class Reference

**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](#).

```
#include <gdcmULAction.h>
```

Inheritance diagram for `gdc::network::ULAction`:



## Public Member Functions

- [ULAction](#) ()
- virtual [~ULAction](#) ()

- virtual [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaiting←  
ForEvent, [EEventID](#) &outRaisedEvent)=0

### 27.291.1 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 thos filestreams, AETitles to establish connections, whatever. The actual parameters that the user will pass via an action will come through a Payload object, which should, in itself, be some gdcmm-based object (but not all objects can be payloads; sending a single dataelement as a payload isn't meaningful). As such, each action has its own particular payload.

For the sake of keeping files together, both the particular payload class and the action class will be defined in the same header file. Payloads should JUST be data (or streams), NO METHODS.

Some actions perform changes that should raise events on the local system, and some actions perform changes that will require waiting for events from the remote system.

Therefore, this base action has been modified so that those events are set by each action. When the event loop runs an action, it will then test to see if a local event was raised by the action, and if so, perform the appropriate subsequent action. If the action requires waiting for a response from the remote system, then the event loop will sit there (presumably with the ARTIM timer running) and wait for a response from the remote system. Once a response is obtained, then the the rest of the state transitions can happen.

### 27.291.2 Constructor & Destructor Documentation

27.291.2.1 `gdcmm::network::ULAction::ULAction ( ) [inline]`

27.291.2.2 `virtual gdcmm::network::ULAction::~~ULAction ( ) [inline],[virtual]`

### 27.291.3 Member Function Documentation

27.291.3.1 `virtual EStateID gdcmm::network::ULAction::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [pure virtual]`

Implemented in [gdcmm::network::ULActionAR10](#), [gdcmm::network::ULActionAR9](#), [gdcmm::network::ULActionAE8](#), [gdcmm::network::ULActionAA8](#), [gdcmm::network::ULActionAR8](#), [gdcmm::network::ULActionAE7](#), [gdcmm::network::ULActionA←A7](#), [gdcmm::network::ULActionAR7](#), [gdcmm::network::ULActionAE6](#), [gdcmm::network::ULActionAA6](#), [gdcmm::network::ULActionAR6](#), [gdcmm::network::ULActionAA5](#), [gdcmm::network::ULActionAE5](#), [gdcmm::network::ULActionAR5](#), [gdcmm::network::ULActionAA4](#), [gdcmm::network::ULActionAE4](#), [gdcmm::network::ULActionAR4](#), [gdcmm::network::ULActionA←A3](#), [gdcmm::network::ULActionAE3](#), [gdcmm::network::ULActionAR3](#), [gdcmm::network::ULActionAA2](#), [gdcmm::network::ULActionAE2](#), [gdcmm::network::ULActionAR2](#), [gdcmm::network::ULActionDT2](#), [gdcmm::network::ULActionAA1](#), [gdcmm::network::ULActionAE1](#), [gdcmm::network::ULActionAR1](#), and [gdcmm::network::ULActionDT1](#).

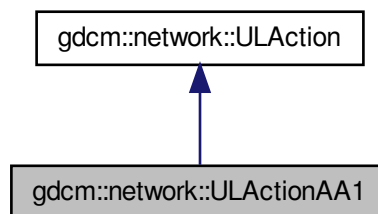
The documentation for this class was generated from the following file:

- [gdcmmULAction.h](#)

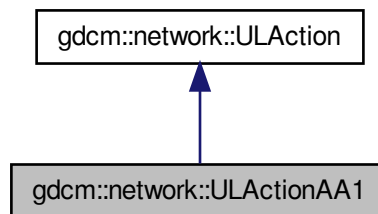
## 27.292 gdcmm::network::ULActionAA1 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA1:



Collaboration diagram for gdcmm::network::ULActionAA1:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.292.1 Member Function Documentation

27.292.1.1 `EStateID gdcmm::network::ULActionAA1::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

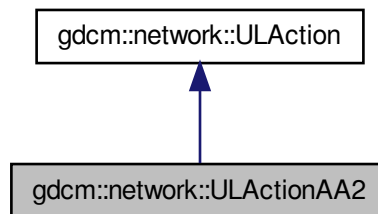
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

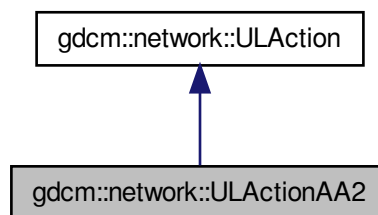
## 27.293 gdcm::network::ULActionAA2 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for gdcm::network::ULActionAA2:



Collaboration diagram for gdcm::network::ULActionAA2:



### Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.293.1 Member Function Documentation

27.293.1.1 **EStateID** `gdcm::network::ULActionAA2::PerformAction ( Subject *s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

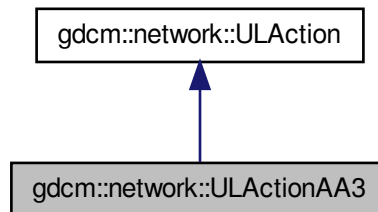
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

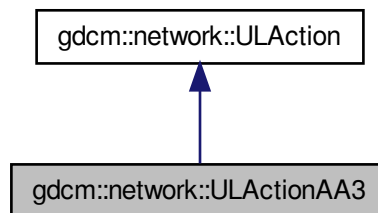
## 27.294 `gdcm::network::ULActionAA3` Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for `gdcm::network::ULActionAA3`:



Collaboration diagram for `gdcm::network::ULActionAA3`:



### Public Member Functions

- [EStateID](#) `PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`



### 27.294.1 Member Function Documentation

27.294.1.1 `EStateID gdcmm::network::ULActionAA3::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

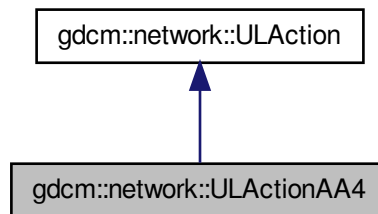
The documentation for this class was generated from the following file:

- [gdcmmULActionAA.h](#)

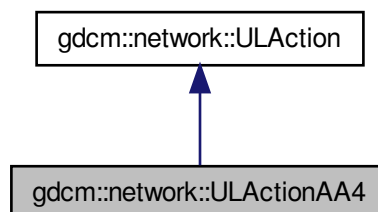
## 27.295 gdcmm::network::ULActionAA4 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA4:



Collaboration diagram for gdcmm::network::ULActionAA4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.295.1 Member Function Documentation

27.295.1.1 [EStateID](#) `gdcm::network::ULActionAA4::PerformAction ( Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

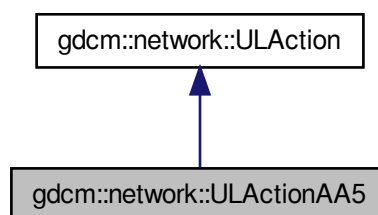
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

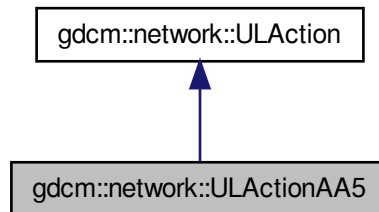
## 27.296 gdcm::network::ULActionAA5 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.296.1 Member Function Documentation

27.296.1.1 **EStateID** `gdcm::network::ULActionAA5::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

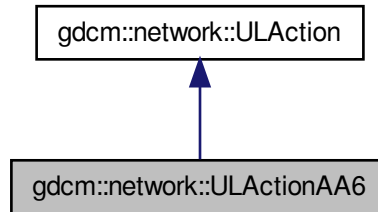
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

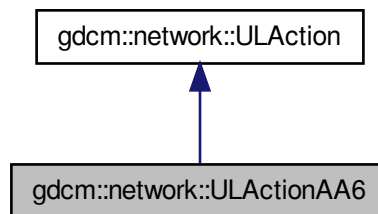
## 27.297 gdcm::network::ULActionAA6 Class Reference

```
#include <gdcmULActionAA.h>
```

Inheritance diagram for `gdcn::network::ULActionAA6`:



Collaboration diagram for `gdcn::network::ULActionAA6`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.297.1 Member Function Documentation

27.297.1.1 `EStateID gdcn::network::ULActionAA6::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

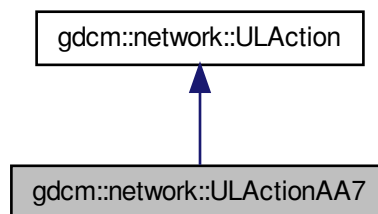
The documentation for this class was generated from the following file:

- [gdcnULActionAA.h](#)

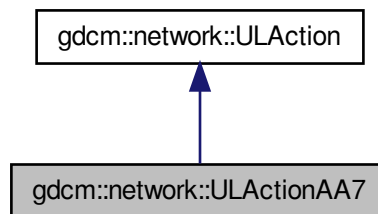
## 27.298 gdcmm::network::ULActionAA7 Class Reference

```
#include <gdcmmULActionAA.h>
```

Inheritance diagram for gdcmm::network::ULActionAA7:



Collaboration diagram for gdcmm::network::ULActionAA7:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.298.1 Member Function Documentation

27.298.1.1 [EStateID gdcmm::network::ULActionAA7::PerformAction](#) ( [Subject](#) \* s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent* ) `[virtual]`

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

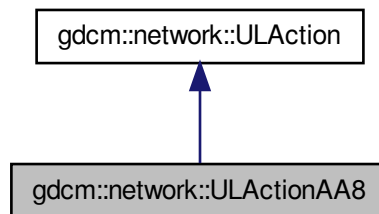
The documentation for this class was generated from the following file:

- [gdcmlActionAA.h](#)

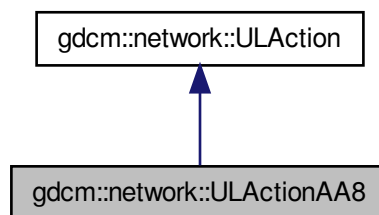
## 27.299 gdcmlnetwork::ULActionAA8 Class Reference

```
#include <gdcmlActionAA.h>
```

Inheritance diagram for gdcmlnetwork::ULActionAA8:



Collaboration diagram for gdcmlnetwork::ULActionAA8:



### Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.299.1 Member Function Documentation

27.299.1.1 `EStateID` `gdcm::network::ULActionAA8::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

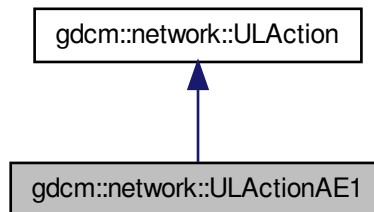
The documentation for this class was generated from the following file:

- [gdcmULActionAA.h](#)

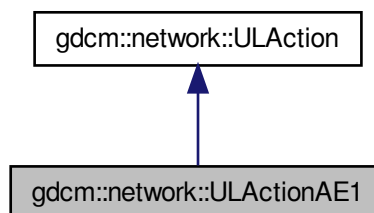
## 27.300 gdcm::network::ULActionAE1 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE1`:



Collaboration diagram for `gdcm::network::ULActionAE1`:



### Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.300.1 Member Function Documentation

27.300.1.1 `EStateID gdcmm::network::ULActionAE1::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

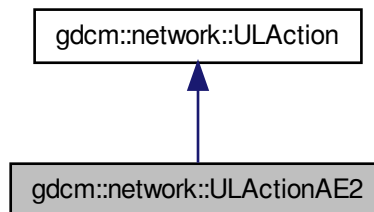
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

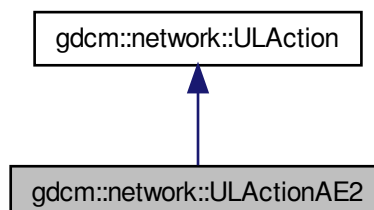
### 27.301 gdcmm::network::ULActionAE2 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE2:



Collaboration diagram for gdcmm::network::ULActionAE2:





## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.301.1 Member Function Documentation

27.301.1.1 [EStateID](#) `gdcm::network::ULActionAE2::PerformAction ( Subject * s, ULError & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

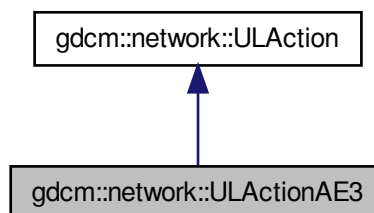
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

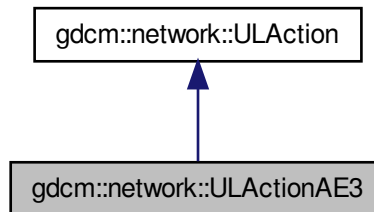
## 27.302 gdcm::network::ULActionAE3 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE3`:



Collaboration diagram for `gdcm::network::ULActionAE3`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.302.1 Member Function Documentation

27.302.1.1 `EStateID gdcm::network::ULActionAE3::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

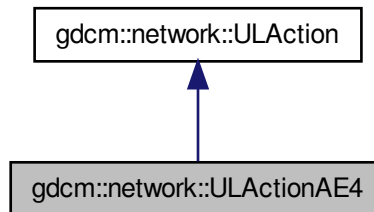
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

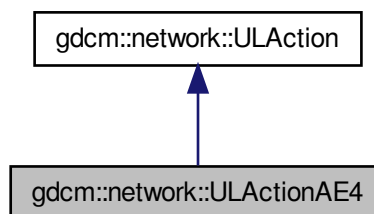
## 27.303 gdcm::network::ULActionAE4 Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for gdcm::network::ULActionAE4:



Collaboration diagram for gdcm::network::ULActionAE4:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.303.1 Member Function Documentation

27.303.1.1 [EStateID](#) [gdcm::network::ULActionAE4::PerformAction](#) ( [Subject](#) \* s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent* ) [virtual]

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

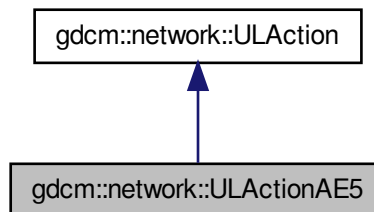
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

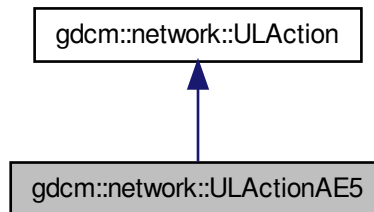
## 27.304 gdcmm::network::ULActionAE5 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE5:



Collaboration diagram for gdcmm::network::ULActionAE5:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.304.1 Member Function Documentation

27.304.1.1 `EStateID gdcmm::network::ULActionAE5::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` `[virtual]`

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

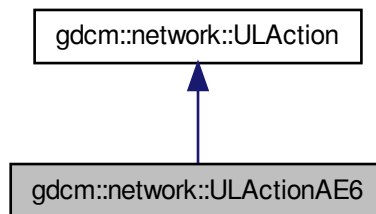
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

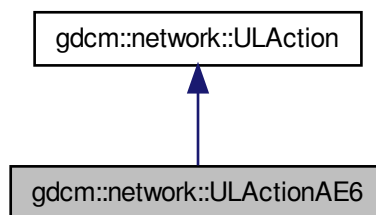
## 27.305 gdcm::network::ULActionAE6 Class Reference

```
#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 &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.305.1 Member Function Documentation

27.305.1.1 **EStateID** `gdcm::network::ULActionAE6::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

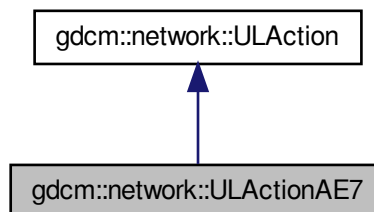
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

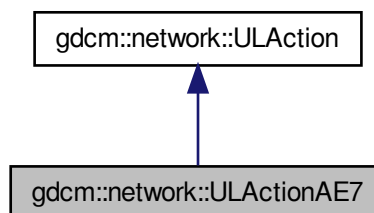
## 27.306 `gdcm::network::ULActionAE7` Class Reference

```
#include <gdcmULActionAE.h>
```

Inheritance diagram for `gdcm::network::ULActionAE7`:



Collaboration diagram for `gdcm::network::ULActionAE7`:



### Public Member Functions

- [EStateID](#) `PerformAction (Subject *s, ULEvent &inEvent, ULConnection &inConnection, bool &outWaitingForEvent, EEventID &outRaisedEvent)`

### 27.306.1 Member Function Documentation

27.306.1.1 `EStateID gdcmm::network::ULActionAE7::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

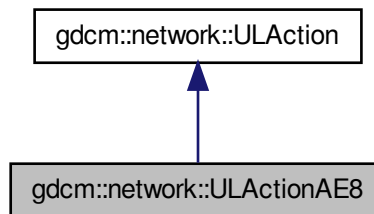
The documentation for this class was generated from the following file:

- [gdcmmULActionAE.h](#)

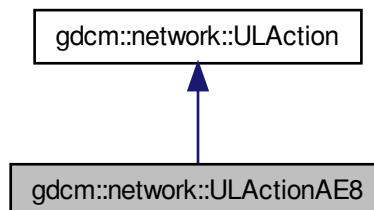
## 27.307 gdcmm::network::ULActionAE8 Class Reference

```
#include <gdcmmULActionAE.h>
```

Inheritance diagram for gdcmm::network::ULActionAE8:



Collaboration diagram for gdcmm::network::ULActionAE8:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.307.1 Member Function Documentation

27.307.1.1 [EStateID](#) `gdcm::network::ULActionAE8::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` `[virtual]`

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

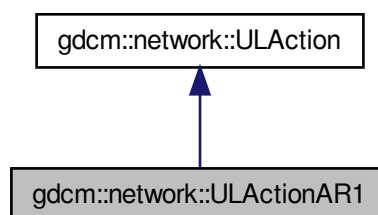
The documentation for this class was generated from the following file:

- [gdcmULActionAE.h](#)

## 27.308 gdcm::network::ULActionAR1 Class Reference

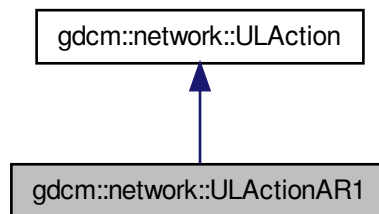
```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR1`:





Collaboration diagram for gdcm::network::ULActionAR1:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.308.1 Member Function Documentation

27.308.1.1 `EStateID gdcm::network::ULActionAR1::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

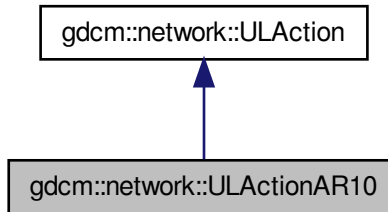
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

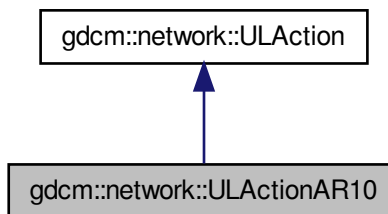
## 27.309 gdcm::network::ULActionAR10 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcn::network::ULActionAR10`:



Collaboration diagram for `gdcn::network::ULActionAR10`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.309.1 Member Function Documentation

27.309.1.1 `EStateID gdcn::network::ULActionAR10::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` `[virtual]`

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

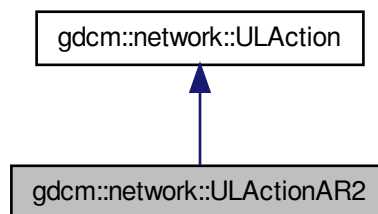
The documentation for this class was generated from the following file:

- [gdcnULActionAR.h](#)

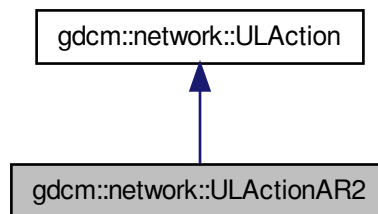
## 27.310 gdcmm::network::ULActionAR2 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR2:



Collaboration diagram for gdcmm::network::ULActionAR2:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.310.1 Member Function Documentation

27.310.1.1 [EStateID gdcmm::network::ULActionAR2::PerformAction](#) ( [Subject](#) \* s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent* ) [virtual]

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

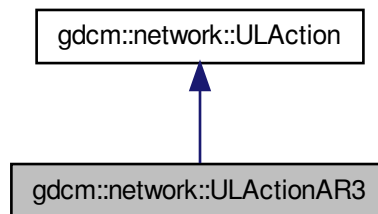
The documentation for this class was generated from the following file:

- [gdcmlActionAR.h](#)

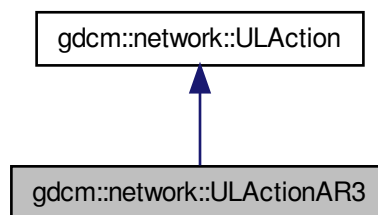
## 27.311 gdcmlnetwork::ULActionAR3 Class Reference

```
#include <gdcmlActionAR.h>
```

Inheritance diagram for gdcmlnetwork::ULActionAR3:



Collaboration diagram for gdcmlnetwork::ULActionAR3:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingFor↵  
Event, [EEventID](#) &outRaisedEvent)

### 27.311.1 Member Function Documentation

27.311.1.1 **EStateID** gdcm::network::ULActionAR3::PerformAction ( **Subject** \* *s*, **UEvent** & *inEvent*, **ULConnection** & *inConnection*, **bool** & *outWaitingForEvent*, **EEventID** & *outRaisedEvent* ) [virtual]

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

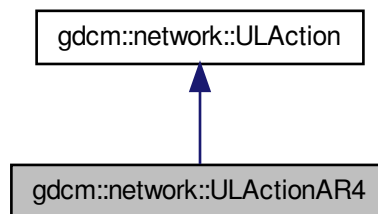
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

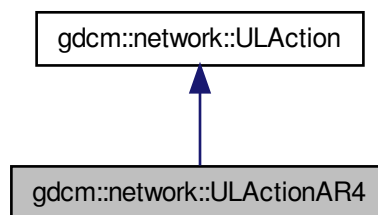
## 27.312 gdcm::network::ULActionAR4 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR4:



Collaboration diagram for gdcm::network::ULActionAR4:



### Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) \**s*, [UEvent](#) &*inEvent*, [ULConnection](#) &*inConnection*, **bool** &*outWaitingForEvent*, [EEventID](#) &*outRaisedEvent*)

### 27.312.1 Member Function Documentation

27.312.1.1 **EStateID** `gdcm::network::ULActionAR4::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` [virtual]

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

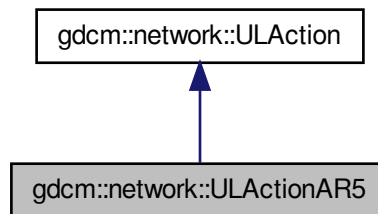
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

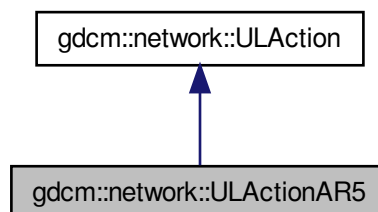
### 27.313 `gdcm::network::ULActionAR5` Class Reference

```
#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)

### 27.313.1 Member Function Documentation

27.313.1.1 [EStateID](#) `gdcm::network::ULActionAR5::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent )` `[virtual]`

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

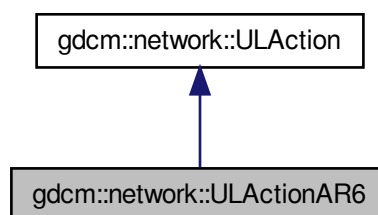
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

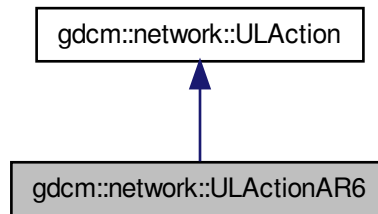
## 27.314 gdcm::network::ULActionAR6 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for `gdcm::network::ULActionAR6`:



Collaboration diagram for `gdcmm::network::ULActionAR6`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.314.1 Member Function Documentation

27.314.1.1 **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:

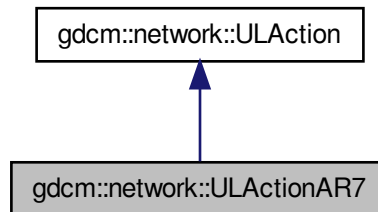
- [gdcmmULActionAR.h](#)

## 27.315 gdcmm::network::ULActionAR7 Class Reference

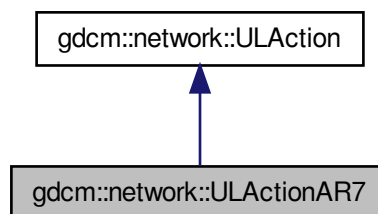
```
#include <gdcmmULActionAR.h>
```



Inheritance diagram for gdcm::network::ULActionAR7:



Collaboration diagram for gdcm::network::ULActionAR7:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

## 27.315.1 Member Function Documentation

27.315.1.1 [EStateID gdcm::network::ULActionAR7::PerformAction](#) ( [Subject](#) \* s, [ULEvent](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent* ) [virtual]

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

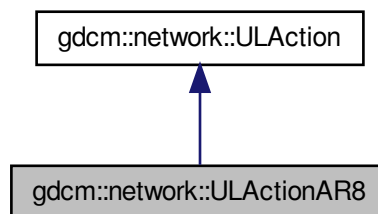
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

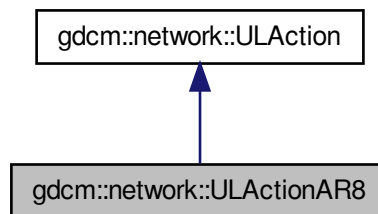
## 27.316 gdcmm::network::ULActionAR8 Class Reference

```
#include <gdcmmULActionAR.h>
```

Inheritance diagram for gdcmm::network::ULActionAR8:



Collaboration diagram for gdcmm::network::ULActionAR8:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.316.1 Member Function Documentation

27.316.1.1 **EStateID** gdcmm::network::ULActionAR8::PerformAction ( **Subject** \* s, **ULEvent** & *inEvent*, **ULConnection** & *inConnection*, bool & *outWaitingForEvent*, **EEventID** & *outRaisedEvent* ) [virtual]

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

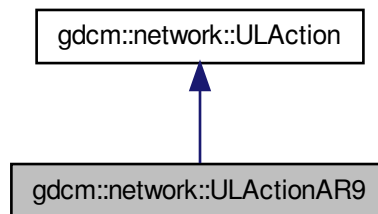
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

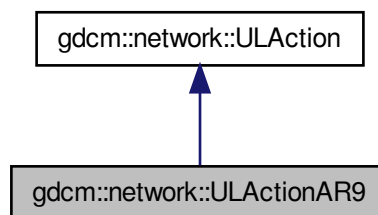
## 27.317 gdcm::network::ULActionAR9 Class Reference

```
#include <gdcmULActionAR.h>
```

Inheritance diagram for gdcm::network::ULActionAR9:



Collaboration diagram for gdcm::network::ULActionAR9:



### Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.317.1 Member Function Documentation

27.317.1.1 **EStateID** `gdcm::network::ULActionAR9::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

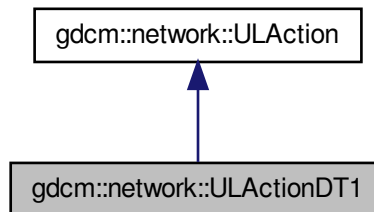
The documentation for this class was generated from the following file:

- [gdcmULActionAR.h](#)

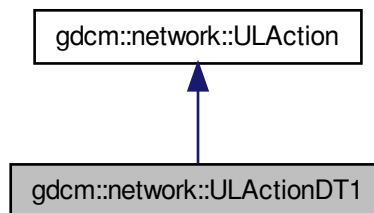
## 27.318 `gdcm::network::ULActionDT1` Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for `gdcm::network::ULActionDT1`:



Collaboration diagram for `gdcm::network::ULActionDT1`:



### Public Member Functions

- [EStateID](#) [PerformAction](#) ([Subject](#) \*s, [ULEvent](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.318.1 Member Function Documentation

27.318.1.1 `EStateID gdcm::network::ULActionDT1::PerformAction ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) [virtual]`

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

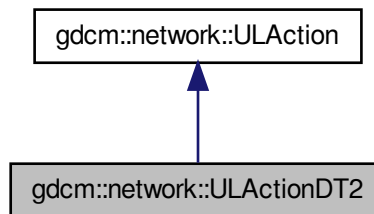
The documentation for this class was generated from the following file:

- [gdcmULActionDT.h](#)

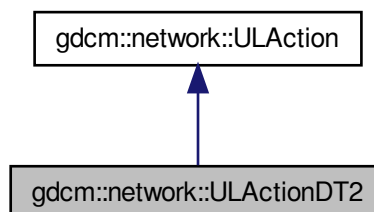
## 27.319 gdcm::network::ULActionDT2 Class Reference

```
#include <gdcmULActionDT.h>
```

Inheritance diagram for `gdcm::network::ULActionDT2`:



Collaboration diagram for `gdcm::network::ULActionDT2`:



## Public Member Functions

- [EStateID PerformAction](#) ([Subject](#) \*s, [ULError](#) &inEvent, [ULConnection](#) &inConnection, bool &outWaitingForEvent, [EEventID](#) &outRaisedEvent)

### 27.319.1 Member Function Documentation

27.319.1.1 [EStateID gdcmm::network::ULActionDT2::PerformAction](#) ( [Subject](#) \* s, [ULError](#) & *inEvent*, [ULConnection](#) & *inConnection*, bool & *outWaitingForEvent*, [EEventID](#) & *outRaisedEvent* ) [virtual]

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

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

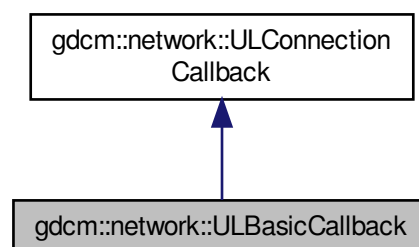
- [gdcmmULActionDT.h](#)

### 27.320 gdcmm::network::ULBasicCallback Class Reference

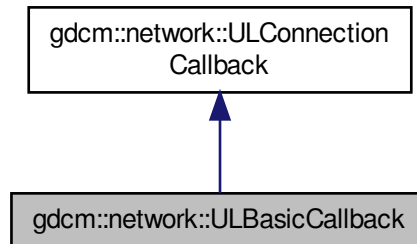
[ULBasicCallback](#) This is the most basic of callbacks for how the [ULConnectionManager](#) handles incoming datasets. DataSets are just concatenated to the mDataSets vector, and the result can be pulled out of the vector by later code. Alternatives to this method include progress updates, saving to disk, etc. This class is NOT THREAD SAFE. Access the dataset vector after the entire set of datasets has been returned by the [ULConnectionManager](#).

```
#include <gdcmmULBasicCallback.h>
```

Inheritance diagram for [gdcmm::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

### 27.320.1 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](#).

### 27.320.2 Constructor & Destructor Documentation

27.320.2.1 `gdcm::network::ULBasicCallback::ULBasicCallback ( )` `[inline]`

27.320.2.2 `virtual gdcm::network::ULBasicCallback::~~ULBasicCallback ( )` `[inline], [virtual]`

### 27.320.3 Member Function Documentation

27.320.3.1 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetDataSets ( )` const

27.320.3.2 `std::vector<DataSet> const& gdcm::network::ULBasicCallback::GetResponses ( )` const

27.320.3.3 `virtual void gdcm::network::ULBasicCallback::HandleDataSet ( const DataSet & inDataSet )` `[virtual]`

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

27.320.3.4 virtual void gdcmm::network::ULBasicCallback::HandleResponse ( const DataSet & inDataSet ) [virtual]

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

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

- [gdcmmULBasicCallback.h](#)

## 27.321 gdcmm::network::ULConnection Class Reference

[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.

```
#include <gdcmmULConnection.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](#)

### 27.321.1 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.

### 27.321.2 Constructor & Destructor Documentation

27.321.2.1 `gdcm::network::ULConnection::ULConnection ( const ULConnectionInfo & inUserInfo )`

27.321.2.2 `virtual gdcm::network::ULConnection::~~ULConnection ( ) [virtual]`

### 27.321.3 Member Function Documentation

27.321.3.1 `void gdcm::network::ULConnection::AddAcceptedPresentationContext ( const PresentationContextAC & inPC )`

27.321.3.2 `PresentationContextRQ gdcm::network::ULConnection::FindContext ( const DataElement & de ) const`

27.321.3.3 `std::vector<PresentationContextAC> const& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( ) const`

27.321.3.4 `std::vector<PresentationContextAC>& gdcm::network::ULConnection::GetAcceptedPresentationContexts ( )`

27.321.3.5 `const ULConnectionInfo & gdcm::network::ULConnection::GetConnectionInfo ( ) const`

27.321.3.6 `uint32_t gdcm::network::ULConnection::GetMaxPDUSize ( ) const`

27.321.3.7 `const PresentationContextAC* gdcm::network::ULConnection::GetPresentationContextACByID ( uint8_t id ) const`

27.321.3.8 `uint8_t gdcm::network::ULConnection::GetPresentationContextIDFromPresentationContext ( PresentationContextRQ const & pc ) const`

return 0 upon error

27.321.3.9 `const PresentationContextRQ* gdcm::network::ULConnection::GetPresentationContextRQByID ( uint8_t id ) const`

27.321.3.10 `std::vector<PresentationContextRQ> const& gdcm::network::ULConnection::GetPresentationContexts ( ) const`

27.321.3.11 `std::iostream* gdcm::network::ULConnection::GetProtocol ( )`

27.321.3.12 `EStateID gdcm::network::ULConnection::GetState ( ) const`

27.321.3.13 `ARTIMTimer& gdcm::network::ULConnection::GetTimer ( )`

27.321.3.14 `bool gdcm::network::ULConnection::InitializeConnection ( )`

used to establish scu connections

27.321.3.15 `bool gdcm::network::ULConnection::InitializeIncomingConnection ( )`

used to establish scp connections

27.321.3.16 `void gdcm::network::ULConnection::SetMaxPDUSize ( uint32_t inSize )`

27.321.3.17 `void gdcm::network::ULConnection::SetPresentationContexts ( const std::vector< PresentationContextRQ > & inContexts )`

27.321.3.18 `void gdcm::network::ULConnection::SetPresentationContexts ( const std::vector< PresentationContext > & inContexts )`

27.321.3.19 `void gdcm::network::ULConnection::SetState ( const EStateID & inState )`

27.321.3.20 `void gdcm::network::ULConnection::StopProtocol ( )`

## 27.321.4 Friends And Related Function Documentation

27.321.4.1 `friend class ULActionAE6 [friend]`

27.321.4.2 `friend class ULConnectionManager [friend]`

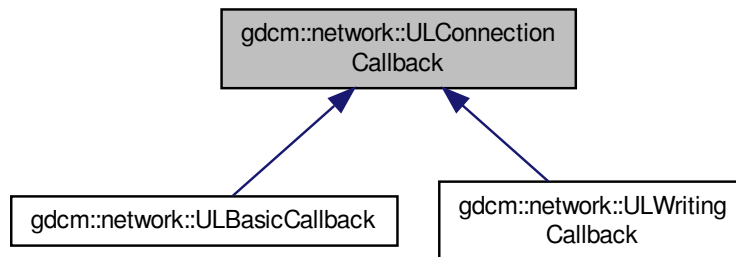
The documentation for this class was generated from the following file:

- [gdcmULConnection.h](#)

## 27.322 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](#)

#### 27.322.1 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.

#### 27.322.2 Constructor & Destructor Documentation

27.322.2.1 `gdcm::network::ULConnectionCallback::ULConnectionCallback ( )` `[inline]`

27.322.2.2 `virtual gdcm::network::ULConnectionCallback::~~ULConnectionCallback ( ) [inline],[virtual]`

### 27.322.3 Member Function Documentation

27.322.3.1 `void gdcm::network::ULConnectionCallback::DataSetHandled ( ) [inline],[protected]`

27.322.3.2 `bool gdcm::network::ULConnectionCallback::DataSetHandles ( ) const [inline]`

27.322.3.3 `virtual void gdcm::network::ULConnectionCallback::HandleDataSet ( const DataSet & inDataSet ) [pure virtual]`

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

27.322.3.4 `virtual void gdcm::network::ULConnectionCallback::HandleResponse ( const DataSet & inDataSet ) [pure virtual]`

Implemented in [gdcm::network::ULBasicCallback](#), and [gdcm::network::ULWritingCallback](#).

27.322.3.5 `void gdcm::network::ULConnectionCallback::ResetHandledDataSet ( ) [inline]`

27.322.3.6 `void gdcm::network::ULConnectionCallback::SetImplicitFlag ( const bool imp ) [inline]`

### 27.322.4 Member Data Documentation

27.322.4.1 `bool gdcm::network::ULConnectionCallback::mImplicit [protected]`

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

- [gdcmULConnectionCallback.h](#)

## 27.323 gdcm::network::ULConnectionInfo Class Reference

[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.

```
#include <gdcmULConnectionInfo.h>
```

### Public Member Functions

- [ULConnectionInfo](#) ( )
- const char \* [GetCalledAETitle](#) ( ) const
- std::string [GetCalledComputerName](#) ( ) const
- unsigned long [GetCalledIPAddress](#) ( ) const
- int [GetCalledIPPort](#) ( ) const
- const char \* [GetCallingAETitle](#) ( ) const
- unsigned long [GetMaxPDULength](#) ( ) const
- bool [Initialize](#) (UserInformation const &inUserInformation, const char \*inCalledAETitle, const char \*inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName)
- void [SetMaxPDULength](#) (unsigned long inMaxPDULength)

### 27.323.1 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.

### 27.323.2 Constructor & Destructor Documentation

27.323.2.1 `gdcm::network::ULConnectionInfo::ULConnectionInfo ( )`

### 27.323.3 Member Function Documentation

27.323.3.1 `const char* gdcm::network::ULConnectionInfo::GetCalledAETitle ( ) const`

27.323.3.2 `std::string gdcm::network::ULConnectionInfo::GetCalledComputerName ( ) const`

27.323.3.3 `unsigned long gdcm::network::ULConnectionInfo::GetCalledIPAddress ( ) const`

27.323.3.4 `int gdcm::network::ULConnectionInfo::GetCalledIPPort ( ) const`

27.323.3.5 `const char* gdcm::network::ULConnectionInfo::GetCallingAETitle ( ) const`

27.323.3.6 `unsigned long gdcm::network::ULConnectionInfo::GetMaxPDULength ( ) const`

27.323.3.7 `bool gdcm::network::ULConnectionInfo::Initialize ( UserInformation const & inUserInformation, const char * inCalledAETitle, const char * inCallingAETitle, unsigned long inCalledIPAddress, int inCalledIPPort, std::string inCalledComputerName )`

27.323.3.8 `void gdcm::network::ULConnectionInfo::SetMaxPDULength ( unsigned long inMaxPDULength )`

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

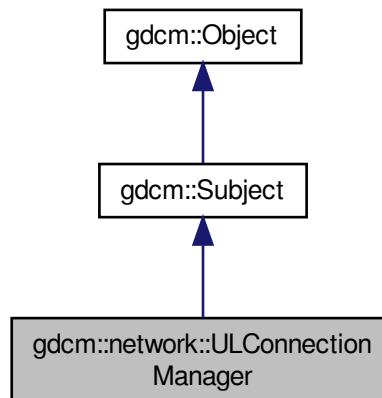
- [gdcmULConnectionInfo.h](#)

## 27.324 gdcm::network::ULConnectionManager Class Reference

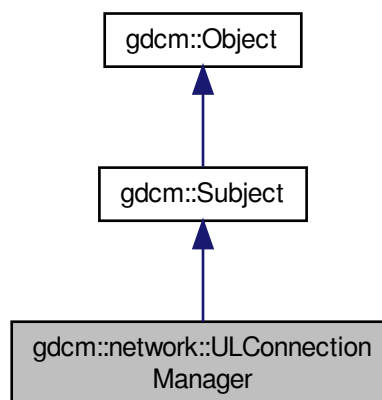
[ULConnectionManager](#) The [ULConnectionManager](#) performs actions on the [ULConnection](#) given inputs from the user and from the state of what's going on around the connection (ie, timeouts of the ARTIM timer, responses from the peer across the connection, etc).

```
#include <gdcmULConnectionManager.h>
```

Inheritance diagram for `gdcm::network::ULConnectionManager`:



Collaboration diagram for `gdcm::network::ULConnectionManager`:



## Public Member Functions

- [ULConnectionManager](#) ()
- [~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](#) > [SendStore](#) (const [File](#) &file)
- void [SendStore](#) (const [File](#) &file, [ULConnectionCallback](#) \*inCallback)  
*callback based API*

## Additional Inherited Members

### 27.324.1 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.

### 27.324.2 Constructor & Destructor Documentation

27.324.2.1 `gdcmm::network::ULConnectionManager::ULConnectionManager ( )`

27.324.2.2 `gdcmm::network::ULConnectionManager::~~ULConnectionManager ( )`

### 27.324.3 Member Function Documentation

27.324.3.1 `bool gdcmm::network::ULConnectionManager::BreakConnection ( const double & inTimeout )`

27.324.3.2 `void gdcmm::network::ULConnectionManager::BreakConnectionNow ( )`

27.324.3.3 `bool gdcmm::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.

27.324.3.4 `bool gdcmm::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

27.324.3.5 `std::vector<PresentationDataValue> gdcmm::network::ULConnectionManager::SendEcho ( )`

27.324.3.6 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendFind ( const BaseRootQuery * inRootQuery )`

27.324.3.7 `void gdcmm::network::ULConnectionManager::SendFind ( const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback )`

27.324.3.8 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendMove ( const BaseRootQuery * inRootQuery )`

27.324.3.9 `bool gdcmm::network::ULConnectionManager::SendMove ( const BaseRootQuery * inRootQuery, ULConnectionCallback * inCallback )`

return false upon error

27.324.3.10 `std::vector<DataSet> gdcmm::network::ULConnectionManager::SendStore ( const File & file )`

27.324.3.11 `void gdcmm::network::ULConnectionManager::SendStore ( const File & file, ULConnectionCallback * inCallback )`

callback based API

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

- [gdcmmULConnectionManager.h](#)

## 27.325 gdcmm::network::ULError Class Reference

[ULError](#) base class for network events.

```
#include <gdcmmULError.h>
```

### Public Member Functions

- [ULError](#) (const [EEventID](#) &*inEventID*, std::vector< [BasePDU](#) \* > const &*inBasePDU*)
- [ULError](#) (const [EEventID](#) &*inEventID*, [BasePDU](#) \**inBasePDU*)
- [~ULError](#) ()
- [EEventID](#) [GetEvent](#) () const
- std::vector< [BasePDU](#) \* > const & [GetPDUs](#) () const
- void [SetEvent](#) (const [EEventID](#) &*inEvent*)
- void [SetPDU](#) (std::vector< [BasePDU](#) \* > const &*inPDU*)

### 27.325.1 Detailed Description

[ULError](#) 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!

### 27.325.2 Constructor & Destructor Documentation

27.325.2.1 `gdcmm::network::ULEvent::ULEvent ( const EEventID & inEventID, std::vector< BasePDU * > const & inBasePDU )` `[inline]`

27.325.2.2 `gdcmm::network::ULEvent::ULEvent ( const EEventID & inEventID, BasePDU * inBasePDU )` `[inline]`

27.325.2.3 `gdcmm::network::ULEvent::~~ULEvent ( )` `[inline]`

### 27.325.3 Member Function Documentation

27.325.3.1 `EEventID gdcmm::network::ULEvent::GetEvent ( ) const` `[inline]`

27.325.3.2 `std::vector<BasePDU*> const& gdcmm::network::ULEvent::GetPDUs ( ) const` `[inline]`

27.325.3.3 `void gdcmm::network::ULEvent::SetEvent ( const EEventID & inEvent )` `[inline]`

27.325.3.4 `void gdcmm::network::ULEvent::SetPDU ( std::vector< BasePDU * > const & inPDU )` `[inline]`

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

- [gdcmmULEvent.h](#)

## 27.326 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, [E↔EventID](#) &outRaisedEvent) const
- void [PrintTable](#) () const

### 27.326.1 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 `TableRow`s. Each row is based on an event, and an event handler in the `Transition↔Table` 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.

## 27.326.2 Constructor & Destructor Documentation

27.326.2.1 `gdcm::network::ULTransitionTable::ULTransitionTable ( )`

## 27.326.3 Member Function Documentation

27.326.3.1 `void gdcm::network::ULTransitionTable::HandleEvent ( Subject * s, ULEvent & inEvent, ULConnection & inConnection, bool & outWaitingForEvent, EEventID & outRaisedEvent ) const`

27.326.3.2 `void gdcm::network::ULTransitionTable::PrintTable ( ) const`

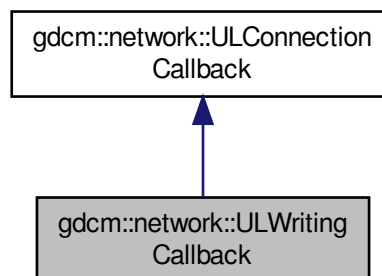
The documentation for this class was generated from the following file:

- [gdcmULTransitionTable.h](#)

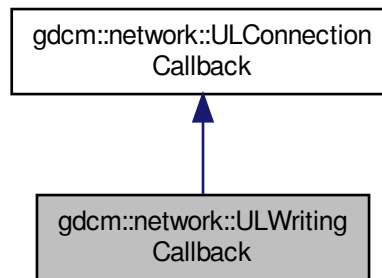
## 27.327 gdcm::network::ULWritingCallback Class Reference

```
#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

### 27.327.1 Constructor & Destructor Documentation

27.327.1.1 `gdcm::network::ULWritingCallback::ULWritingCallback ( )` `[inline]`

27.327.1.2 `virtual gdcm::network::ULWritingCallback::~~ULWritingCallback ( )` `[inline], [virtual]`

### 27.327.2 Member Function Documentation

27.327.2.1 `virtual void gdcm::network::ULWritingCallback::HandleDataSet ( const DataSet & inDataSet )` `[virtual]`

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

27.327.2.2 `virtual void gdcm::network::ULWritingCallback::HandleResponse ( const DataSet & inDataSet )` `[virtual]`

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

27.327.2.3 `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:

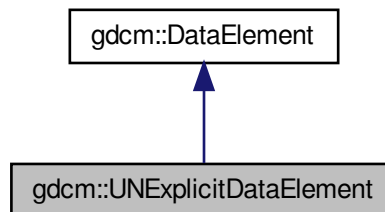
- [gdcmULWritingCallback.h](#)

## 27.328 gdcm::UNExplicitDataElement Class Reference

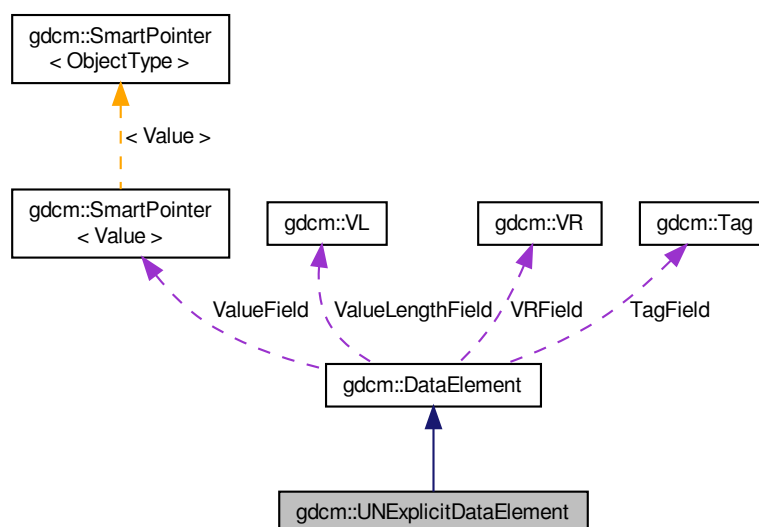
Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

```
#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

### 27.328.1 Detailed Description

Class to read/write a [DataElement](#) as UNExplicit Data [Element](#).

#### Note

bla

### 27.328.2 Member Function Documentation

27.328.2.1 [VL gdcm::UNExplicitDataElement::GetLength \( \) const](#)

27.328.2.2 [template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::Read \( std::istream & is \)](#)

27.328.2.3 [template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadPreValue \( std::istream & is \)](#)

27.328.2.4 [template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadValue \( std::istream & is, bool readvalues = true \)](#)

27.328.2.5 [template<typename TSwap > std::istream& gdcm::UNExplicitDataElement::ReadWithLength \( std::istream & is, \[VL\]\(#\) & length \)](#)

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

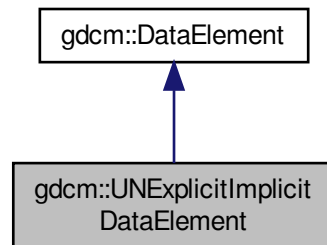
- [gdcmUNExplicitDataElement.h](#)

## 27.329 gdcm::UNExplicitImplicitDataElement Class Reference

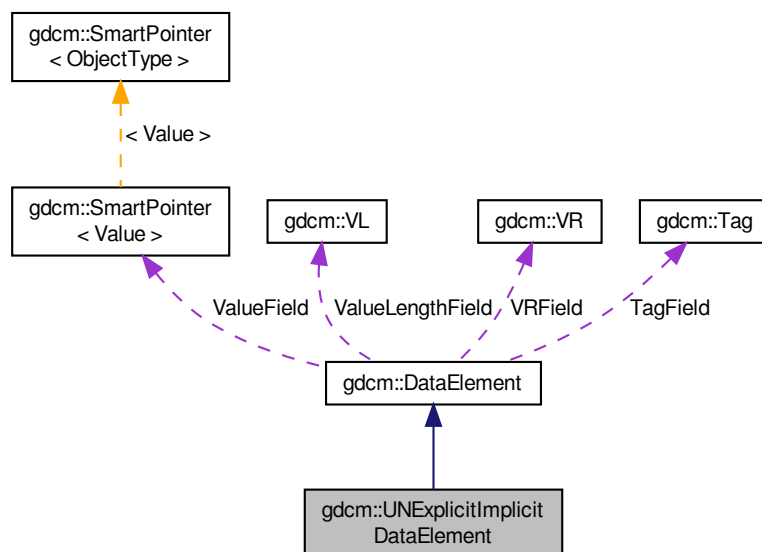
Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#) This class gather two known bugs:

```
#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

### 27.329.1 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/TherapysGDCM120Bug.dcm`

### 27.329.2 Member Function Documentation

27.329.2.1 `VL gdcm::UNExplicitImplicitDataElement::GetLength ( ) const`

27.329.2.2 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::Read ( std::istream & is )`

27.329.2.3 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadPreValue ( std::istream & is )`

27.329.2.4 `template<typename TSwap > std::istream& gdcm::UNExplicitImplicitDataElement::ReadValue ( std::istream & is )`

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

- [gdcmUNExplicitImplicitDataElement.h](#)

## 27.330 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)

### 27.330.1 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](#)

## 27.330.2 Member Function Documentation

**27.330.2.1** `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  $(n / 2) * 3$

**27.330.2.2** `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](#)

## 27.331 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)

## 27.331.1 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.

## 27.331.2 Member Enumeration Documentation

### 27.331.2.1 enum gdcmm::Usage::UsageType

#### Enumerator

***Mandatory***

***Conditional***

***UserOption***

***Invalid***

## 27.331.3 Constructor & Destructor Documentation

27.331.3.1 gdcmm::Usage ( [UsageType type](#) = Invalid ) [inline]

## 27.331.4 Member Function Documentation

27.331.4.1 static const char\* gdcmm::Usage::GetUsageString ( [UsageType type](#) ) [static]

Referenced by gdcmm::operator<<().

27.331.4.2 `static UsageType gdcM::Usage::GetUsageType ( const char * type )` `[static]`

27.331.4.3 `gdcM::Usage::operator UsageType ( ) const` `[inline]`

### 27.331.5 Friends And Related Function Documentation

27.331.5.1 `std::ostream& operator<< ( std::ostream & os, const Usage & vr )` `[friend]`

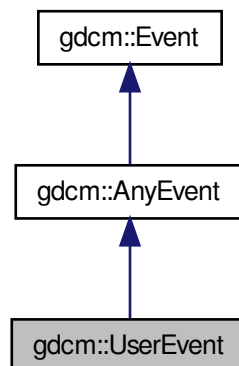
The documentation for this class was generated from the following file:

- [gdcMUsage.h](#)

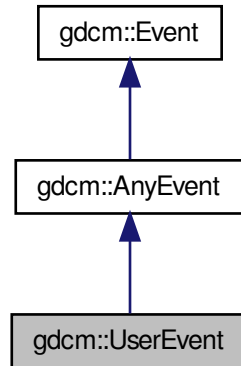
## 27.332 gdcM::UserEvent Class Reference

```
#include <gdcMEvent.h>
```

Inheritance diagram for gdcM::UserEvent:



Collaboration diagram for gdcmm::UserEvent:



### Additional Inherited Members

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

- [gdcmmEvent.h](#)

## 27.333 gdcmm::network::UserInformation Class Reference

[UserInformation](#) Table 9-16 USER INFORMATION ITEM FIELDS.

```
#include <gdcmmUserInformation.h>
```

### Public Member Functions

- [UserInformation](#) ()
- [~UserInformation](#) ()
- void [AddRoleSelectionSub](#) ([RoleSelectionSub](#) const &r)
- void [AddSOPClassExtendedNegociationSub](#) ([SOPClassExtendedNegociationSub](#) const &s)
- const [MaximumLengthSub](#) & [GetMaximumLengthSub](#) () const
- [MaximumLengthSub](#) & [GetMaximumLengthSub](#) ()
- [UserInformation](#) & [operator=](#) (const [UserInformation](#) &)
- void [Print](#) (std::ostream &os) const
- std::istream & [Read](#) (std::istream &is)
- size\_t [Size](#) () const
- const std::ostream & [Write](#) (std::ostream &os) const

### 27.333.1 Detailed Description

[UserInfo](#) Table 9-16 USER INFORMATION ITEM FIELDS.

TODO what is the goal of :

[Table](#) 9-20 USER INFORMATION ITEM FIELDS

### 27.333.2 Constructor & Destructor Documentation

27.333.2.1 `gdcm::network::UserInfo::UserInfo ( )`

27.333.2.2 `gdcm::network::UserInfo::~~UserInfo ( )`

### 27.333.3 Member Function Documentation

27.333.3.1 `void gdcm::network::UserInfo::AddRoleSelectionSub ( RoleSelectionSub const & r )`

27.333.3.2 `void gdcm::network::UserInfo::AddSOPClassExtendedNegociationSub ( SOPClassExtendedNegociationSub const & s )`

27.333.3.3 `const MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( ) const` `[inline]`

27.333.3.4 `MaximumLengthSub& gdcm::network::UserInfo::GetMaximumLengthSub ( )` `[inline]`

27.333.3.5 `UserInfo& gdcm::network::UserInfo::operator= ( const UserInfo & )`

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

27.333.3.7 `std::istream& gdcm::network::UserInfo::Read ( std::istream & is )`

27.333.3.8 `size_t gdcm::network::UserInfo::Size ( ) const`

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

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

- [gdcmUserInfo.h](#)

## 27.334 gdcm::UUIDGenerator Class Reference

Class for generating unique UUID generate DCE 1.1 uid.

```
#include <gdcmUUIDGenerator.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.*

### 27.334.1 Detailed Description

Class for generating unique UUID generate DCE 1.1 uid.

### 27.334.2 Member Function Documentation

#### 27.334.2.1 const char\* gdcm::UUIDGenerator::Generate ( )

Return the generated uuid NOT THREAD SAFE

#### 27.334.2.2 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:

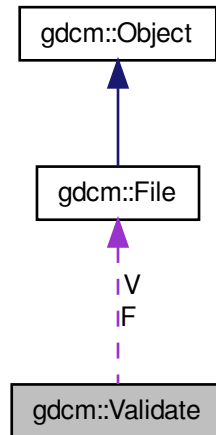
- [gdcmUUIDGenerator.h](#)

## 27.335 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](#)

### 27.335.1 Detailed Description

[Validate](#) class.

### 27.335.2 Constructor & Destructor Documentation

27.335.2.1 `gdcm::Validate::Validate ( )`

27.335.2.2 `gdcm::Validate::~~Validate ( )`

### 27.335.3 Member Function Documentation

27.335.3.1 `const File& gdcm::Validate::GetValidatedFile ( ) [inline]`

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

27.335.3.3 `void gdcm::Validate::Validation ( )`

#### 27.335.4 Member Data Documentation

27.335.4.1 `const File* gdcm::Validate::F [protected]`

27.335.4.2 `File gdcm::Validate::V [protected]`

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

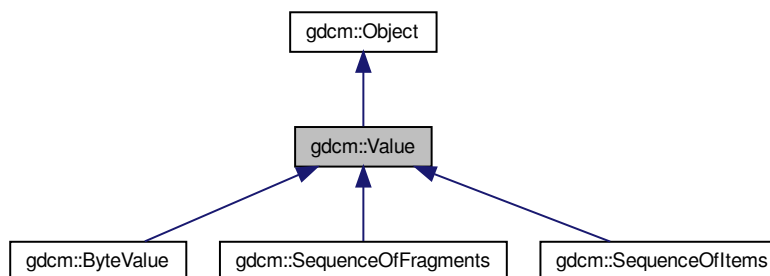
- [gdcmValidate.h](#)

## 27.336 gdcm::Value Class Reference

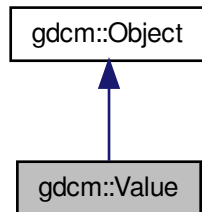
Class to represent the value of a Data [Element](#).

```
#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](#)

## 27.336.1 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.

## 27.336.2 Constructor & Destructor Documentation

27.336.2.1 `gdcm::Value::Value ( )` [`inline`]

27.336.2.2 `gdcm::Value::~~Value ( )` [`inline`]

## 27.336.3 Member Function Documentation



27.336.3.1 virtual void gdcm::Value::Clear ( ) [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

27.336.3.2 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().

27.336.3.3 virtual bool gdcm::Value::operator== ( const Value & val ) const [pure virtual]

Implemented in [gdcm::SequenceOfFragments](#), [gdcm::SequenceOfItems](#), and [gdcm::ByteValue](#).

27.336.3.4 virtual void gdcm::Value::SetLength ( VL / ) [pure virtual]

Implemented in [gdcm::ByteValue](#), [gdcm::SequenceOfItems](#), and [gdcm::SequenceOfFragments](#).

27.336.3.5 virtual void gdcm::Value::SetLengthOnly ( VL / ) [protected],[virtual]

Reimplemented in [gdcm::ByteValue](#).

## 27.336.4 Friends And Related Function Documentation

27.336.4.1 friend class DataElement [friend]

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

- [gdcmValue.h](#)

## 27.337 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)

### 27.337.1 Detailed Description

```
template<typename TDE, typename TSwap, typename TType = uint8_t>class gdcm::ValueIO< TDE, TSwap, TType >
```

Class to dispatch template calls.

## 27.337.2 Member Function Documentation

27.337.2.1 `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]`

27.337.2.2 `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](#)

## 27.338 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)

## 27.338.1 Detailed Description

major/minor and build version

## 27.338.2 Constructor & Destructor Documentation

27.338.2.1 `gdcM::Version::Version ( ) [inline]`

27.338.2.2 `gdcM::Version::~~Version ( ) [inline]`

## 27.338.3 Member Function Documentation

27.338.3.1 static int gdcm::Version::GetBuildVersion ( ) [static]

27.338.3.2 static int gdcm::Version::GetMajorVersion ( ) [static]

27.338.3.3 static int gdcm::Version::GetMinorVersion ( ) [static]

27.338.3.4 static const char\* gdcm::Version::GetVersion ( ) [static]

27.338.3.5 void gdcm::Version::Print ( std::ostream & os = std::cout ) const

Referenced by gdcm::operator<<().

## 27.338.4 Friends And Related Function Documentation

27.338.4.1 std::ostream& operator<< ( std::ostream & \_os, const Version & v ) [friend]

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

- [gdcmVersion.h](#)

## 27.339 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)`

## 27.339.1 Detailed Description

[Value](#) Length.

### Warning

this is a 4bytes value ! Do not try to use it for 2bytes value length

### Examples:

[DecompressImageMultiframe.cs](#), [DecompressJPEGFile.cs](#), [FileChangeTS.cs](#), [MpegVideoInfo.cs](#), [New↔Sequence.cs](#), and [rle2img.cxx](#).

## 27.339.2 Member Typedef Documentation

### 27.339.2.1 `typedef uint32_t gdcm::VL::Type`

## 27.339.3 Constructor & Destructor Documentation

### 27.339.3.1 `gdcm::VL::VL ( uint32_t vl = 0 ) [inline]`

## 27.339.4 Member Function Documentation

### 27.339.4.1 `VL gdcm::VL::GetLength ( ) const [inline]`

Referenced by `gdcm::FileMetaInformation::GetFullLength()`, and `gdcm::Item::Write()`.

### 27.339.4.2 `static uint16_t gdcm::VL::GetVL16Max ( ) [inline], [static]`

### 27.339.4.3 `static uint32_t gdcm::VL::GetVL32Max ( ) [inline], [static]`

### 27.339.4.4 `bool gdcm::VL::IsOdd ( ) const [inline]`

Return whether or not the [VL](#) is odd or not.

Referenced by `gdcm::ByteValue::SetLength()`.

27.339.4.5 `bool gdcm::VL::IsUndefined ( ) const [inline]`

Referenced by `gdcm::ByteValue::SetLength()`.

27.339.4.6 `gdcm::VL::operator uint32_t ( ) const [inline]`

27.339.4.7 `VL& gdcm::VL::operator++ ( ) [inline]`

27.339.4.8 `VL gdcm::VL::operator++ ( int ) [inline]`

27.339.4.9 `VL& gdcm::VL::operator+= ( VL const & v/ ) [inline]`

`+=` operator

27.339.4.10 `template<typename TSwap > std::istream& gdcm::VL::Read ( std::istream & is ) [inline]`

27.339.4.11 `template<typename TSwap > std::istream& gdcm::VL::Read16 ( std::istream & is ) [inline]`

27.339.4.12 `void gdcm::VL::SetToUndefined ( ) [inline]`

27.339.4.13 `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()`.

27.339.4.14 `template<typename TSwap > const std::ostream& gdcm::VL::Write16 ( std::ostream & os ) const [inline]`

## 27.339.5 Friends And Related Function Documentation

27.339.5.1 `std::ostream& operator<< ( std::ostream & os, const VL & v/ ) [friend]`

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

- [gdcmVL.h](#)

## 27.340 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 unsigned int [GetNumberOfElementsFromArray](#) (const char \*array, unsigned int 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)

## 27.340.1 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

## 27.340.2 Member Enumeration Documentation

### 27.340.2.1 enum gdcm::VM::VMType

Enumerator

**VM0**  
**VM1**  
**VM2**  
**VM3**  
**VM4**  
**VM5**  
**VM6**  
**VM8**  
**VM9**  
**VM10**  
**VM12**  
**VM16**  
**VM18**  
**VM24**  
**VM28**  
**VM32**

**VM35**  
**VM99**  
**VM256**  
**VM1\_2**  
**VM1\_3**  
**VM1\_4**  
**VM1\_5**  
**VM1\_8**  
**VM1\_32**  
**VM1\_99**  
**VM1\_n**  
**VM2\_2n**  
**VM2\_n**  
**VM3\_4**  
**VM3\_3n**  
**VM3\_n**  
**VM4\_4n**  
**VM6\_6n**  
**VM7\_7n**  
**VM30\_30n**  
**VM47\_47n**  
**VM\_END**

### 27.340.3 Constructor & Destructor Documentation

27.340.3.1 `gdcm::VM::VM ( VMType type = VM0 ) [inline]`

### 27.340.4 Member Function Documentation

27.340.4.1 `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.

27.340.4.2 `static unsigned int gdcm::VM::GetIndex ( VMType vm ) [static], [protected]`

27.340.4.3 `unsigned int gdcm::VM::GetLength ( ) const`

27.340.4.4 `static unsigned int gdcm::VM::GetNumberOfElementsFromArray ( const char * array, unsigned int length ) [static]`

27.340.4.5 `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<<()`.



27.340.4.6 static VMType gdcm::VM::GetVMType ( const char \* *vm* ) [static]

27.340.4.7 static VMType gdcm::VM::GetVMTypeFromLength ( unsigned int *length*, unsigned int *size* ) [static]

27.340.4.8 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

27.340.4.9 gdcm::VM::operator VMType ( ) const [inline]

## 27.340.5 Friends And Related Function Documentation

27.340.5.1 std::ostream& operator<< ( std::ostream & *os*, const VM & *vm* ) [friend]

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

- [gdcmVM.h](#)

## 27.341 gdcm::VMToLength< T > Struct Template Reference

```
#include <gdcmVM.h>
```

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

- [gdcmVM.h](#)

## 27.342 gdcm::VR Class Reference

**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.

```
#include <gdcmVR.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,
  - `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 | 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 | OW | SL | SQ | SS | UL | UN | US,
  - `VR_VM1` = AS | LT | ST | UT | SQ | OF | OD | OW | OB | UN,
  - `VRALL` = VRASCII | VRBINARY,
  - `VR_END` = UT+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)

## 27.342.1 Detailed Description

[VR](#) class This is adapted from DICOM standard The biggest difference is the INVALID [VR](#) and the composite one that differ from standard (more like an addition) This allow us to represent all the possible case express in the DICOMV3 dict.

#### Note

VALUE REPRESENTATION ([VR](#)) Specifies the data type and format of the Value(s) contained in the [Value](#) Field of a Data [Element](#). VALUE REPRESENTATION FIELD: The field where the [Value](#) Representation of a Data [Element](#) is stored in the encoding of a Data [Element](#) structure with explicit [VR](#).

#### Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [NewSequence.cs](#).

## 27.342.2 Member Enumeration Documentation

### 27.342.2.1 enum gdcm::VR::VRType

#### Enumerator

**INVALID**  
**AE**  
**AS**  
**AT**  
**CS**

*DA*  
*DS*  
*DT*  
*FD*  
*FL*  
*IS*  
*LO*  
*LT*  
*OB*  
*OD*  
*OF*  
*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*

### 27.342.3 Constructor & Destructor Documentation

27.342.3.1 `gdcm::VR::VR ( VRType vr = INVALID )` `[inline]`

### 27.342.4 Member Function Documentation

27.342.4.1 `static bool gdcm::VR::CanDisplay ( VRType vr )` `[static]`

27.342.4.2 `bool gdcm::VR::Compatible ( VR const & vr ) const`

27.342.4.3 `int gdcm::VR::GetLength ( ) const [inline]`

27.342.4.4 `static uint32_t gdcm::VR::GetLength ( VRType vr ) [inline],[static]`

27.342.4.5 `unsigned int gdcm::VR::GetSize ( ) const [inline]`

References AE, US\_SS, and VRTypeTemplateCase.

27.342.4.6 `unsigned int gdcm::VR::GetSizeof ( ) const`

27.342.4.7 `static const char* gdcm::VR::GetVRString ( VRType vr ) [static]`

Referenced by `gdcm::operator<<()`.

27.342.4.8 `static const char* gdcm::VR::GetVRStringFromFile ( VRType vr ) [static]`

27.342.4.9 `static VRType gdcm::VR::GetVRType ( const char * vr ) [static]`

27.342.4.10 `static VRType gdcm::VR::GetVRTypeFromFile ( const char * vr ) [static]`

27.342.4.11 `static bool gdcm::VR::IsASCII ( VRType vr ) [static]`

27.342.4.12 `static bool gdcm::VR::IsASCII2 ( VRType vr ) [static]`

27.342.4.13 `static bool gdcm::VR::IsBinary ( VRType vr ) [static]`

27.342.4.14 `static bool gdcm::VR::IsBinary2 ( VRType vr ) [static]`

27.342.4.15 `bool gdcm::VR::IsDual ( ) const`

27.342.4.16 `static bool gdcm::VR::IsSwap ( const char * vr ) [static]`

27.342.4.17 `static bool gdcm::VR::IsValid ( const char * vr ) [static]`

27.342.4.18 `static bool gdcm::VR::IsValid ( const char * vr1, VRType vr2 ) [static]`

27.342.4.19 `bool gdcm::VR::IsVRFile ( ) const`

Referenced by `gdcm::DataElement::SetVR()`.

27.342.4.20 `gdcm::VR::operator VRType ( ) const [inline]`

27.342.4.21 `std::istream& gdcm::VR::Read ( std::istream & is ) [inline]`

References `gdcmDebugMacro`, `INVALID`, and `VR_END`.

27.342.4.22 `const std::ostream& gdc::VR::Write ( std::ostream & os ) const` `[inline]`

References `gdc::AssertAlwaysMacro`, and `INVALID`.

## 27.342.5 Friends And Related Function Documentation

27.342.5.1 `std::ostream& operator<< ( std::ostream & os, const VR & vr )` `[friend]`

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

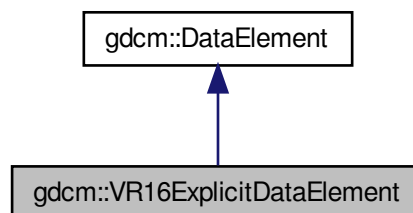
- [gdc::VR.h](#)

## 27.343 gdc::VR16ExplicitDataElement Class Reference

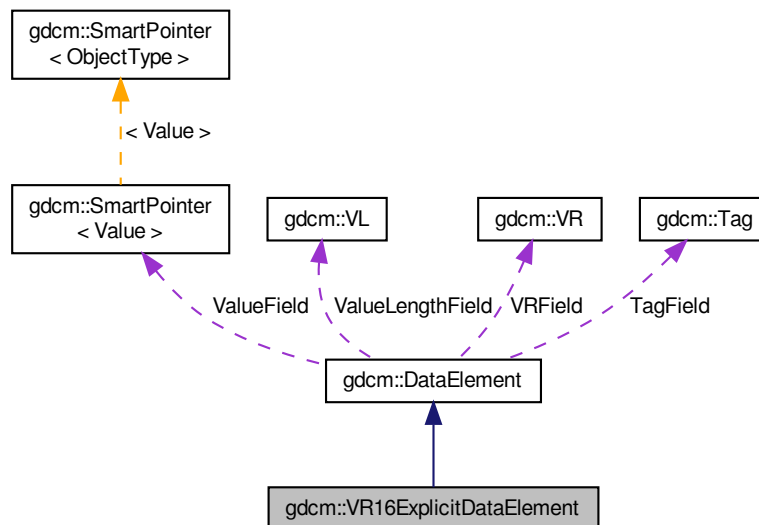
Class to read/write a [DataElement](#) as Explicit Data [Element](#).

```
#include <gdcVR16ExplicitDataElement.h>
```

Inheritance diagram for `gdc::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

### 27.343.1 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

### 27.343.2 Member Function Documentation

#### 27.343.2.1 VL gdcm::VR16ExplicitDataElement::GetLength ( ) const

27.343.2.2 `template<typename TSwap > std::istream& gdcM::VR16ExplicitDataElement::Read ( std::istream & is )`

27.343.2.3 `template<typename TSwap > std::istream& gdcM::VR16ExplicitDataElement::ReadPreValue ( std::istream & is )`

27.343.2.4 `template<typename TSwap > std::istream& gdcM::VR16ExplicitDataElement::ReadValue ( std::istream & is, bool readvalues = true )`

27.343.2.5 `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](#)

## 27.344 gdcM::VRToEncoding< T > Struct Template Reference

```
#include <gdcMVR.h>
```

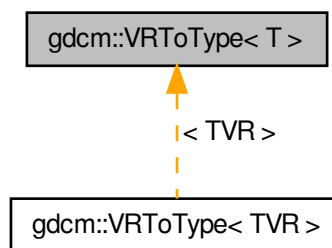
The documentation for this struct was generated from the following file:

- [gdcMVR.h](#)

## 27.345 gdcM::VRToType< T > Struct Template Reference

```
#include <gdcMVR.h>
```

Inheritance diagram for gdcM::VRToType< T >:



### 27.345.1 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](#)

## 27.346 gdcm::VRVLSIZE< T > Class Template Reference

```
#include <gdcmAttribute.h>
```

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

- [gdcmAttribute.h](#)

## 27.347 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)

### 27.347.1 Member Function Documentation

27.347.1.1 static uint16\_t gdcm::VRVLSIZE< 0 >::Read ( std::istream &\_is ) [inline], [static]

27.347.1.2 static void gdcm::VRVLSIZE< 0 >::Write ( std::ostream &os ) [inline], [static]

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

- [gdcmAttribute.h](#)

## 27.348 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)

### 27.348.1 Member Function Documentation

27.348.1.1 static uint32\_t gdcm::VRVLSIZE< 1 >::Read ( std::istream &\_is ) [inline], [static]

27.348.1.2 static void gdcm::VRVLSIZE< 1 >::Write ( std::ostream &os ) [inline], [static]

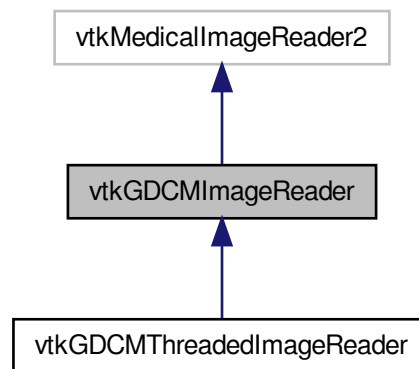
The documentation for this class was generated from the following file:

- [gdcmlAttribute.h](#)

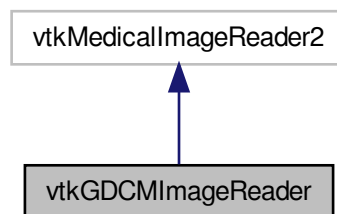
## 27.349 vtkGDCMImageReader Class Reference

```
#include <vtkGDCMImageReader.h>
```

Inheritance diagram for vtkGDCMImageReader:



Collaboration diagram for vtkGDCMImageReader:



### Public Member Functions

- virtual int [CanReadFile](#) (const char \*fname)
- virtual const char \* [GetDescriptiveName](#) ()
- virtual const char \* [GetFileExtensions](#) ()
- vtkImageData \* [GetIconImage](#) ()

- vtkImageData \* [GetOverlay](#) (int i)
- virtual void [PrintSelf](#) (ostream &os, vtkIndent indent)
- virtual void [SetCurve](#) (vtkPolyData \*pd)
- virtual void [SetFileNames](#) (vtkStringArray \*)
- virtual void [SetMedicalImageProperties](#) (vtkMedicalImageProperties \*pd)
- [vtkBooleanMacro](#) (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)
- [vtkTypeRevisionMacro](#) (vtkGDCMImageReader, vtkMedicalImageReader2)

## Static Public Member Functions

- static [vtkGDCMImageReader](#) \* [New](#) ()

## Protected Member Functions

- [vtkGDCMImageReader](#) ()
- [~vtkGDCMImageReader](#) ()
- void [ExecuteData](#) (vtkDataObject \*out)
- void [ExecuteInformation](#) ()
- void [FillMedicalImageInformation](#) (const [gdcm::ImageReader](#) &reader)
- int [LoadSingleFile](#) (const char \*filename, char \*pointer, unsigned long &outlen)
- int [RequestDataCompat](#) ()
- int [RequestInformationCompat](#) ()
- void [SetFilePattern](#) (const char \*)
- void [SetFilePrefix](#) (const char \*)
- [vtkGetStringMacro](#) (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](#)

## 27.349.1 Detailed Description

### Examples:

[AWTMedical3.java](#), [Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmtexture.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](#).

## 27.349.2 Constructor & Destructor Documentation

27.349.2.1 `vtkGDCMImageReader::vtkGDCMImageReader ( )` [protected]

27.349.2.2 `vtkGDCMImageReader::~~vtkGDCMImageReader ( )` [protected]

## 27.349.3 Member Function Documentation

27.349.3.1 `virtual int vtkGDCMImageReader::CanReadFile ( const char * fname )` [virtual]

27.349.3.2 `void vtkGDCMImageReader::ExecuteData ( vtkDataObject * out )` [protected]

27.349.3.3 `void vtkGDCMImageReader::ExecuteInformation ( )` [protected]

- 27.349.3.4 `void vtkGDCMImageReader::FillMedicalImageInformation ( const gdcm::ImageReader & reader )`  
[protected]
- 27.349.3.5 `virtual const char* vtkGDCMImageReader::GetDescriptiveName ( )` [inline],[virtual]
- 27.349.3.6 `virtual const char* vtkGDCMImageReader::GetFileExtensions ( )` [inline],[virtual]
- 27.349.3.7 `vtkImageData* vtkGDCMImageReader::GetIconImage ( )`
- 27.349.3.8 `vtkImageData* vtkGDCMImageReader::GetOverlay ( int i )`
- 27.349.3.9 `int vtkGDCMImageReader::LoadSingleFile ( const char * filename, char * pointer, unsigned long & outlen )`  
[protected]
- 27.349.3.10 `static vtkGDCMImageReader* vtkGDCMImageReader::New ( )` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorthoplanes.cxx](#), [gdcmreslice.cxx](#), [gdcmtexture.cxx](#), [gdcmvolume.cxx](#), [MagnifyFile.cxx](#), [offscreenimage.cxx](#), and [reslicesphere.cxx](#).

- 27.349.3.11 `virtual void vtkGDCMImageReader::PrintSelf ( ostream & os, vtkIndent indent )` [virtual]

Reimplemented in [vtkGDCMThreadedImageReader](#).

- 27.349.3.12 `int vtkGDCMImageReader::RequestDataCompat ( )` [protected]
- 27.349.3.13 `int vtkGDCMImageReader::RequestInformationCompat ( )` [protected]
- 27.349.3.14 `virtual void vtkGDCMImageReader::SetCurve ( vtkPolyData * pd )` [virtual]
- 27.349.3.15 `virtual void vtkGDCMImageReader::SetFileNames ( vtkStringArray * )` [virtual]

Examples:

[gdcmorthoplanes.cxx](#).

- 27.349.3.16 `void vtkGDCMImageReader::SetFilePattern ( const char * )` [inline],[protected]
- 27.349.3.17 `void vtkGDCMImageReader::SetFilePrefix ( const char * )` [inline],[protected]
- 27.349.3.18 `virtual void vtkGDCMImageReader::SetMedicalImageProperties ( vtkMedicalImageProperties * pd )` [virtual]
- 27.349.3.19 `vtkGDCMImageReader::vtkBooleanMacro ( LoadOverlays , int )`
- 27.349.3.20 `vtkGDCMImageReader::vtkBooleanMacro ( LoadIconImage , int )`
- 27.349.3.21 `vtkGDCMImageReader::vtkBooleanMacro ( LossyFlag , int )`

- 27.349.3.22 `vtkGDCMImageReader::vtkBooleanMacro ( ApplyLookupTable , int )`
- 27.349.3.23 `int vtkGDCMImageReader::vtkBooleanMacro ( ApplyYBRToRGB , int )`
- 27.349.3.24 `vtkGDCMImageReader::vtkGetMacro ( LoadOverlays , int )`
- 27.349.3.25 `vtkGDCMImageReader::vtkGetMacro ( LoadIconImage , int )`
- 27.349.3.26 `vtkGDCMImageReader::vtkGetMacro ( LossyFlag , int )`
- 27.349.3.27 `vtkGDCMImageReader::vtkGetMacro ( NumberOfOverlays , int )`
- 27.349.3.28 `vtkGDCMImageReader::vtkGetMacro ( NumberOfIconImages , int )`
- 27.349.3.29 `vtkGDCMImageReader::vtkGetMacro ( ApplyLookupTable , int )`
- 27.349.3.30 `vtkGDCMImageReader::vtkGetMacro ( ApplyYBRToRGB , int )`
- 27.349.3.31 `vtkGDCMImageReader::vtkGetMacro ( ImageFormat , int )`
- 27.349.3.32 `vtkGDCMImageReader::vtkGetMacro ( PlanarConfiguration , int )`
- 27.349.3.33 `vtkGDCMImageReader::vtkGetMacro ( Shift , double )`
- 27.349.3.34 `vtkGDCMImageReader::vtkGetMacro ( Scale , double )`
- 27.349.3.35 `vtkGDCMImageReader::vtkGetObjectMacro ( DirectionCosines , vtkMatrix4x4 )`
- 27.349.3.36 `vtkGDCMImageReader::vtkGetObjectMacro ( MedicalImageProperties , vtkMedicalImageProperties )`
- 27.349.3.37 `vtkGDCMImageReader::vtkGetObjectMacro ( FileNames , vtkStringArray )`
- 27.349.3.38 `vtkGDCMImageReader::vtkGetObjectMacro ( Curve , vtkPolyData )`
- 27.349.3.39 `vtkGDCMImageReader::vtkGetStringMacro ( FilePrefix ) [protected]`
- 27.349.3.40 `vtkGDCMImageReader::vtkGetStringMacro ( FilePattern ) [protected]`
- 27.349.3.41 `vtkGDCMImageReader::vtkGetVector3Macro ( ImagePositionPatient , double )`
- 27.349.3.42 `vtkGDCMImageReader::vtkGetVector6Macro ( ImageOrientationPatient , double )`
- 27.349.3.43 `vtkGDCMImageReader::vtkSetMacro ( LoadOverlays , int )`
- 27.349.3.44 `vtkGDCMImageReader::vtkSetMacro ( LoadIconImage , int )`
- 27.349.3.45 `vtkGDCMImageReader::vtkSetMacro ( LossyFlag , int )`
- 27.349.3.46 `vtkGDCMImageReader::vtkSetMacro ( ApplyLookupTable , int )`
- 27.349.3.47 `vtkGDCMImageReader::vtkSetVector6Macro ( ImageOrientationPatient , double ) [protected]`

27.349.3.48 `vtkGDCMImageReader::vtkTypeRevisionMacro ( vtkGDCMImageReader , vtkMedicalImageReader2 )`

#### 27.349.4 Member Data Documentation

27.349.4.1 `int vtkGDCMImageReader::ApplyInverseVideo` [protected]

27.349.4.2 `int vtkGDCMImageReader::ApplyLookupTable` [protected]

27.349.4.3 `int vtkGDCMImageReader::ApplyPlanarConfiguration` [protected]

27.349.4.4 `int vtkGDCMImageReader::ApplyShiftScale` [protected]

27.349.4.5 `int vtkGDCMImageReader::ApplyYBRToRGB` [protected]

27.349.4.6 `vtkPolyData* vtkGDCMImageReader::Curve` [protected]

27.349.4.7 `vtkMatrix4x4* vtkGDCMImageReader::DirectionCosines` [protected]

27.349.4.8 `vtkStringArray* vtkGDCMImageReader::FileNames` [protected]

27.349.4.9 `int vtkGDCMImageReader::ForceRescale` [protected]

27.349.4.10 `int vtkGDCMImageReader::IconDataScalarType` [protected]

27.349.4.11 `int vtkGDCMImageReader::IconImageDataExtent[6]` [protected]

27.349.4.12 `int vtkGDCMImageReader::IconNumberOfScalarComponents` [protected]

27.349.4.13 `int vtkGDCMImageReader::ImageFormat` [protected]

27.349.4.14 `double vtkGDCMImageReader::ImageOrientationPatient[6]` [protected]

27.349.4.15 `double vtkGDCMImageReader::ImagePositionPatient[3]` [protected]

27.349.4.16 `int vtkGDCMImageReader::LoadIconImage` [protected]

27.349.4.17 `int vtkGDCMImageReader::LoadOverlays` [protected]

27.349.4.18 `int vtkGDCMImageReader::LossyFlag` [protected]

27.349.4.19 `vtkMedicalImageProperties* vtkGDCMImageReader::MedicalImageProperties` [protected]

27.349.4.20 `int vtkGDCMImageReader::NumberOfIconImages` [protected]

27.349.4.21 `int vtkGDCMImageReader::NumberOfOverlays` [protected]

27.349.4.22 `int vtkGDCMImageReader::PlanarConfiguration` [protected]

27.349.4.23 `double vtkGDCMImageReader::Scale` [protected]

27.349.4.24 `double vtkGDCMImageReader::Shift` [protected]

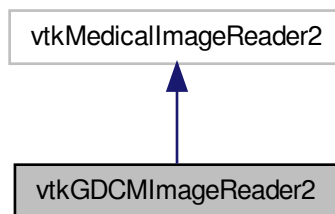
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader.h](#)

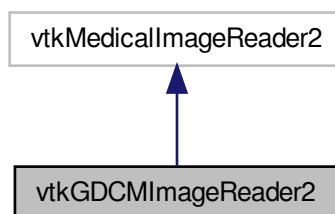
## 27.350 `vtkGDCMImageReader2` Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (vtkGDCMImageReader2, vtkMedicalImageReader2)

### Static Public Member Functions

- static [vtkGDCMImageReader2](#) \* [New](#) ()

### Protected Member Functions

- [vtkGDCMImageReader2](#) ()
- [~vtkGDCMImageReader2](#) ()
- void [FillMedicalImageInformation](#) (const [gdcmm::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](#)

## 27.350.1 Constructor & Destructor Documentation

27.350.1.1 [vtkGDCMImageReader2::vtkGDCMImageReader2](#) ( ) [protected]

27.350.1.2 [vtkGDCMImageReader2::~~vtkGDCMImageReader2](#) ( ) [protected]

## 27.350.2 Member Function Documentation

27.350.2.1 [virtual int vtkGDCMImageReader2::CanReadFile](#) ( const char \* *fname* ) [virtual]

27.350.2.2 [void vtkGDCMImageReader2::FillMedicalImageInformation](#) ( const [gdcm::ImageReader](#) & *reader* ) [protected]

27.350.2.3 [virtual const char\\* vtkGDCMImageReader2::GetDescriptiveName](#) ( ) [inline],[virtual]

27.350.2.4 [virtual const char\\* vtkGDCMImageReader2::GetFileExtensions](#) ( ) [inline],[virtual]

27.350.2.5 [vtkImageData\\* vtkGDCMImageReader2::GetIconImage](#) ( )

- 27.350.2.6 `vtkAlgorithmOutput* vtkGDCMImageReader2::GetIconImagePort ( )`
- 27.350.2.7 `vtkImageData* vtkGDCMImageReader2::GetOverlay ( int i )`
- 27.350.2.8 `vtkAlgorithmOutput* vtkGDCMImageReader2::GetOverlayPort ( int index )`
- 27.350.2.9 `int vtkGDCMImageReader2::LoadSingleFile ( const char * filename, char * pointer, unsigned long & outlen )`  
[protected]
- 27.350.2.10 `static vtkGDCMImageReader2* vtkGDCMImageReader2::New ( )` [static]
- 27.350.2.11 `virtual void vtkGDCMImageReader2::PrintSelf ( ostream & os, vtkIndent indent )` [virtual]
- 27.350.2.12 `int vtkGDCMImageReader2::ProcessRequest ( vtkInformation * request, vtkInformationVector ** inputVector,  
vtkInformationVector * outputVector )` [protected]
- 27.350.2.13 `int vtkGDCMImageReader2::RequestData ( vtkInformation * request, vtkInformationVector ** inputVector,  
vtkInformationVector * outputVector )` [protected]
- 27.350.2.14 `int vtkGDCMImageReader2::RequestDataCompat ( )` [protected]
- 27.350.2.15 `int vtkGDCMImageReader2::RequestInformation ( vtkInformation * request, vtkInformationVector ** inputVector,  
vtkInformationVector * outputVector )` [protected]
- 27.350.2.16 `int vtkGDCMImageReader2::RequestInformationCompat ( )` [protected]
- 27.350.2.17 `virtual void vtkGDCMImageReader2::SetCurve ( vtkPolyData * pd )` [virtual]
- 27.350.2.18 `void vtkGDCMImageReader2::SetFilePattern ( const char * )` [inline],[protected]
- 27.350.2.19 `void vtkGDCMImageReader2::SetFilePrefix ( const char * )` [inline],[protected]
- 27.350.2.20 `virtual void vtkGDCMImageReader2::SetMedicalImageProperties ( vtkMedicalImageProperties * pd )` [virtual]
- 27.350.2.21 `vtkGDCMImageReader2::vtkBooleanMacro ( LoadOverlays , int )`
- 27.350.2.22 `vtkGDCMImageReader2::vtkBooleanMacro ( LoadIconImage , int )`
- 27.350.2.23 `vtkGDCMImageReader2::vtkBooleanMacro ( LossyFlag , int )`
- 27.350.2.24 `vtkGDCMImageReader2::vtkBooleanMacro ( ApplyLookupTable , int )`
- 27.350.2.25 `int vtkGDCMImageReader2::vtkBooleanMacro ( ApplyYBRToRGB , int )`
- 27.350.2.26 `vtkGDCMImageReader2::vtkGetMacro ( LoadOverlays , int )`
- 27.350.2.27 `vtkGDCMImageReader2::vtkGetMacro ( LoadIconImage , int )`
- 27.350.2.28 `vtkGDCMImageReader2::vtkGetMacro ( LossyFlag , int )`
- 27.350.2.29 `vtkGDCMImageReader2::vtkGetMacro ( NumberOfOverlays , int )`

- 27.350.2.30 `vtkGDCMImageReader2::vtkGetMacro ( NumberOfIconImages , int )`
- 27.350.2.31 `vtkGDCMImageReader2::vtkGetMacro ( ApplyLookupTable , int )`
- 27.350.2.32 `vtkGDCMImageReader2::vtkGetMacro ( ApplyYBRToRGB , int )`
- 27.350.2.33 `vtkGDCMImageReader2::vtkGetMacro ( ImageFormat , int )`
- 27.350.2.34 `vtkGDCMImageReader2::vtkGetMacro ( PlanarConfiguration , int )`
- 27.350.2.35 `vtkGDCMImageReader2::vtkGetMacro ( Shift , double )`
- 27.350.2.36 `vtkGDCMImageReader2::vtkGetMacro ( Scale , double )`
- 27.350.2.37 `vtkGDCMImageReader2::vtkGetObjectMacro ( DirectionCosines , vtkMatrix4x4 )`
- 27.350.2.38 `vtkGDCMImageReader2::vtkGetObjectMacro ( Curve , vtkPolyData )`
- 27.350.2.39 `vtkGDCMImageReader2::vtkGetStringMacro ( FilePrefix ) [protected]`
- 27.350.2.40 `vtkGDCMImageReader2::vtkGetStringMacro ( FilePattern ) [protected]`
- 27.350.2.41 `vtkGDCMImageReader2::vtkGetVector3Macro ( ImagePositionPatient , double )`
- 27.350.2.42 `vtkGDCMImageReader2::vtkGetVector6Macro ( ImageOrientationPatient , double )`
- 27.350.2.43 `vtkGDCMImageReader2::vtkSetMacro ( LoadOverlays , int )`
- 27.350.2.44 `vtkGDCMImageReader2::vtkSetMacro ( LoadIconImage , int )`
- 27.350.2.45 `vtkGDCMImageReader2::vtkSetMacro ( LossyFlag , int )`
- 27.350.2.46 `vtkGDCMImageReader2::vtkSetMacro ( ApplyLookupTable , int )`
- 27.350.2.47 `vtkGDCMImageReader2::vtkSetVector6Macro ( ImageOrientationPatient , double ) [protected]`
- 27.350.2.48 `vtkGDCMImageReader2::vtkTypeRevisionMacro ( vtkGDCMImageReader2 , vtkMedicalImageReader2 )`

### 27.350.3 Member Data Documentation

- 27.350.3.1 `int vtkGDCMImageReader2::ApplyInverseVideo [protected]`
- 27.350.3.2 `int vtkGDCMImageReader2::ApplyLookupTable [protected]`
- 27.350.3.3 `int vtkGDCMImageReader2::ApplyPlanarConfiguration [protected]`
- 27.350.3.4 `int vtkGDCMImageReader2::ApplyShiftScale [protected]`
- 27.350.3.5 `int vtkGDCMImageReader2::ApplyYBRToRGB [protected]`
- 27.350.3.6 `vtkPolyData* vtkGDCMImageReader2::Curve [protected]`

- 27.350.3.7 `vtkMatrix4x4*` `vtkGDCMImageReader2::DirectionCosines` [protected]
- 27.350.3.8 `int` `vtkGDCMImageReader2::ForceRescale` [protected]
- 27.350.3.9 `int` `vtkGDCMImageReader2::IconDataScalarType` [protected]
- 27.350.3.10 `int` `vtkGDCMImageReader2::IconImageDataExtent[6]` [protected]
- 27.350.3.11 `int` `vtkGDCMImageReader2::IconNumberOfScalarComponents` [protected]
- 27.350.3.12 `int` `vtkGDCMImageReader2::ImageFormat` [protected]
- 27.350.3.13 `double` `vtkGDCMImageReader2::ImageOrientationPatient[6]` [protected]
- 27.350.3.14 `double` `vtkGDCMImageReader2::ImagePositionPatient[3]` [protected]
- 27.350.3.15 `int` `vtkGDCMImageReader2::LoadIconImage` [protected]
- 27.350.3.16 `int` `vtkGDCMImageReader2::LoadOverlays` [protected]
- 27.350.3.17 `int` `vtkGDCMImageReader2::LossyFlag` [protected]
- 27.350.3.18 `int` `vtkGDCMImageReader2::NumberOfIconImages` [protected]
- 27.350.3.19 `int` `vtkGDCMImageReader2::NumberOfOverlays` [protected]
- 27.350.3.20 `int` `vtkGDCMImageReader2::PlanarConfiguration` [protected]
- 27.350.3.21 `double` `vtkGDCMImageReader2::Scale` [protected]
- 27.350.3.22 `double` `vtkGDCMImageReader2::Shift` [protected]

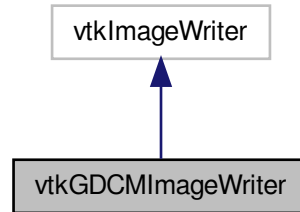
The documentation for this class was generated from the following file:

- [vtkGDCMImageReader2.h](#)

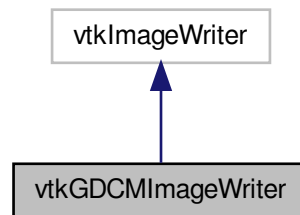
## 27.351 vtkGDCMImageWriter Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (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)

### 27.351.1 Detailed Description

#### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), [HelloActiviz.cs](#), [HelloActiviz2.cs](#), [HelloVTKWorld.cs](#), [HelloVTKWorld.java](#), [HelloVTKWorld2.cs](#), [MagnifyFile.cxx](#), and [RefCounting.cs](#).

## 27.351.2 Member Enumeration Documentation

### 27.351.2.1 enum vtkGDCMImageWriter::CompressionTypes

Enumerator

***NO\_COMPRESSION***  
***JPEG\_COMPRESSION***  
***JPEG2000\_COMPRESSION***  
***JPEGLS\_COMPRESSION***  
***RLE\_COMPRESSION***

## 27.351.3 Constructor & Destructor Documentation

27.351.3.1 `vtkGDCMImageWriter::vtkGDCMImageWriter ( )` [protected]

27.351.3.2 `vtkGDCMImageWriter::~~vtkGDCMImageWriter ( )` [protected]

## 27.351.4 Member Function Documentation

27.351.4.1 `virtual const char* vtkGDCMImageWriter::GetDescriptiveName ( )` [inline],[virtual]

27.351.4.2 `virtual const char* vtkGDCMImageWriter::GetFileExtensions ( )` [inline],[virtual]

27.351.4.3 `virtual char* vtkGDCMImageWriter::GetFileName ( )` [protected],[virtual]

27.351.4.4 `static vtkGDCMImageWriter* vtkGDCMImageWriter::New ( )` [static]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

27.351.4.5 `virtual void vtkGDCMImageWriter::PrintSelf ( ostream & os, vtkIndent indent )` [virtual]

27.351.4.6 `virtual void vtkGDCMImageWriter::SetDirectionCosines ( vtkMatrix4x4 * matrix )` [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

27.351.4.7 `virtual void vtkGDCMImageWriter::SetDirectionCosinesFromImageOrientationPatient ( const double dircos[6] )` [virtual]

27.351.4.8 `virtual void vtkGDCMImageWriter::SetFileNames ( vtkStringArray * )` [virtual]

Examples:

[ConvertMultiFrameToSingleFrame.cxx](#).



27.351.4.9 virtual void vtkGDCMImageWriter::SetMedicalImageProperties ( vtkMedicalImageProperties \* ) [virtual]

Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

27.351.4.10 vtkGDCMImageWriter::vtkBooleanMacro ( LossyFlag , int )

27.351.4.11 vtkGDCMImageWriter::vtkBooleanMacro ( FileLowerLeft , int )

27.351.4.12 vtkGDCMImageWriter::vtkGetMacro ( LossyFlag , int )

27.351.4.13 vtkGDCMImageWriter::vtkGetMacro ( Shift , double )

27.351.4.14 vtkGDCMImageWriter::vtkGetMacro ( Scale , double )

27.351.4.15 vtkGDCMImageWriter::vtkGetMacro ( ImageFormat , int )

27.351.4.16 vtkGDCMImageWriter::vtkGetMacro ( FileLowerLeft , int )

27.351.4.17 vtkGDCMImageWriter::vtkGetMacro ( PlanarConfiguration , int )

27.351.4.18 vtkGDCMImageWriter::vtkGetMacro ( CompressionType , int )

27.351.4.19 vtkGDCMImageWriter::vtkGetObjectMacro ( MedicalImageProperties , vtkMedicalImageProperties )

27.351.4.20 vtkGDCMImageWriter::vtkGetObjectMacro ( FileNames , vtkStringArray )

27.351.4.21 vtkGDCMImageWriter::vtkGetObjectMacro ( DirectionCosines , vtkMatrix4x4 )

27.351.4.22 vtkGDCMImageWriter::vtkGetStringMacro ( StudyUID )

27.351.4.23 vtkGDCMImageWriter::vtkGetStringMacro ( SeriesUID )

27.351.4.24 vtkGDCMImageWriter::vtkSetMacro ( LossyFlag , int )

27.351.4.25 vtkGDCMImageWriter::vtkSetMacro ( Shift , double )

27.351.4.26 vtkGDCMImageWriter::vtkSetMacro ( Scale , double )

27.351.4.27 vtkGDCMImageWriter::vtkSetMacro ( ImageFormat , int )

27.351.4.28 vtkGDCMImageWriter::vtkSetMacro ( FileLowerLeft , int )

27.351.4.29 vtkGDCMImageWriter::vtkSetMacro ( PlanarConfiguration , int )

27.351.4.30 vtkGDCMImageWriter::vtkSetMacro ( CompressionType , int )

27.351.4.31 vtkGDCMImageWriter::vtkSetStringMacro ( StudyUID )

27.351.4.32 `vtkGDCMImageWriter::vtkSetStringMacro ( SeriesUID )`

27.351.4.33 `vtkGDCMImageWriter::vtkTypeRevisionMacro ( vtkGDCMImageWriter , vtkImageWriter )`

27.351.4.34 `virtual void vtkGDCMImageWriter::Write ( )` [virtual]

#### Examples:

[Convert16BitsTo8Bits.cxx](#), [ConvertMultiFrameToSingleFrame.cxx](#), [ConvertRGBToLuminance.cxx](#), [ConvertSingleBitTo8Bits.cxx](#), [gdcmorphoplanes.cxx](#), and [MagnifyFile.cxx](#).

27.351.4.35 `int vtkGDCMImageWriter::WriteGDCMData ( vtkImageData * data, int timeStep )` [protected]

27.351.4.36 `void vtkGDCMImageWriter::WriteSlice ( vtkImageData * data )` [protected]

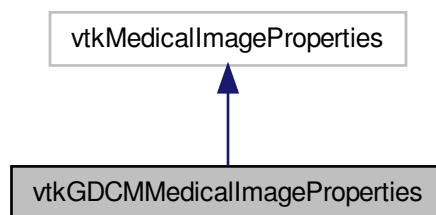
The documentation for this class was generated from the following file:

- [vtkGDCMImageWriter.h](#)

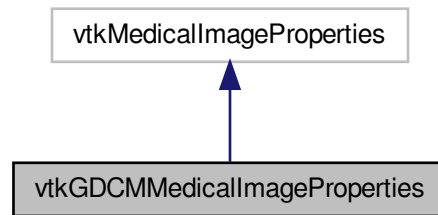
## 27.352 vtkGDCMMedicalImageProperties Class Reference

```
#include <vtkGDCMMedicalImageProperties.h>
```

Inheritance diagram for vtkGDCMMedicalImageProperties:



Collaboration diagram for vtkGDCMMedicalImageProperties:



### Public Member Functions

- virtual void [Clear](#) ()
- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([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](#)

## 27.352.1 Constructor & Destructor Documentation

27.352.1.1 `vtkGDCMMedicalImageProperties::vtkGDCMMedicalImageProperties ( )` [protected]

27.352.1.2 `vtkGDCMMedicalImageProperties::~~vtkGDCMMedicalImageProperties ( )` [protected]

## 27.352.2 Member Function Documentation

- 27.352.2.1 `virtual void vtkGDCMMedicalImageProperties::Clear ( )` `[virtual]`
- 27.352.2.2 `gdcmm::File const& vtkGDCMMedicalImageProperties::GetFile ( unsigned int t )` `[protected]`
- 27.352.2.3 `static vtkGDCMMedicalImageProperties* vtkGDCMMedicalImageProperties::New ( )` `[static]`
- 27.352.2.4 `void vtkGDCMMedicalImageProperties::PrintSelf ( ostream & os, vtkIndent indent )`
- 27.352.2.5 `void vtkGDCMMedicalImageProperties::PushBackFile ( gdcmm::File const & f )` `[protected]`
- 27.352.2.6 `vtkGDCMMedicalImageProperties::vtkTypeRevisionMacro ( vtkGDCMMedicalImageProperties ,  
vtkMedicalImageProperties )`

### 27.352.3 Friends And Related Function Documentation

- 27.352.3.1 `friend class vtkGDCMImageReader` `[friend]`
- 27.352.3.2 `friend class vtkGDCMImageReader2` `[friend]`
- 27.352.3.3 `friend class vtkGDCMImageWriter` `[friend]`

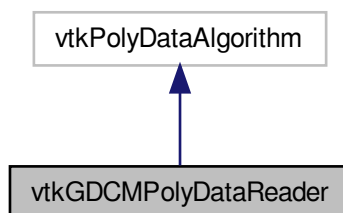
The documentation for this class was generated from the following file:

- [vtkGDCMMedicalImageProperties.h](#)

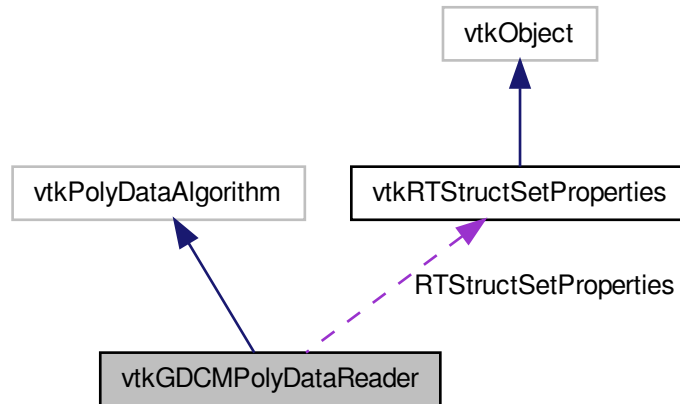
## 27.353 vtkGDCMPolyDataReader Class Reference

```
#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](#))
- [vtkTypeRevisionMacro](#) ([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](#)

### 27.353.1 Detailed Description

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 27.353.2 Constructor & Destructor Documentation

27.353.2.1 `vtkGDCMPolyDataReader::vtkGDCMPolyDataReader ( )` [protected]

27.353.2.2 `vtkGDCMPolyDataReader::~~vtkGDCMPolyDataReader ( )` [protected]

### 27.353.3 Member Function Documentation

27.353.3.1 `void vtkGDCMPolyDataReader::FillMedicalImageInformation ( const gdcm::Reader & reader )` [protected]

27.353.3.2 `static vtkGDCMPolyDataReader* vtkGDCMPolyDataReader::New ( )` [static]

Examples:

[gdcmscene.cxx](#), [GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.353.3.3 `virtual void vtkGDCMPolyDataReader::PrintSelf ( ostream & os, vtkIndent indent )` [virtual]

27.353.3.4 `int vtkGDCMPolyDataReader::RequestData ( vtkInformation *, vtkInformationVector **, vtkInformationVector * )` [protected]

27.353.3.5 `int vtkGDCMPolyDataReader::RequestData_HemodynamicWaveformStorage ( gdcm::Reader const & reader, vtkInformationVector * outputVector )` [protected]

27.353.3.6 `int vtkGDCMPolyDataReader::RequestData_RTStructureSetStorage ( gdcm::Reader const & reader, vtkInformationVector * outputVector )` [protected]

27.353.3.7 `int vtkGDCMPolyDataReader::RequestInformation ( vtkInformation * vtkNotUsedrequest, vtkInformationVector **, vtkNotUsedinputVector, vtkInformationVector * outputVector )` [protected]

27.353.3.8 `int vtkGDCMPolyDataReader::RequestInformation_HemodynamicWaveformStorage ( gdcm::Reader const & reader )` [protected]

27.353.3.9 `int vtkGDCMPolyDataReader::RequestInformation_RTStructureSetStorage ( gdcm::Reader const & reader )` [protected]

27.353.3.10 `vtkGDCMPolyDataReader::vtkGetObjectMacro ( MedicalImageProperties , vtkMedicalImageProperties )`

27.353.3.11 `vtkGDCMPolyDataReader::vtkGetObjectMacro ( RTStructSetProperties , vtkRTStructSetProperties )`

27.353.3.12 `vtkGDCMPolyDataReader::vtkGetStringMacro ( FileName )`

27.353.3.13 `vtkGDCMPolyDataReader::vtkSetStringMacro ( FileName )`

27.353.3.14 `vtkGDCMPolyDataReader::vtkTypeRevisionMacro ( vtkGDCMPolyDataReader , vtkPolyDataAlgorithm )`

## 27.353.4 Member Data Documentation

27.353.4.1 `char* vtkGDCMPolyDataReader::FileName` [protected]

27.353.4.2 `vtkMedicalImageProperties* vtkGDCMPolyDataReader::MedicalImageProperties` [protected]

27.353.4.3 `vtkRTStructSetProperties* vtkGDCMPolyDataReader::RTStructSetProperties` [protected]

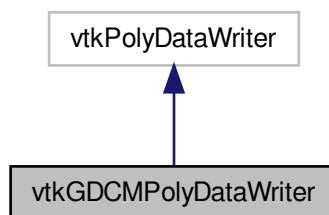
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataReader.h](#)

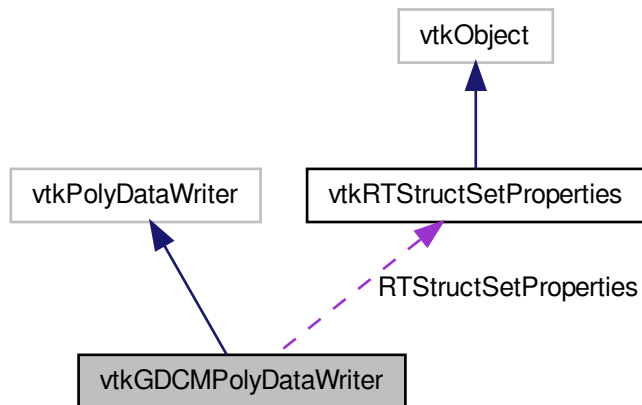
## 27.354 vtkGDCMPolyDataWriter Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (vtkGDCMPolyDataWriter, vtkPolyDataWriter)

### Static Public Member Functions

- static [vtkGDCMPolyDataWriter \\* New](#) ()

### Protected Member Functions

- [vtkGDCMPolyDataWriter](#) ()
- [~vtkGDCMPolyDataWriter](#) ()
- void [WriteData](#) ()
- void [WriteRTSTRUCTData](#) (gdcmm::File &file, int num)
- void [WriteRTSTRUCTInfo](#) (gdcmm::File &file)

### Protected Attributes

- vtkMedicalImageProperties \* [MedicalImageProperties](#)
- [vtkRTStructSetProperties](#) \* [RTStructSetProperties](#)



### 27.354.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

### 27.354.2 Constructor & Destructor Documentation

27.354.2.1 `vtkGDCMPolyDataWriter::vtkGDCMPolyDataWriter ( )` `[protected]`

27.354.2.2 `vtkGDCMPolyDataWriter::~~vtkGDCMPolyDataWriter ( )` `[protected]`

### 27.354.3 Member Function Documentation

27.354.3.1 `void vtkGDCMPolyDataWriter::InitializeRTStructSet ( vtkStdString inDirectory, vtkStdString inStructLabel, vtkStdString inStructName, vtkStringArray * inROINames, vtkStringArray * inROIAlgorithmName, vtkStringArray * inROIType )`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.354.3.2 `static vtkGDCMPolyDataWriter* vtkGDCMPolyDataWriter::New ( )` `[static]`

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.354.3.3 `virtual void vtkGDCMPolyDataWriter::PrintSelf ( ostream & os, vtkIndent indent )` `[virtual]`

27.354.3.4 `virtual void vtkGDCMPolyDataWriter::SetMedicalImageProperties ( vtkMedicalImageProperties * pd )` `[virtual]`

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.354.3.5 `void vtkGDCMPolyDataWriter::SetNumberOfInputPorts ( int n )`

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.354.3.6 `virtual void vtkGDCMPolyDataWriter::SetRTStructSetProperties ( vtkRTStructSetProperties * pd )` `[virtual]`

Examples:

[GenerateRTSTRUCT.cxx](#), and [rtstructapp.cxx](#).

27.354.3.7 `vtkGDCMPolyDataWriter::vtkTypeRevisionMacro ( vtkGDCMPolyDataWriter , vtkPolyDataWriter )`

27.354.3.8 `void vtkGDCMPolyDataWriter::WriteData ( )` [protected]

27.354.3.9 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTData ( gdcm::File & file, int num )` [protected]

27.354.3.10 `void vtkGDCMPolyDataWriter::WriteRTSTRUCTInfo ( gdcm::File & file )` [protected]

## 27.354.4 Member Data Documentation

27.354.4.1 `vtkMedicalImageProperties* vtkGDCMPolyDataWriter::MedicalImageProperties` [protected]

27.354.4.2 `vtkRTStructSetProperties* vtkGDCMPolyDataWriter::RTStructSetProperties` [protected]

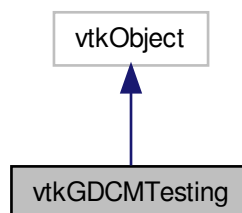
The documentation for this class was generated from the following file:

- [vtkGDCMPolyDataWriter.h](#)

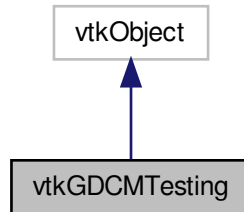
## 27.355 vtkGDCMTesting Class Reference

```
#include <vtkGDCMTesting.h>
```

Inheritance diagram for vtkGDCMTesting:



Collaboration diagram for vtkGDCMTesting:



## Public Types

- typedef const char \*const (\* [MD5MetalmagesType](#) )[3]

## Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) (vtkGDCMTesting, vtkObject)

## Static Public Member Functions

- static const char \* [GetGDCMDataRoot](#) ()
- static const char \*const \* [GetMD5Metalmage](#) (unsigned int file)
- static const char \* [GetMHDMD5FromFile](#) (const char \*filepath)
- static unsigned int [GetNumberOfMD5Metalmages](#) ()
- static const char \* [GetRAWMD5FromFile](#) (const char \*filepath)
- static const char \* [GetVTKDataRoot](#) ()
- static [vtkGDCMTesting](#) \* [New](#) ()

## Protected Member Functions

- [vtkGDCMTesting](#) ()
- [~vtkGDCMTesting](#) ()

## 27.355.1 Detailed Description

Examples:

[ReadSeriesIntoVTK.java](#), and [RefCounting.cs](#).

## 27.355.2 Member Typedef Documentation

27.355.2.1 `typedef const char* const(* vtkGDCMTesting::MD5MetalmagesType)[3]`

## 27.355.3 Constructor & Destructor Documentation

27.355.3.1 `vtkGDCMTesting::vtkGDCMTesting ( )` [protected]

27.355.3.2 `vtkGDCMTesting::~~vtkGDCMTesting ( )` [protected]

## 27.355.4 Member Function Documentation

27.355.4.1 `static const char* vtkGDCMTesting::GetGDCMDataRoot ( )` [static]

27.355.4.2 `static const char* const* vtkGDCMTesting::GetMD5Metalmage ( unsigned int file )` [static]

27.355.4.3 `static const char* vtkGDCMTesting::GetMHDMD5FromFile ( const char * filepath )` [static]

27.355.4.4 `static unsigned int vtkGDCMTesting::GetNumberOfMD5Metalmages ( )` [static]

27.355.4.5 `static const char* vtkGDCMTesting::GetRAWMD5FromFile ( const char * filepath )` [static]

27.355.4.6 `static const char* vtkGDCMTesting::GetVTKDataRoot ( )` [static]

27.355.4.7 `static vtkGDCMTesting* vtkGDCMTesting::New ( )` [static]

27.355.4.8 `void vtkGDCMTesting::PrintSelf ( ostream & os, vtkIndent indent )`

27.355.4.9 `vtkGDCMTesting::vtkTypeRevisionMacro ( vtkGDCMTesting , vtkObject )`

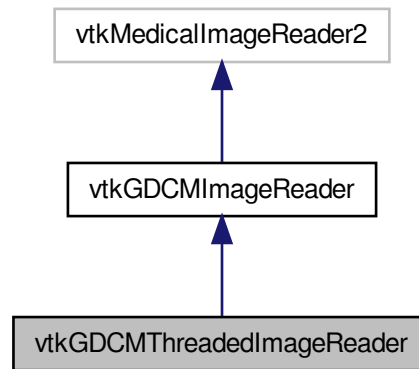
The documentation for this class was generated from the following file:

- [vtkGDCMTesting.h](#)

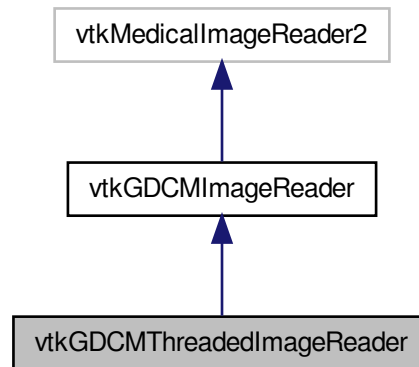
## 27.356 vtkGDCMThreadedImageReader Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (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

### 27.356.1 Constructor & Destructor Documentation

27.356.1.1 [vtkGDCMThreadedImageReader::vtkGDCMThreadedImageReader](#) ( ) [protected]

27.356.1.2 [vtkGDCMThreadedImageReader::~~vtkGDCMThreadedImageReader](#) ( ) [protected]

### 27.356.2 Member Function Documentation

27.356.2.1 void [vtkGDCMThreadedImageReader::ExecuteData](#) ( vtkDataObject \* *out* ) [protected]

27.356.2.2 void [vtkGDCMThreadedImageReader::ExecuteInformation](#) ( ) [protected]

27.356.2.3 static [vtkGDCMThreadedImageReader\\*](#) [vtkGDCMThreadedImageReader::New](#) ( ) [static]

27.356.2.4 virtual void [vtkGDCMThreadedImageReader::PrintSelf](#) ( ostream & *os*, vtkIndent *indent* ) [virtual]

Reimplemented from [vtkGDCMImageReader](#).

27.356.2.5 void [vtkGDCMThreadedImageReader::ReadFiles](#) ( unsigned int *nfiles*, const char \* *filenames*[] ) [protected]

27.356.2.6 void [vtkGDCMThreadedImageReader::RequestDataCompat](#) ( ) [protected]

27.356.2.7 [vtkGDCMThreadedImageReader::vtkBooleanMacro](#) ( UseShiftScale , int )

27.356.2.8 [vtkGDCMThreadedImageReader::vtkGetMacro](#) ( UseShiftScale , int )

27.356.2.9 [vtkGDCMThreadedImageReader::vtkSetMacro](#) ( Shift , double )

27.356.2.10 [vtkGDCMThreadedImageReader::vtkSetMacro](#) ( Scale , double )

27.356.2.11 [vtkGDCMThreadedImageReader::vtkSetMacro](#) ( UseShiftScale , int )

27.356.2.12 [vtkGDCMThreadedImageReader::vtkTypeRevisionMacro](#) ( [vtkGDCMThreadedImageReader](#) , [vtkGDCMImageReader](#) )

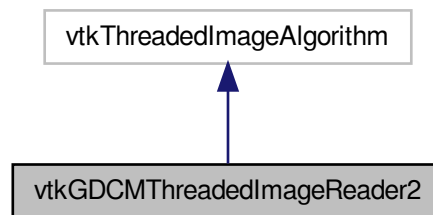
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader.h](#)

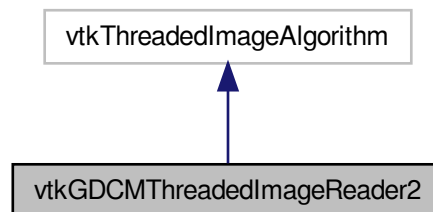
## 27.357 vtkGDCMThreadedImageReader2 Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) ([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)

## 27.357.1 Constructor & Destructor Documentation

27.357.1.1 [vtkGDCMThreadedImageReader2::vtkGDCMThreadedImageReader2](#) ( ) [protected]

27.357.1.2 [vtkGDCMThreadedImageReader2::~~vtkGDCMThreadedImageReader2](#) ( ) [protected]

## 27.357.2 Member Function Documentation

27.357.2.1 virtual const char\* [vtkGDCMThreadedImageReader2::GetFileName](#) ( int *i* = 0 ) [virtual]

27.357.2.2 static [vtkGDCMThreadedImageReader2](#)\* [vtkGDCMThreadedImageReader2::New](#) ( ) [static]



- 27.357.2.3 virtual void vtkGDCMThreadedImageReader2::PrintSelf ( ostream & *os*, vtkIndent *indent* ) [virtual]
- 27.357.2.4 int vtkGDCMThreadedImageReader2::RequestInformation ( vtkInformation \* *request*, vtkInformationVector \*\* *inputVector*, vtkInformationVector \* *outputVector* ) [protected]
- 27.357.2.5 virtual void vtkGDCMThreadedImageReader2::SetFileName ( const char \* *filename* ) [virtual]
- 27.357.2.6 virtual void vtkGDCMThreadedImageReader2::SetFileNames ( vtkStringArray \* ) [virtual]
- 27.357.2.7 int vtkGDCMThreadedImageReader2::SplitExtent ( int *splitExt[6]*, int *startExt[6]*, int *num*, int *total* )
- 27.357.2.8 void vtkGDCMThreadedImageReader2::ThreadedRequestData ( vtkInformation \* *request*, vtkInformationVector \*\* *inputVector*, vtkInformationVector \* *outputVector*, vtkImageData \*\*\* *inData*, vtkImageData \*\* *outData*, int *outExt[6]*, int *id* ) [protected]
- 27.357.2.9 vtkGDCMThreadedImageReader2::vtkBooleanMacro ( FileLowerLeft , int )
- 27.357.2.10 vtkGDCMThreadedImageReader2::vtkBooleanMacro ( LoadOverlays , int )
- 27.357.2.11 vtkGDCMThreadedImageReader2::vtkBooleanMacro ( UseShiftScale , int )
- 27.357.2.12 vtkGDCMThreadedImageReader2::vtkGetMacro ( FileLowerLeft , int )
- 27.357.2.13 vtkGDCMThreadedImageReader2::vtkGetMacro ( NumberOfOverlays , int )
- 27.357.2.14 vtkGDCMThreadedImageReader2::vtkGetMacro ( DataScalarType , int )
- 27.357.2.15 vtkGDCMThreadedImageReader2::vtkGetMacro ( NumberOfScalarComponents , int )
- 27.357.2.16 vtkGDCMThreadedImageReader2::vtkGetMacro ( LoadOverlays , int )
- 27.357.2.17 vtkGDCMThreadedImageReader2::vtkGetMacro ( Shift , double )
- 27.357.2.18 vtkGDCMThreadedImageReader2::vtkGetMacro ( Scale , double )
- 27.357.2.19 vtkGDCMThreadedImageReader2::vtkGetMacro ( UseShiftScale , int )
- 27.357.2.20 vtkGDCMThreadedImageReader2::vtkGetObjectMacro ( FileNames , vtkStringArray )
- 27.357.2.21 vtkGDCMThreadedImageReader2::vtkGetVector3Macro ( DataOrigin , double )
- 27.357.2.22 vtkGDCMThreadedImageReader2::vtkGetVector3Macro ( DataSpacing , double )
- 27.357.2.23 vtkGDCMThreadedImageReader2::vtkGetVector6Macro ( DataExtent , int )
- 27.357.2.24 vtkGDCMThreadedImageReader2::vtkSetMacro ( FileLowerLeft , int )
- 27.357.2.25 vtkGDCMThreadedImageReader2::vtkSetMacro ( DataScalarType , int )
- 27.357.2.26 vtkGDCMThreadedImageReader2::vtkSetMacro ( NumberOfScalarComponents , int )
- 27.357.2.27 vtkGDCMThreadedImageReader2::vtkSetMacro ( LoadOverlays , int )

27.357.2.28 `vtkGDCMThreadedImageReader2::vtkSetMacro ( Shift , double )`

27.357.2.29 `vtkGDCMThreadedImageReader2::vtkSetMacro ( Scale , double )`

27.357.2.30 `vtkGDCMThreadedImageReader2::vtkSetMacro ( UseShiftScale , int )`

27.357.2.31 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro ( DataOrigin , double )`

27.357.2.32 `vtkGDCMThreadedImageReader2::vtkSetVector3Macro ( DataSpacing , double )`

27.357.2.33 `vtkGDCMThreadedImageReader2::vtkSetVector6Macro ( DataExtent , int )`

27.357.2.34 `vtkGDCMThreadedImageReader2::vtkTypeRevisionMacro ( vtkGDCMThreadedImageReader2 ,  
vtkThreadedImageAlgorithm )`

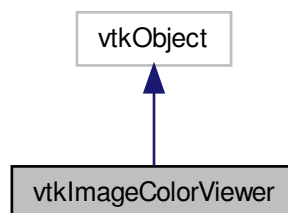
The documentation for this class was generated from the following file:

- [vtkGDCMThreadedImageReader2.h](#)

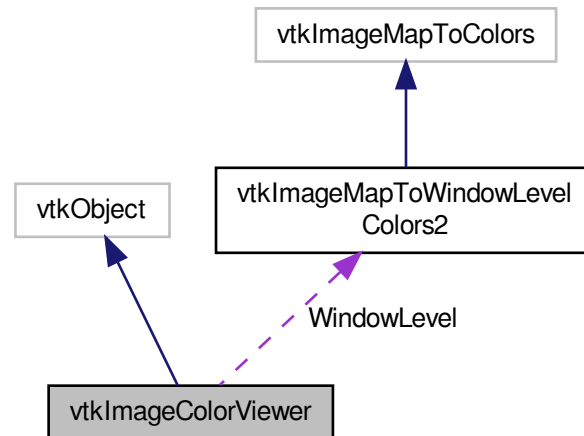
## 27.358 vtkImageColorViewer Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (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](#)

### 27.358.1 Detailed Description

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

### 27.358.2 Member Enumeration Documentation

27.358.2.1 anonymous enum

Enumerator

***SLICE\_ORIENTATION\_YZ***  
***SLICE\_ORIENTATION\_XZ***  
***SLICE\_ORIENTATION\_XY***

### 27.358.3 Constructor & Destructor Documentation

27.358.3.1 `vtkImageColorViewer::vtkImageColorViewer ( )` [protected]

27.358.3.2 `vtkImageColorViewer::~~vtkImageColorViewer ( )` [protected]

### 27.358.4 Member Function Documentation

27.358.4.1 `virtual void vtkImageColorViewer::AddInput ( vtkImageData * input )` [virtual]

27.358.4.2 `virtual void vtkImageColorViewer::AddInputConnection ( vtkAlgorithmOutput * input )` [virtual]

27.358.4.3 `virtual double vtkImageColorViewer::GetColorLevel ( )` [virtual]

27.358.4.4 `virtual double vtkImageColorViewer::GetColorWindow ( )` [virtual]

27.358.4.5 `virtual vtkImageData* vtkImageColorViewer::GetInput ( )` [virtual]

```

27.358.4.6  virtual int vtkImageColorViewer::GetOffScreenRendering ( ) [virtual]

27.358.4.7  double vtkImageColorViewer::GetOverlayVisibility ( )

27.358.4.8  virtual int* vtkImageColorViewer::GetPosition ( ) [virtual]

27.358.4.9  virtual int* vtkImageColorViewer::GetSize ( ) [virtual]

27.358.4.10 virtual int vtkImageColorViewer::GetSliceMax ( ) [virtual]

27.358.4.11 virtual int vtkImageColorViewer::GetSliceMin ( ) [virtual]

27.358.4.12 virtual void vtkImageColorViewer::GetSliceRange ( int range[2] ) [inline],[virtual]

27.358.4.13 virtual void vtkImageColorViewer::GetSliceRange ( int & min, int & max ) [virtual]

27.358.4.14 virtual int* vtkImageColorViewer::GetSliceRange ( ) [virtual]

27.358.4.15 virtual const char* vtkImageColorViewer::GetWindowName ( ) [virtual]

27.358.4.16 virtual void vtkImageColorViewer::InstallPipeline ( ) [protected],[virtual]

27.358.4.17 static vtkImageColorViewer* vtkImageColorViewer::New ( ) [static]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

```

27.358.4.18 void vtkImageColorViewer::PrintSelf ( ostream & os, vtkIndent indent )

27.358.4.19 virtual void vtkImageColorViewer::Render ( void ) [virtual]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

```

27.358.4.20 virtual void vtkImageColorViewer::SetColorLevel ( double s ) [virtual]

27.358.4.21 virtual void vtkImageColorViewer::SetColorWindow ( double s ) [virtual]

27.358.4.22 virtual void vtkImageColorViewer::SetDisplayId ( void * a ) [virtual]

27.358.4.23 virtual void vtkImageColorViewer::SetInput ( vtkImageData * in ) [virtual]

```

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

27.358.4.24 `virtual void vtkImageColorViewer::SetInputConnection ( vtkAlgorithmOutput * input )` [virtual]

27.358.4.25 `virtual void vtkImageColorViewer::SetOffScreenRendering ( int )` [virtual]

27.358.4.26 `void vtkImageColorViewer::SetOverlayVisibility ( double vis )`

27.358.4.27 `virtual void vtkImageColorViewer::SetParentId ( void * a )` [virtual]

27.358.4.28 `virtual void vtkImageColorViewer::SetPosition ( int a, int b )` [virtual]

27.358.4.29 `virtual void vtkImageColorViewer::SetPosition ( int a[2] )` [inline],[virtual]

References `SetPosition()`.

Referenced by `SetPosition()`.

27.358.4.30 `virtual void vtkImageColorViewer::SetRenderer ( vtkRenderer * arg )` [virtual]

27.358.4.31 `virtual void vtkImageColorViewer::SetRenderWindow ( vtkRenderWindow * arg )` [virtual]

27.358.4.32 `virtual void vtkImageColorViewer::SetSize ( int a, int b )` [virtual]

Examples:

[gdcmrtionplan.cxx](#), and [gdcmrtplan.cxx](#).

27.358.4.33 `virtual void vtkImageColorViewer::SetSize ( int a[2] )` [inline],[virtual]

References `SetSize()`.

Referenced by `SetSize()`.

27.358.4.34 `virtual void vtkImageColorViewer::SetSlice ( int s )` [virtual]

27.358.4.35 `virtual void vtkImageColorViewer::SetSliceOrientation ( int orientation )` [virtual]

27.358.4.36 `virtual void vtkImageColorViewer::SetSliceOrientationToXY ( )` [inline],[virtual]

References `SLICE_ORIENTATION_XY`.

27.358.4.37 `virtual void vtkImageColorViewer::SetSliceOrientationToXZ ( )` [inline],[virtual]

References `SLICE_ORIENTATION_XZ`.

27.358.4.38 `virtual void vtkImageColorViewer::SetSliceOrientationToYZ ( )` [inline],[virtual]

References `SLICE_ORIENTATION_YZ`.

27.358.4.39 `virtual void vtkImageColorViewer::SetupInteractor ( vtkRenderWindowInteractor * )` [virtual]

Examples:

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

27.358.4.40 `virtual void vtkImageColorViewer::SetWindowId ( void * a )` [virtual]

27.358.4.41 `virtual void vtkImageColorViewer::UnInstallPipeline ( )` [protected],[virtual]

27.358.4.42 `virtual void vtkImageColorViewer::UpdateDisplayExtent ( )` [virtual]

27.358.4.43 `virtual void vtkImageColorViewer::UpdateOrientation ( )` [protected],[virtual]

27.358.4.44 `vtkImageColorViewer::VTK_LEGACY ( int GetWholeZMin() )`

27.358.4.45 `vtkImageColorViewer::VTK_LEGACY ( int GetWholeZMax() )`

27.358.4.46 `vtkImageColorViewer::VTK_LEGACY ( int GetZSlice() )`

27.358.4.47 `vtkImageColorViewer::VTK_LEGACY ( void SetZSliceint )`

27.358.4.48 `vtkImageColorViewer::vtkBooleanMacro ( OffScreenRendering , int )`

27.358.4.49 `vtkImageColorViewer::vtkGetMacro ( SliceOrientation , int )`

27.358.4.50 `vtkImageColorViewer::vtkGetMacro ( Slice , int )`

27.358.4.51 `vtkImageColorViewer::vtkGetObjectMacro ( RenderWindow , vtkRenderWindow )`

27.358.4.52 `vtkImageColorViewer::vtkGetObjectMacro ( Renderer , vtkRenderer )`

27.358.4.53 `vtkImageColorViewer::vtkGetObjectMacro ( ImageActor , vtkImageActor )`

27.358.4.54 `vtkImageColorViewer::vtkGetObjectMacro ( WindowLevel , vtkImageMapToWindowLevelColors2 )`

27.358.4.55 `vtkImageColorViewer::vtkGetObjectMacro ( InteractorStyle , vtkInteractorStyleImage )`

27.358.4.56 `vtkImageColorViewer::vtkTypeRevisionMacro ( vtkImageColorViewer , vtkObject )`

## 27.358.5 Friends And Related Function Documentation

27.358.5.1 `friend class vtkImageColorViewerCallback` [friend]

## 27.358.6 Member Data Documentation

27.358.6.1 `int vtkImageColorViewer::FirstRender` [protected]

27.358.6.2 `vtkImageActor* vtkImageColorViewer::ImageActor` [protected]

27.358.6.3 `vtkRenderWindowInteractor* vtkImageColorViewer::Interactor` [protected]



27.358.6.4 `vtkInteractorStyleImage* vtkImageColorViewer::InteractorStyle` `[protected]`

27.358.6.5 `vtkImageActor* vtkImageColorViewer::OverlayImageActor` `[protected]`

27.358.6.6 `vtkRenderer* vtkImageColorViewer::Renderer` `[protected]`

27.358.6.7 `vtkRenderWindow* vtkImageColorViewer::RenderWindow` `[protected]`

27.358.6.8 `int vtkImageColorViewer::Slice` `[protected]`

27.358.6.9 `int vtkImageColorViewer::SliceOrientation` `[protected]`

27.358.6.10 `vtkImageMapToWindowLevelColors2* vtkImageColorViewer::WindowLevel` `[protected]`

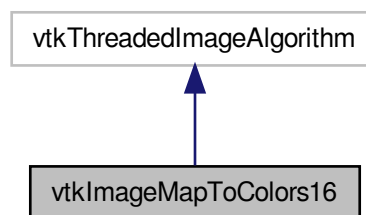
The documentation for this class was generated from the following file:

- [vtkImageColorViewer.h](#)

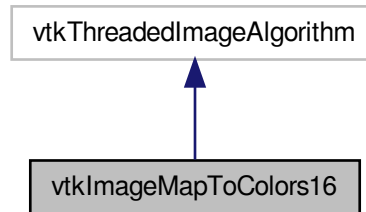
## 27.359 vtkImageMapToColors16 Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) (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](#)

## 27.359.1 Constructor & Destructor Documentation

27.359.1.1 `vtkImageMapToColors16::vtkImageMapToColors16 ( )` `[protected]`

27.359.1.2 `vtkImageMapToColors16::~~vtkImageMapToColors16 ( )` `[protected]`

## 27.359.2 Member Function Documentation

27.359.2.1 `virtual unsigned long vtkImageMapToColors16::GetMTime ( )` `[virtual]`

27.359.2.2 `static vtkImageMapToColors16* vtkImageMapToColors16::New ( )` `[static]`

27.359.2.3 `void vtkImageMapToColors16::PrintSelf ( ostream & os, vtkIndent indent )`

27.359.2.4 `virtual int vtkImageMapToColors16::RequestData ( vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector )` `[protected]`, `[virtual]`

27.359.2.5 `virtual int vtkImageMapToColors16::RequestInformation ( vtkInformation * , vtkInformationVector ** , vtkInformationVector * )` `[protected]`, `[virtual]`

27.359.2.6 `virtual void vtkImageMapToColors16::SetLookupTable ( vtkScalarsToColors * )` `[virtual]`

27.359.2.7 `void vtkImageMapToColors16::SetOutputFormatToLuminance ( )` `[inline]`

27.359.2.8 `void vtkImageMapToColors16::SetOutputFormatToLuminanceAlpha ( )` `[inline]`

27.359.2.9 `void vtkImageMapToColors16::SetOutputFormatToRGB ( )` `[inline]`

27.359.2.10 `void vtkImageMapToColors16::SetOutputFormatToRGBA ( )` `[inline]`

27.359.2.11 `void vtkImageMapToColors16::ThreadedRequestData ( vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id )` `[protected]`

27.359.2.12 `vtkImageMapToColors16::vtkBooleanMacro ( PassAlphaToOutput , int )`

27.359.2.13 `vtkImageMapToColors16::vtkGetMacro ( OutputFormat , int )`

27.359.2.14 `vtkImageMapToColors16::vtkGetMacro ( ActiveComponent , int )`

27.359.2.15 `vtkImageMapToColors16::vtkGetMacro ( PassAlphaToOutput , int )`

27.359.2.16 `vtkImageMapToColors16::vtkGetObjectMacro ( LookupTable , vtkScalarsToColors )`

27.359.2.17 `vtkImageMapToColors16::vtkSetMacro ( OutputFormat , int )`

27.359.2.18 `vtkImageMapToColors16::vtkSetMacro ( ActiveComponent , int )`

27.359.2.19 `vtkImageMapToColors16::vtkSetMacro ( PassAlphaToOutput , int )`

27.359.2.20 `vtkImageMapToColors16::vtkTypeRevisionMacro ( vtkImageMapToColors16 , vtkThreadedImageAlgorithm )`

### 27.359.3 Member Data Documentation

27.359.3.1 `int vtkImageMapToColors16::ActiveComponent` [protected]

27.359.3.2 `int vtkImageMapToColors16::DataWasPassed` [protected]

27.359.3.3 `vtkScalarsToColors* vtkImageMapToColors16::LookupTable` [protected]

27.359.3.4 `int vtkImageMapToColors16::OutputFormat` [protected]

27.359.3.5 `int vtkImageMapToColors16::PassAlphaToOutput` [protected]

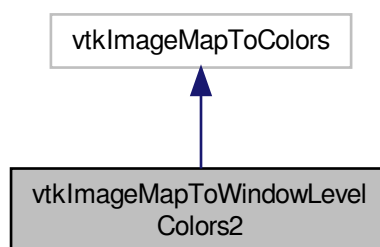
The documentation for this class was generated from the following file:

- [vtkImageMapToColors16.h](#)

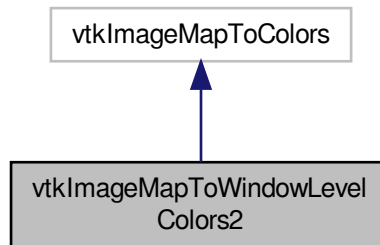
## 27.360 vtkImageMapToWindowLevelColors2 Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) ([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](#)

### 27.360.1 Constructor & Destructor Documentation

27.360.1.1 `vtkImageMapToWindowLevelColors2::vtkImageMapToWindowLevelColors2 ( )` [protected]

27.360.1.2 `vtkImageMapToWindowLevelColors2::~~vtkImageMapToWindowLevelColors2 ( )` [protected]

### 27.360.2 Member Function Documentation

27.360.2.1 `static vtkImageMapToWindowLevelColors2* vtkImageMapToWindowLevelColors2::New ( )` [static]

27.360.2.2 `void vtkImageMapToWindowLevelColors2::PrintSelf ( ostream & os, vtkIndent indent )`

27.360.2.3 `virtual int vtkImageMapToWindowLevelColors2::RequestData ( vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector )` [protected], [virtual]

27.360.2.4 `virtual int vtkImageMapToWindowLevelColors2::RequestInformation ( vtkInformation * , vtkInformationVector ** , vtkInformationVector * )` [protected], [virtual]

27.360.2.5 `void vtkImageMapToWindowLevelColors2::ThreadedRequestData ( vtkInformation * request, vtkInformationVector ** inputVector, vtkInformationVector * outputVector, vtkImageData *** inData, vtkImageData ** outData, int extent[6], int id )` [protected]

27.360.2.6 `vtkImageMapToWindowLevelColors2::vtkGetMacro ( Window , double )`

27.360.2.7 `vtkImageMapToWindowLevelColors2::vtkGetMacro ( Level , double )`

27.360.2.8 `vtkImageMapToWindowLevelColors2::vtkSetMacro ( Window , double )`

27.360.2.9 `vtkImageMapToWindowLevelColors2::vtkSetMacro ( Level , double )`

27.360.2.10 `vtkImageMapToWindowLevelColors2::vtkTypeRevisionMacro ( vtkImageMapToWindowLevelColors2 , vtkImageMapToColors )`

### 27.360.3 Member Data Documentation

27.360.3.1 `double vtkImageMapToWindowLevelColors2::Level` [protected]

27.360.3.2 `double vtkImageMapToWindowLevelColors2::Window` [protected]

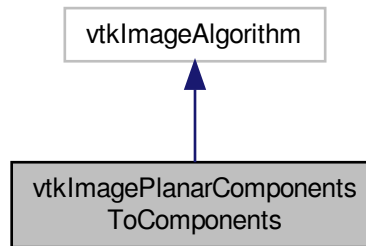
The documentation for this class was generated from the following file:

- [vtkImageMapToWindowLevelColors2.h](#)

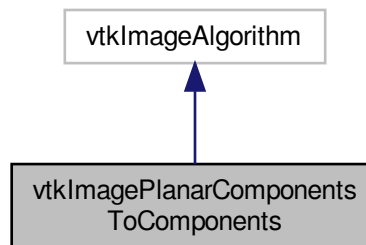
## 27.361 vtkImagePlanarComponentsToComponents Class Reference

```
#include <vtkImagePlanarComponentsToComponents.h>
```

Inheritance diagram for vtkImagePlanarComponentsToComponents:



Collaboration diagram for vtkImagePlanarComponentsToComponents:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([vtkImagePlanarComponentsToComponents](#), vtkImageAlgorithm)

### Static Public Member Functions

- static  
[vtkImagePlanarComponentsToComponents](#) \* [New](#) ()

### Protected Member Functions

- [vtkImagePlanarComponentsToComponents](#) ()
- [~vtkImagePlanarComponentsToComponents](#) ()
- virtual int [RequestData](#) (vtkInformation \*, vtkInformationVector \*\*, vtkInformationVector \*)

## 27.361.1 Constructor & Destructor Documentation

27.361.1.1 `vtkImagePlanarComponentsToComponents::vtkImagePlanarComponentsToComponents ( )` `[protected]`

27.361.1.2 `vtkImagePlanarComponentsToComponents::~~vtkImagePlanarComponentsToComponents ( )` `[inline]`, `[protected]`

## 27.361.2 Member Function Documentation

27.361.2.1 `static vtkImagePlanarComponentsToComponents* vtkImagePlanarComponentsToComponents::New ( )` `[static]`

27.361.2.2 `void vtkImagePlanarComponentsToComponents::PrintSelf ( ostream & os, vtkIndent indent )`

27.361.2.3 `virtual int vtkImagePlanarComponentsToComponents::RequestData ( vtkInformation *, vtkInformationVector **, vtkInformationVector * )` `[protected]`, `[virtual]`

27.361.2.4 `vtkImagePlanarComponentsToComponents::vtkTypeRevisionMacro ( vtkImagePlanarComponentsToComponents, vtkImageAlgorithm )`

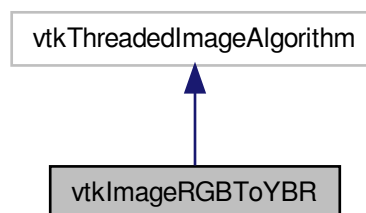
The documentation for this class was generated from the following file:

- [vtkImagePlanarComponentsToComponents.h](#)

## 27.362 vtkImageRGBToYBR Class Reference

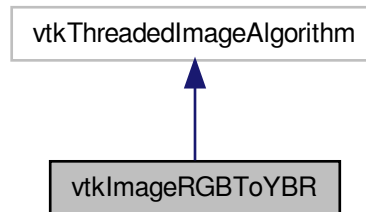
```
#include <vtkImageRGBToYBR.h>
```

Inheritance diagram for `vtkImageRGBToYBR`:





Collaboration diagram for vtkImageRGBToYBR:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) (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)

## 27.362.1 Constructor & Destructor Documentation

27.362.1.1 `vtkImageRGBToYBR::vtkImageRGBToYBR ( )` [protected]

27.362.1.2 `vtkImageRGBToYBR::~~vtkImageRGBToYBR ( )` [inline], [protected]

## 27.362.2 Member Function Documentation

27.362.2.1 `static vtkImageRGBToYBR* vtkImageRGBToYBR::New ( )` [static]

27.362.2.2 `void vtkImageRGBToYBR::PrintSelf ( ostream & os, vtkIndent indent )`

27.362.2.3 `void vtkImageRGBToYBR::ThreadedExecute ( vtkImageData * inData, vtkImageData * outData, int ext[6], int id )`  
[protected]

27.362.2.4 `vtkImageRGBToYBR::vtkTypeRevisionMacro ( vtkImageRGBToYBR , vtkThreadedImageAlgorithm )`

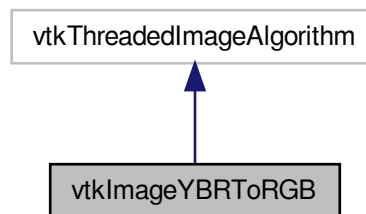
The documentation for this class was generated from the following file:

- [vtkImageRGBToYBR.h](#)

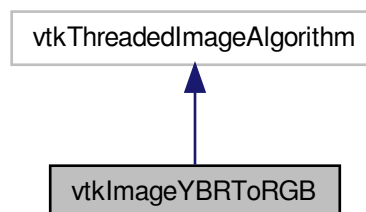
## 27.363 vtkImageYBRToRGB Class Reference

```
#include <vtkImageYBRToRGB.h>
```

Inheritance diagram for vtkImageYBRToRGB:



Collaboration diagram for vtkImageYBRToRGB:



### Public Member Functions

- void [PrintSelf](#) (ostream &os, vtkIndent indent)
- [vtkTypeRevisionMacro](#) ([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)

## 27.363.1 Constructor & Destructor Documentation

27.363.1.1 `vtkImageYBRToRGB::vtkImageYBRToRGB ( )` [protected]

27.363.1.2 `vtkImageYBRToRGB::~~vtkImageYBRToRGB ( )` [inline], [protected]

## 27.363.2 Member Function Documentation

27.363.2.1 `static vtkImageYBRToRGB* vtkImageYBRToRGB::New ( )` [static]

27.363.2.2 `void vtkImageYBRToRGB::PrintSelf ( ostream & os, vtkIndent indent )`

27.363.2.3 `void vtkImageYBRToRGB::ThreadedExecute ( vtkImageData * inData, vtkImageData * outData, int ext[6], int id )`  
[protected]

27.363.2.4 `vtkImageYBRToRGB::vtkTypeRevisionMacro ( vtkImageYBRToRGB , vtkThreadedImageAlgorithm )`

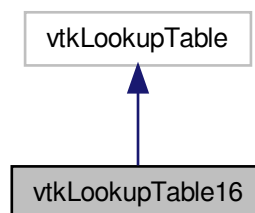
The documentation for this class was generated from the following file:

- [vtkImageYBRToRGB.h](#)

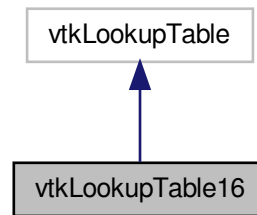
## 27.364 vtkLookupTable16 Class Reference

```
#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)
- [vtkTypeRevisionMacro](#) ([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](#)

## 27.364.1 Constructor & Destructor Documentation

27.364.1.1 `vtkLookupTable16::vtkLookupTable16 ( int size = 256, int ext = 256 )` [protected]

27.364.1.2 `vtkLookupTable16::~~vtkLookupTable16 ( )` [protected]

## 27.364.2 Member Function Documentation

- 27.364.2.1 void vtkLookupTable16::Build ( )
- 27.364.2.2 unsigned short\* vtkLookupTable16::GetPointer ( const vtkIdType *id* ) [inline]
- 27.364.2.3 void vtkLookupTable16::MapScalarsThroughTable2 ( void \* *input*, unsigned char \* *output*, int *inputDataType*, int *numberOfValues*, int *inputIncrement*, int *outputFormat* ) [protected]
- 27.364.2.4 static vtkLookupTable16\* vtkLookupTable16::New ( ) [static]
- 27.364.2.5 void vtkLookupTable16::PrintSelf ( ostream & *os*, vtkIndent *indent* )
- 27.364.2.6 void vtkLookupTable16::SetNumberOfTableValues ( vtkIdType *number* )
- 27.364.2.7 vtkLookupTable16::vtkTypeRevisionMacro ( vtkLookupTable16 , vtkLookupTable )
- 27.364.2.8 unsigned char \* vtkLookupTable16::WritePointer ( const vtkIdType *id*, const int *number* ) [inline]

References Table16.

### 27.364.3 Member Data Documentation

- 27.364.3.1 vtkUnsignedShortArray\* vtkLookupTable16::Table16 [protected]

Referenced by WritePointer().

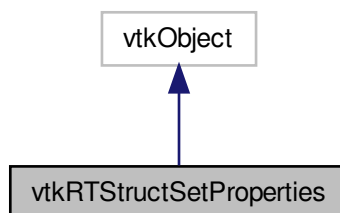
The documentation for this class was generated from the following file:

- [vtkLookupTable16.h](#)

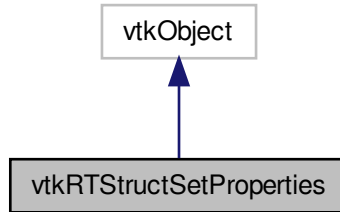
## 27.365 vtkRTStructSetProperties Class Reference

```
#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 \*refframerefid, 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 \(ReferenceFrameOfReferenceUID\)](#)
- [vtkTypeRevisionMacro \(vtkRTStructSetProperties, vtkObject\)](#)

### 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](#)

## 27.365.1 Detailed Description

Examples:

[GenerateRTSTRUCT.cxx](#).

## 27.365.2 Constructor & Destructor Documentation

27.365.2.1 [vtkRTStructSetProperties::vtkRTStructSetProperties \( \)](#) [protected]

27.365.2.2 [vtkRTStructSetProperties::~~vtkRTStructSetProperties \( \)](#) [protected]

## 27.365.3 Member Function Documentation

- 27.365.3.1 void vtkRTStructSetProperties::AddContourReferencedFrameOfReference ( vtkIdType *pdnum*, const char \* *classuid*, const char \* *instanceuid* )
- 27.365.3.2 void vtkRTStructSetProperties::AddReferencedFrameOfReference ( const char \* *classuid*, const char \* *instanceuid* )
- 27.365.3.3 void vtkRTStructSetProperties::AddStructureSetROI ( int *roinumber*, const char \* *refframerefid*, const char \* *roiname*, const char \* *ROIGenerationAlgorithm*, const char \* *ROIDescription* = 0 )
- 27.365.3.4 void vtkRTStructSetProperties::AddStructureSetROIObservation ( int *refnumber*, int *observationnumber*, const char \* *rtroiinterpretedtype*, const char \* *roiinterpreter*, const char \* *roiobservationlabel* = 0 )
- 27.365.3.5 virtual void vtkRTStructSetProperties::Clear ( ) [virtual]
- 27.365.3.6 virtual void vtkRTStructSetProperties::DeepCopy ( vtkRTStructSetProperties \* *p* ) [virtual]
- 27.365.3.7 const char\* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceClassUID ( vtkIdType *pdnum*, vtkIdType *id* )
- 27.365.3.8 const char\* vtkRTStructSetProperties::GetContourReferencedFrameOfReferenceInstanceUID ( vtkIdType *pdnum*, vtkIdType *id* )
- 27.365.3.9 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( )
- 27.365.3.10 vtkIdType vtkRTStructSetProperties::GetNumberOfContourReferencedFrameOfReferences ( vtkIdType *pdnum* )
- 27.365.3.11 vtkIdType vtkRTStructSetProperties::GetNumberOfReferencedFrameOfReferences ( )
- 27.365.3.12 vtkIdType vtkRTStructSetProperties::GetNumberOfStructureSetROIs ( )
- 27.365.3.13 const char\* vtkRTStructSetProperties::GetReferencedFrameOfReferenceClassUID ( vtkIdType *id* )
- 27.365.3.14 const char\* vtkRTStructSetProperties::GetReferencedFrameOfReferenceInstanceUID ( vtkIdType *id* )
- 27.365.3.15 int vtkRTStructSetProperties::GetStructureSetObservationNumber ( vtkIdType *id* )
- 27.365.3.16 const char\* vtkRTStructSetProperties::GetStructureSetROIDescription ( vtkIdType *id* )
- 27.365.3.17 const char\* vtkRTStructSetProperties::GetStructureSetROIGenerationAlgorithm ( vtkIdType )
- 27.365.3.18 const char\* vtkRTStructSetProperties::GetStructureSetROIName ( vtkIdType )
- 27.365.3.19 int vtkRTStructSetProperties::GetStructureSetROINumber ( vtkIdType *id* )
- 27.365.3.20 const char\* vtkRTStructSetProperties::GetStructureSetROIObservationLabel ( vtkIdType *id* )
- 27.365.3.21 const char\* vtkRTStructSetProperties::GetStructureSetROIRefFrameRefUID ( vtkIdType )
- 27.365.3.22 const char\* vtkRTStructSetProperties::GetStructureSetRTROIInterpretedType ( vtkIdType *id* )



27.365.3.23 `static vtkRTStructSetProperties* vtkRTStructSetProperties::New ( )` `[static]`

Examples:

[GenerateRTSTRUCT.cxx](#).

27.365.3.24 `void vtkRTStructSetProperties::PrintSelf ( ostream & os, vtkIndent indent )`

27.365.3.25 `vtkRTStructSetProperties::vtkGetStringMacro ( StructureSetLabel )`

27.365.3.26 `vtkRTStructSetProperties::vtkGetStringMacro ( StructureSetName )`

27.365.3.27 `vtkRTStructSetProperties::vtkGetStringMacro ( StructureSetDate )`

27.365.3.28 `vtkRTStructSetProperties::vtkGetStringMacro ( StructureSetTime )`

27.365.3.29 `vtkRTStructSetProperties::vtkGetStringMacro ( SOPInstanceUID )`

27.365.3.30 `vtkRTStructSetProperties::vtkGetStringMacro ( StudyInstanceUID )`

27.365.3.31 `vtkRTStructSetProperties::vtkGetStringMacro ( SeriesInstanceUID )`

27.365.3.32 `vtkRTStructSetProperties::vtkGetStringMacro ( ReferenceSeriesInstanceUID )`

27.365.3.33 `vtkRTStructSetProperties::vtkGetStringMacro ( ReferenceFrameOfReferenceUID )`

27.365.3.34 `vtkRTStructSetProperties::vtkSetStringMacro ( StructureSetLabel )`

27.365.3.35 `vtkRTStructSetProperties::vtkSetStringMacro ( StructureSetName )`

27.365.3.36 `vtkRTStructSetProperties::vtkSetStringMacro ( StructureSetDate )`

27.365.3.37 `vtkRTStructSetProperties::vtkSetStringMacro ( StructureSetTime )`

27.365.3.38 `vtkRTStructSetProperties::vtkSetStringMacro ( SOPInstanceUID )`

27.365.3.39 `vtkRTStructSetProperties::vtkSetStringMacro ( StudyInstanceUID )`

27.365.3.40 `vtkRTStructSetProperties::vtkSetStringMacro ( SeriesInstanceUID )`

27.365.3.41 `vtkRTStructSetProperties::vtkSetStringMacro ( ReferenceSeriesInstanceUID )`

27.365.3.42 `vtkRTStructSetProperties::vtkSetStringMacro ( ReferenceFrameOfReferenceUID )`

27.365.3.43 `vtkRTStructSetProperties::vtkTypeRevisionMacro ( vtkRTStructSetProperties , vtkObject )`

## 27.365.4 Member Data Documentation

27.365.4.1 `vtkRTStructSetPropertiesInternals* vtkRTStructSetProperties::Internals` `[protected]`

27.365.4.2 `char* vtkRTStructSetProperties::ReferenceFrameOfReferenceUID` `[protected]`

27.365.4.3 `char* vtkRTStructSetProperties::ReferenceSeriesInstanceUID` [protected]

27.365.4.4 `char* vtkRTStructSetProperties::SeriesInstanceUID` [protected]

27.365.4.5 `char* vtkRTStructSetProperties::SOPInstanceUID` [protected]

27.365.4.6 `char* vtkRTStructSetProperties::StructureSetDate` [protected]

27.365.4.7 `char* vtkRTStructSetProperties::StructureSetLabel` [protected]

27.365.4.8 `char* vtkRTStructSetProperties::StructureSetName` [protected]

27.365.4.9 `char* vtkRTStructSetProperties::StructureSetTime` [protected]

27.365.4.10 `char* vtkRTStructSetProperties::StudyInstanceUID` [protected]

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

- [vtkRTStructSetProperties.h](#)

## 27.366 gdcm::Waveform Class Reference

[Waveform](#) class.

```
#include <gdcmWaveform.h>
```

### Public Member Functions

- [Waveform](#) ()

### 27.366.1 Detailed Description

[Waveform](#) class.

### 27.366.2 Constructor & Destructor Documentation

27.366.2.1 `gdcm::Waveform::Waveform ( )` [inline]

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

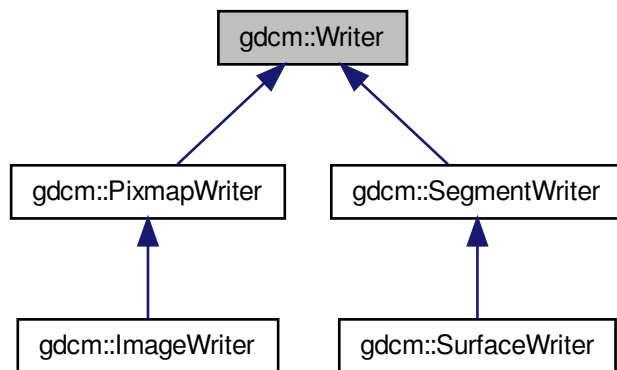
- [gdcmWaveform.h](#)

## 27.367 gdcm::Writer Class Reference

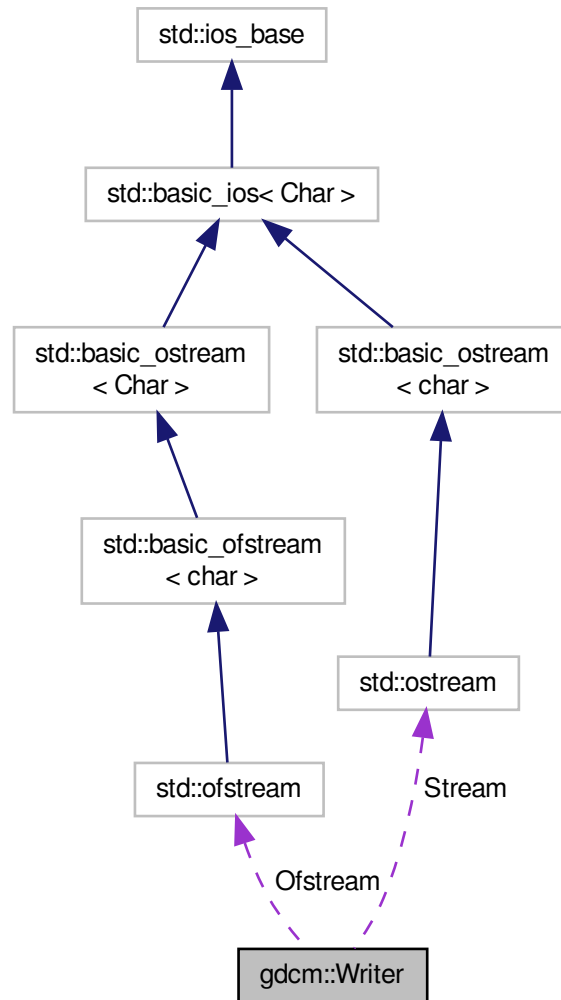
[Writer](#) ala DOM (Document [Object](#) Model) This class is a non-validating writer, it will only performs well- formedness check only.

```
#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

- std::ostream \* [GetStreamPtr](#) () const
- void [SetWriteDataSetOnly](#) (bool b)

### Protected Attributes

- std::ofstream \* [Ofstream](#)
- std::ostream \* [Stream](#)

### Friends

- class [StreamImageWriter](#)

## 27.367.1 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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [Extracting\\_All\\_Resolution.cxx](#), [Fake\\_Image\\_Using\\_Stream\\_Image\\_Writer.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAllVR.cxx](#), [GenerateDICOMDIR.cs](#), [GenFakeIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [NewSequence.cs](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), [QIDO-RS.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

## 27.367.2 Constructor & Destructor Documentation

27.367.2.1 `gdcm::Writer::Writer ( )`

27.367.2.2 `virtual gdcm::Writer::~~Writer ( )` `[virtual]`

## 27.367.3 Member Function Documentation

27.367.3.1 `void gdcm::Writer::CheckFileMetaInformationOff ( )` `[inline]`

Examples:

[FixBrokenJ2K.cxx](#), and [HelloWorld.cxx](#).

27.367.3.2 `void gdcm::Writer::CheckFileMetaInformationOn ( )` `[inline]`

27.367.3.3 `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](#), [iU22tomultisc.cxx](#), [pmsct\\_rgb1.cxx](#), [rle2img.cxx](#), and [StreamImageReaderTest.cxx](#).

27.367.3.4 `std::ostream* gdcm::Writer::GetStreamPtr ( ) const` `[inline]`, `[protected]`

27.367.3.5 `void gdcm::Writer::SetCheckFileMetaInformation ( bool b )` `[inline]`

Undocumented function, do not use (= leave default)

Examples:

[GenAllVR.cxx](#), [GenFakeIdentifyFile.cxx](#), and [PatchFile.cxx](#).

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

Set/Get the DICOM file ([DataSet](#) + Header)

## Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [DuplicatePCDE.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenFakelImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**27.367.3.7** void `gdcm::Writer::SetFileName ( const char * filename_native )`

Set the filename of DICOM file to write:

## Examples:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CompressImage.cxx](#), [CreateARGBImage.cxx](#), [CreateCMYKImage.cxx](#), [CreateJPIPDataSet.cxx](#), [csa2img.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAIIVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenFakelImage.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloVizWorld.cxx](#), [HelloWorld.cxx](#), [iU22tomultisc.cxx](#), [LargeVRDSExplicit.cxx](#), [MergeTwoFiles.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**27.367.3.8** void `gdcm::Writer::SetStream ( std::ostream & output_stream )` [inline]

Set user ostream buffer.

**27.367.3.9** void `gdcm::Writer::SetWriteDataSetOnly ( bool b )` [inline],[protected]

**27.367.3.10** 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:

[ChangeSequenceUltrasound.cxx](#), [ClinicalTrialAnnotate.cxx](#), [CreateJPIPDataSet.cxx](#), [DuplicatePCDE.cxx](#), [EncapsulateFileInRawData.cxx](#), [FixBrokenJ2K.cxx](#), [FixJAIBugJPEGLS.cxx](#), [GenAIIVR.cxx](#), [GenFakelIdentifyFile.cxx](#), [GenLongSeqs.cxx](#), [GenSeqs.cxx](#), [HelloWorld.cxx](#), [LargeVRDSExplicit.cxx](#), [PatchFile.cxx](#), [pmsct\\_rgb1.cxx](#), and [rle2img.cxx](#).

**27.367.4 Friends And Related Function Documentation**

**27.367.4.1** friend class `StreamImageWriter` [friend]

**27.367.5 Member Data Documentation**

**27.367.5.1** std::ostream\* `gdcm::Writer::Ofstream` [protected]

**27.367.5.2** std::ostream\* `gdcm::Writer::Stream` [protected]

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

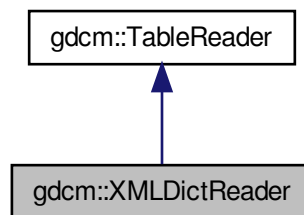
- [gdcmWriter.h](#)

## 27.368 gdcm::XMLDictReader Class Reference

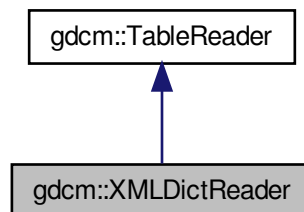
Class for representing a [XMLDictReader](#).

```
#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)



### 27.368.1 Detailed Description

Class for representing a [XMLDictReader](#).

#### Note

bla Will read the DICOMV3.xml file

### 27.368.2 Constructor & Destructor Documentation

27.368.2.1 `gdcm::XMLDictReader::XMLDictReader ( )`

27.368.2.2 `gdcm::XMLDictReader::~XMLDictReader ( )` `[inline]`

### 27.368.3 Member Function Documentation

27.368.3.1 `void gdcm::XMLDictReader::CharacterDataHandler ( const char * data, int length )` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.368.3.2 `void gdcm::XMLDictReader::EndElement ( const char * name )` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.368.3.3 `const Dict& gdcm::XMLDictReader::GetDict ( )` `[inline]`

27.368.3.4 `void gdcm::XMLDictReader::HandleDescription ( const char ** atts )` `[protected]`

27.368.3.5 `void gdcm::XMLDictReader::HandleEntry ( const char ** atts )` `[protected]`

27.368.3.6 `void gdcm::XMLDictReader::StartElement ( const char * name, const char ** atts )` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

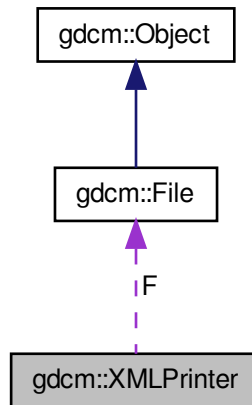
The documentation for this class was generated from the following file:

- [gdcmXMLDictReader.h](#)

## 27.369 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`

## 27.369.1 Member Enumeration Documentation

### 27.369.1.1 enum gdcm::XMLPrinter::PrintStyles

Enumerator

***OnlyUUID***

***LOADBULKDATA***

## 27.369.2 Constructor & Destructor Documentation

### 27.369.2.1 gdcm::XMLPrinter::XMLPrinter ( )

### 27.369.2.2 virtual gdcm::XMLPrinter::~~XMLPrinter ( ) [virtual]

## 27.369.3 Member Function Documentation

### 27.369.3.1 PrintStyles gdcm::XMLPrinter::GetPrintStyle ( ) const [inline]

### 27.369.3.2 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

### 27.369.3.3 void gdcm::XMLPrinter::Print ( std::ostream & *os* )

### 27.369.3.4 VR gdcm::XMLPrinter::PrintDataElement ( std::ostream & *os*, const Dicts & *dicts*, const DataSet & *ds*, const DataElement & *de*, const TransferSyntax & *ts* ) [protected]

### 27.369.3.5 void gdcm::XMLPrinter::PrintDataSet ( const DataSet & *ds*, const TransferSyntax & *ts*, std::ostream & *os* )

### 27.369.3.6 void gdcm::XMLPrinter::PrintSQ ( const SequenceOfItems \* *sqi*, const TransferSyntax & *ts*, std::ostream & *os* ) [protected]

### 27.369.3.7 void gdcm::XMLPrinter::SetFile ( File const & *f* ) [inline]

### 27.369.3.8 void gdcm::XMLPrinter::SetStyle ( PrintStyles *ps* ) [inline]

## 27.369.4 Member Data Documentation

### 27.369.4.1 const File\* gdcm::XMLPrinter::F [protected]

### 27.369.4.2 PrintStyles gdcm::XMLPrinter::PrintStyle [protected]

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

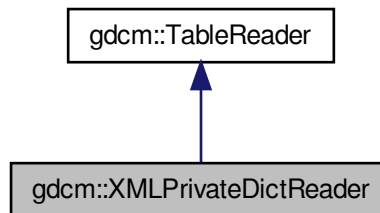
- [gdcmXMLPrinter.h](#)

## 27.370 gdcm::XMLPrivateDictReader Class Reference

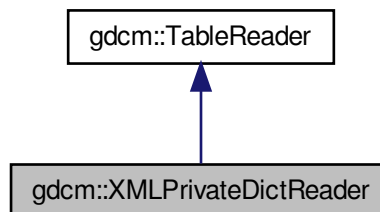
Class for representing a [XMLPrivateDictReader](#).

```
#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)

### 27.370.1 Detailed Description

Class for representing a [XMLPrivateDictReader](#).

#### Note

bla Will read the Private.xml file

### 27.370.2 Constructor & Destructor Documentation

27.370.2.1 `gdcm::XMLPrivateDictReader::XMLPrivateDictReader ( )`

27.370.2.2 `gdcm::XMLPrivateDictReader::~~XMLPrivateDictReader ( )` `[inline]`

### 27.370.3 Member Function Documentation

27.370.3.1 `void gdcm::XMLPrivateDictReader::CharacterDataHandler ( const char * data, int length )` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.370.3.2 `void gdcm::XMLPrivateDictReader::EndElement ( const char * name )` `[virtual]`

Reimplemented from [gdcm::TableReader](#).

27.370.3.3 `const PrivateDict& gdcm::XMLPrivateDictReader::GetPrivateDict ( )` `[inline]`

27.370.3.4 `void gdcm::XMLPrivateDictReader::HandleDescription ( const char ** atts )` `[protected]`

27.370.3.5 `void gdcm::XMLPrivateDictReader::HandleEntry ( const char ** atts )` `[protected]`

27.370.3.6 `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 28

# File Documentation

### 28.1 gdcm2pnm.man File Reference

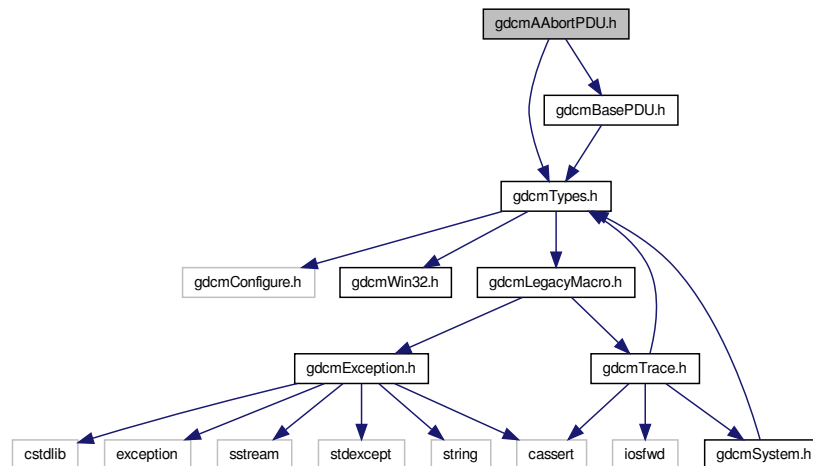
### 28.2 gdcm2vtk.man File Reference

### 28.3 gdcmAAbortPDU.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmBasePDU.h"
```

Include dependency graph for gdcmAAbortPDU.h:



## Classes

- class `gdcm::network::AAbortPDU`

*AAbortPDU Table 9-26 A-ABORT PDU FIELDS.*

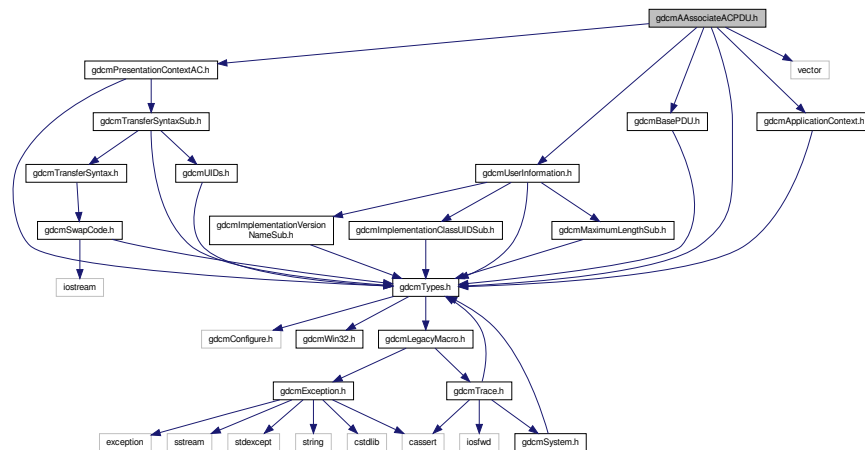
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

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

## Namespaces

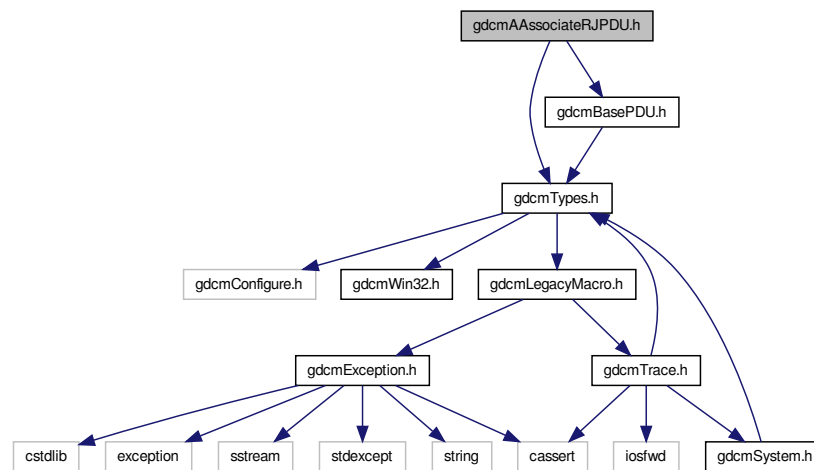
- [gdcm](#)
- [gdcm::network](#)

## 28.5 gdcmAAssociateRJPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```



Include dependency graph for gdcmAAssociateRJPDU.h:



## Classes

- class `gdcm::network::AAssociateRJPDU`

*AAssociateRJPDU Table 9-21 ASSOCIATE-RJ PDU FIELDS.*

## Namespaces

- `gdcm`
- `gdcm::network`

## 28.6 gdcmAAssociateRQPDU.h File Reference

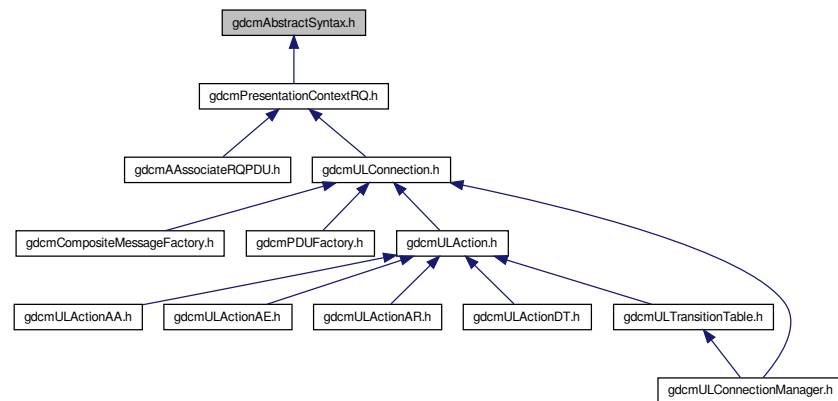
```

#include "gdcmTypes.h"
#include "gdcmVR.h"
#include "gdcmApplicationContext.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmUserInfo.h"
#include "gdcmBasePDU.h"

```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::AbstractSyntax](#)

*AbstractSyntax Table 9-14 ABSTRACT SYNTAX SUB-ITEM FIELDS.*

## Namespaces

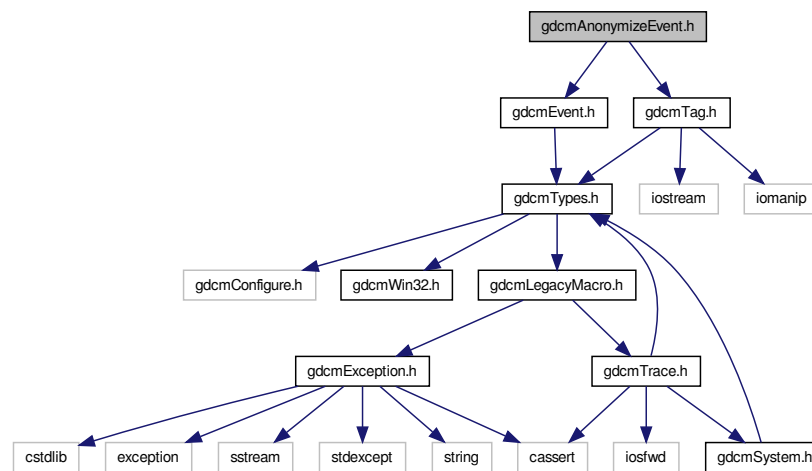
- [gdcm](#)
- [gdcm::network](#)

## 28.8 gdcmanon.man File Reference

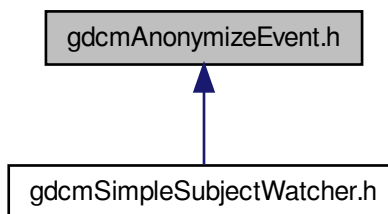
## 28.9 gdcmAnonymizeEvent.h File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```

Include dependency graph for `gdcmAnonymizeEvent.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::AnonymizeEvent`  
*AnonymizeEvent* Special type of event triggered during the Anonymization process.

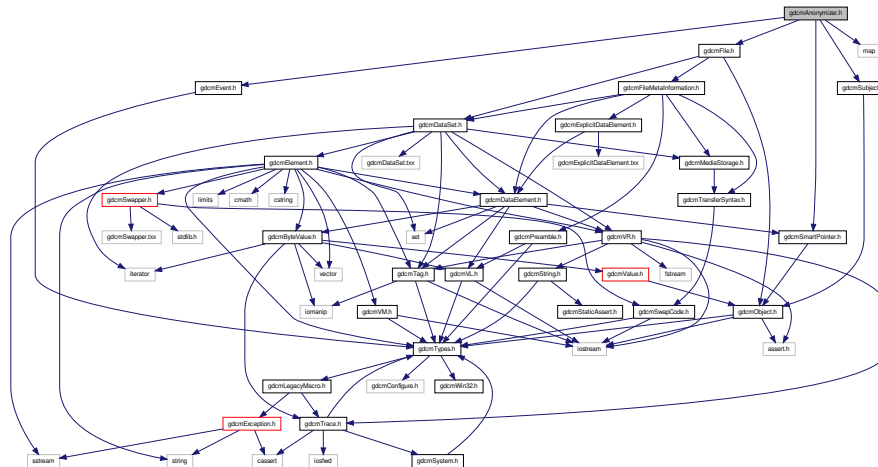
## Namespaces

- `gdcm`

## 28.10 gdcmAnonymizer.h File Reference

```
#include "gdcmFile.h"
#include "gdcmSubject.h"
#include "gdcmEvent.h"
#include "gdcmSmartPointer.h"
#include <map>
```

Include dependency graph for gdcmAnonymizer.h:



### Classes

- class [gdcm::Anonymizer](#)

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

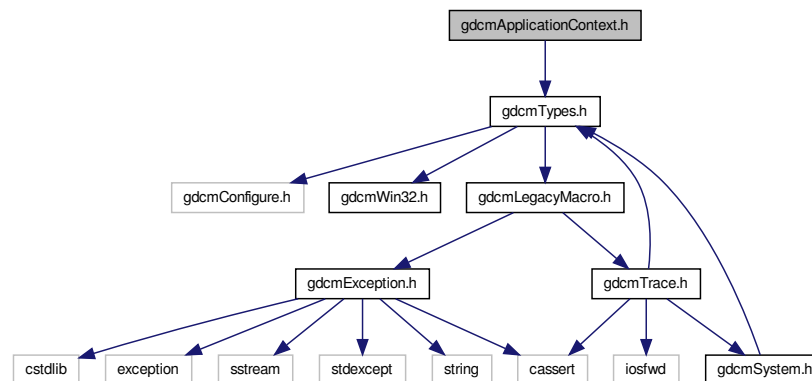
### Namespaces

- [gdcm](#)

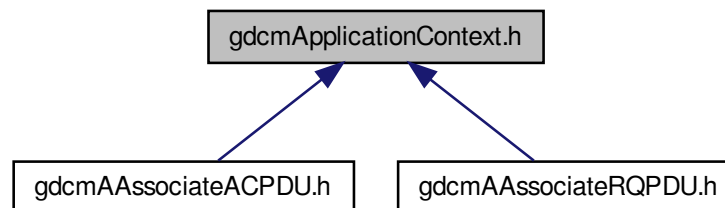
## 28.11 gdcmApplicationContext.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcApplicationContext.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdc::network::ApplicationContext`  
*ApplicationContext Table 9-12 APPLICATION CONTEXT ITEM FIELDS.*

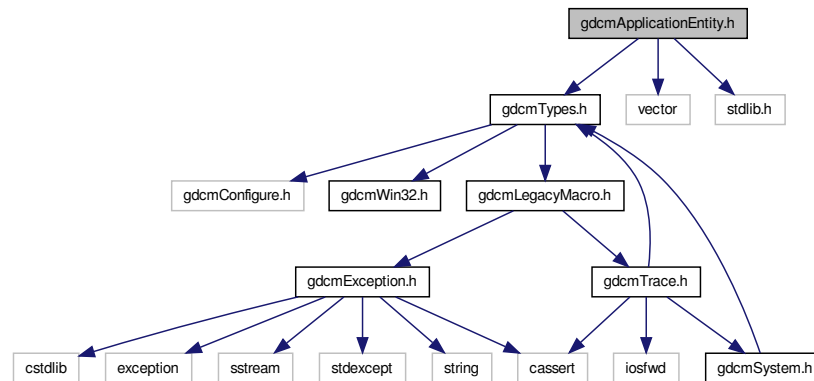
## Namespaces

- `gdc`
- `gdc::network`

## 28.12 gdcApplicationEntity.h File Reference

```
#include "gdcTypes.h"
```

```
#include <vector>
#include <stdlib.h>
Include dependency graph for gdcmApplicationEntity.h:
```



## Classes

- class `gdcm::ApplicationEntity`

*ApplicationEntity.*

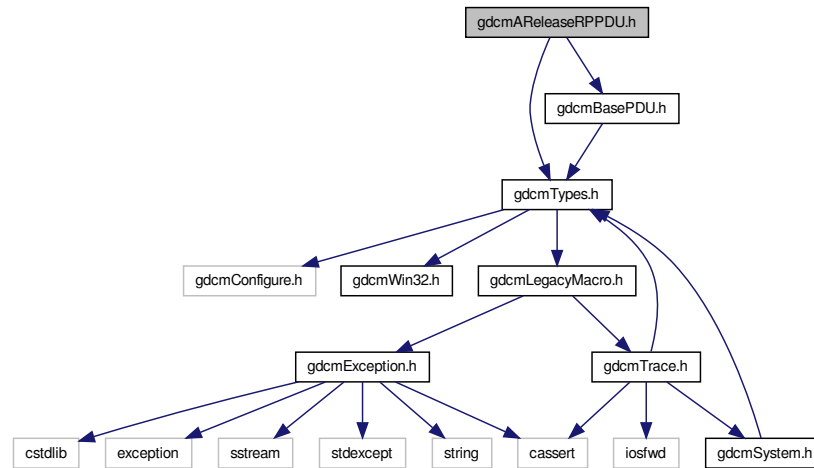
## Namespaces

- `gdcm`

## 28.13 gdcmAReleaseRPPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```

Include dependency graph for `gdcmAReleaseRPPDU.h`:



## Classes

- class `gdcm::network::AReleaseRPPDU`

*AReleaseRPPDU Table 9-25 A-RELEASE-RP PDU fields.*

## Namespaces

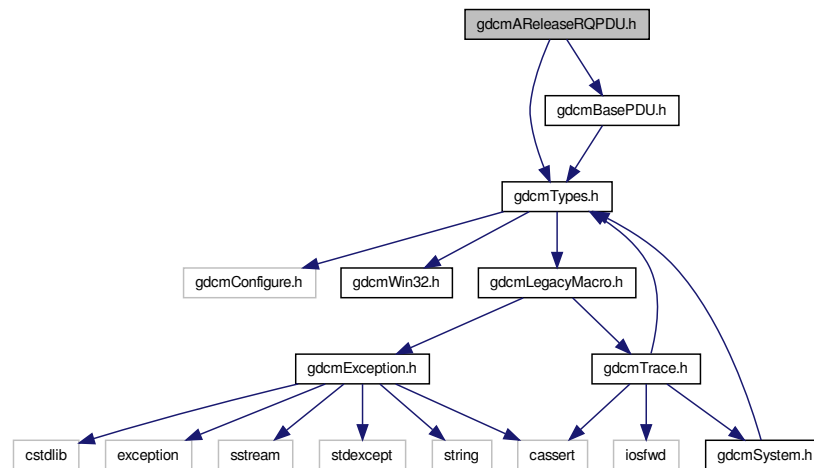
- `gdcm`
- `gdcm::network`

## 28.14 gdcmAReleaseRQPDU.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmBasePDU.h"
```



Include dependency graph for gdcmaReleaseRQPDU.h:



## Classes

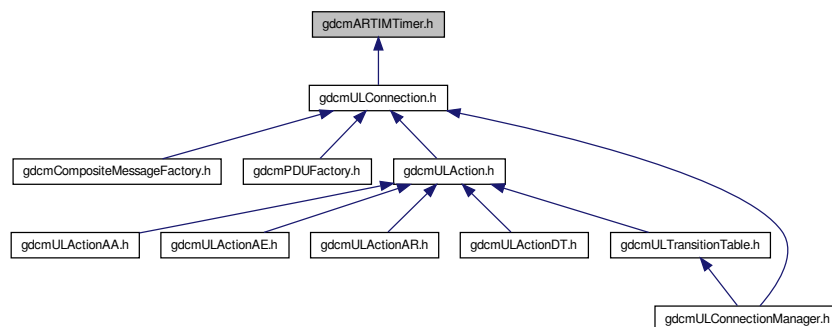
- class [gdcma::network::AReleaseRQPDU](#)  
[AReleaseRQPDU Table 9-24 A-RELEASE-RQ PDU FIELDS.](#)

## Namespaces

- [gdcma](#)
- [gdcma::network](#)

## 28.15 gdcmaRTIMTimer.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ARTIMTimer](#)

[ARTIMTimer](#) This file contains the code for the ARTIM timer.

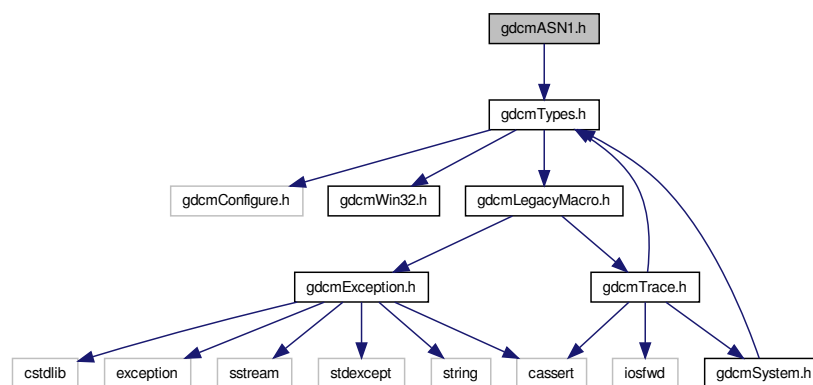
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.16 gdcmASN1.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmASN1.h:



## Classes

- class [gdcm::ASN1](#)

Class for [ASN1](#).

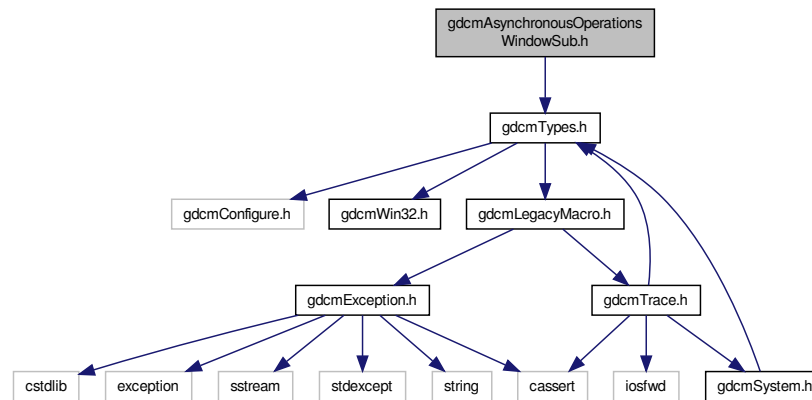
## Namespaces

- [gdcm](#)

## 28.17 gdcmAsynchronousOperationsWindowSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmAsynchronousOperationsWindowSub.h:



### Classes

- class [gdcm::network::AsynchronousOperationsWindowSub](#)

*AsynchronousOperationsWindowSub* PS 3.7 Table D.3-7 ASYNCHRONOUS OPERATIONS WINDOW SUB-ITEM FIE↔  
LDS (A-ASSOCIATE-RQ)

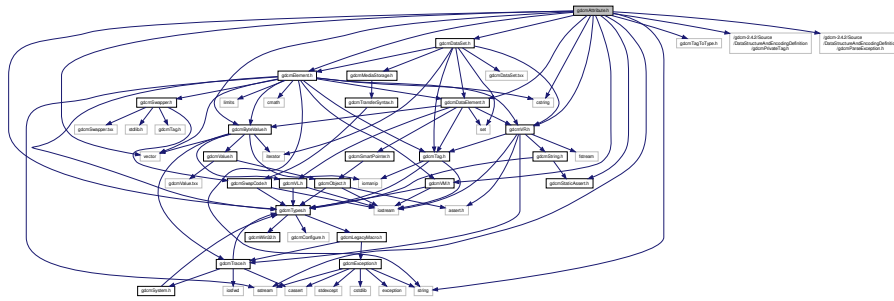
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

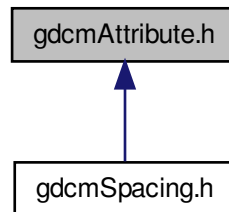
## 28.18 gdcmAttribute.h File Reference

```
#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 <sstream>
```

Include dependency graph for `gdcmAttribute.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::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.*
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_3 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_8 >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM1_n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM2_2n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM2_n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM3_3n >`
- class `gdcm::Attribute< Group, Element, TVR, VM::VM3_n >`
- class `gdcm::VRVLSIZE< T >`
- class `gdcm::VRVLSIZE< 0 >`
- class `gdcm::VRVLSIZE< 1 >`

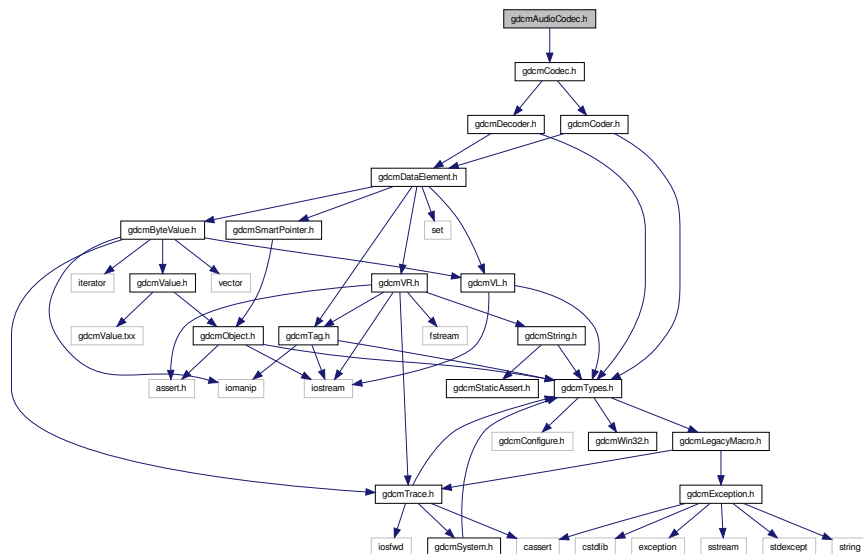
## Namespaces

- `gdcm`

## 28.19 gdcmAudioCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for gdcmAudioCodec.h:



### Classes

- class [gdcm::AudioCodec](#)

*AudioCodec.*

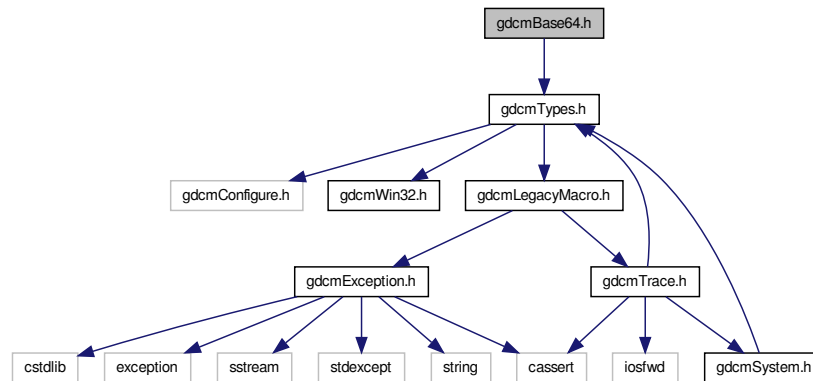
### Namespaces

- [gdcm](#)

## 28.20 gdcmBase64.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmBase64.h`:



## Classes

- class [gdcm::Base64](#)

*Class for [Base64](#).*

## Namespaces

- [gdcm](#)

## 28.21 gdcmBaseCompositeMessage.h File Reference

```

#include "gdcmPresentationDataValue.h"
#include "gdcmBaseRootQuery.h"
#include <vector>

```

[illegible]

```

graph BT
    gdcBase[gdcmBaseCompositeMessage.h]
    gdcEcho[gdcmCEchoMessages.h]
    gdcFind[gdcmCFindMessages.h]
    gdcMove[gdcmCMoveMessages.h]
    gdcStore[gdcmCStoreMessages.h]
    gdcEcho --> gdcBase
    gdcFind --> gdcBase
    gdcMove --> gdcBase
    gdcStore --> gdcBase

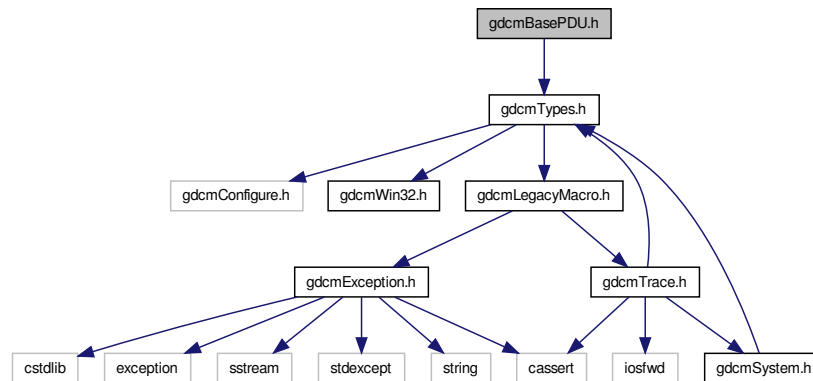
```

- class `gdcm::network::BaseCompositeMessage`

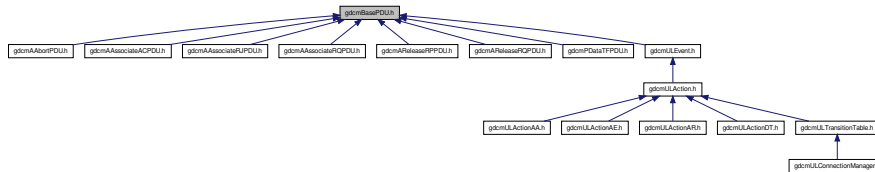
- gdc
- gdc::network

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen

Include dependency graph for `gdcmBasePDU.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::network::BasePDU`  
*BasePDU* base class for PDUs.

## Namespaces

- `gdcm`
- `gdcm::network`

## 28.23 gdcmBaseRootQuery.h File Reference

```

#include "gdcmDataSet.h"
#include "gdcmUIDs.h"
#include "gdcmObject.h"
#include "gdcmQueryPatient.h"
#include "gdcmQueryStudy.h"
#include "gdcmQuerySeries.h"
#include "gdcmQueryImage.h"

```



[illegible]

```

graph TD
    gsdmBasePoolQuery.h --> gsdmBaseCompositeMessage.h
    gsdmBasePoolQuery.h --> gsdmCompositeNetworkFunctions.h
    gsdmBasePoolQuery.h --> gsdmFirstPatientRootQuery.h
    gsdmBasePoolQuery.h --> gsdmFirstStudyRootQuery.h
    gsdmBasePoolQuery.h --> gsdmMoveStudyRootQuery.h
    gsdmBasePoolQuery.h --> gsdmQueryFactory.h
    gsdmBasePoolQuery.h --> gsdmMovePatientRootQuery.h

    gsdmBaseCompositeMessage.h --> gsdmCFindMessages.h
    gsdmBaseCompositeMessage.h --> gsdmCExecMessages.h
    gsdmBaseCompositeMessage.h --> gsdmCStoreMessages.h

    gsdmCompositeNetworkFunctions.h --> gsdmCMoveMessages.h

    gsdmFirstPatientRootQuery.h --> gsdmMovePatientRootQuery.h
  
```

- class `gdcm::BaseRootQuery`

## Namespaces

- **gdcm**

- enum `gdcmm::EQueryLevel` {  
    `gdcmm::ePatient` = 0,  
    `gdcmm::eStudy` = 1,  
    `gdcmm::eSeries` = 2,  
    `gdcmm::eImage` = 3 }
- enum `gdcmm::EQueryType` {  
    `gdcmm::eFind` = 0,  
    `gdcmm::eMove` }



## Namespaces

- **gdcm**

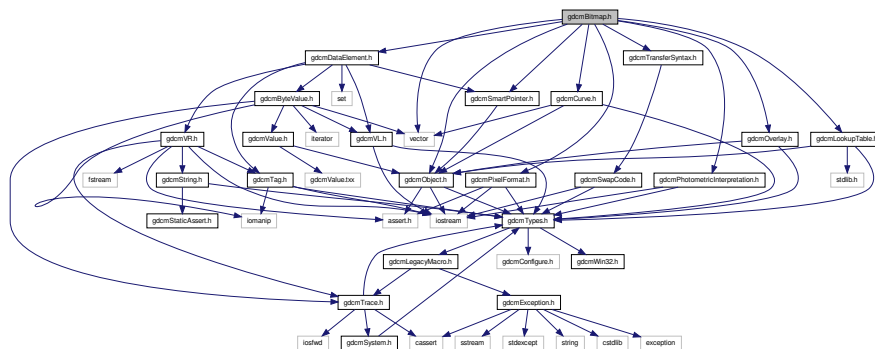
## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const BasicOffsetTable &val)`

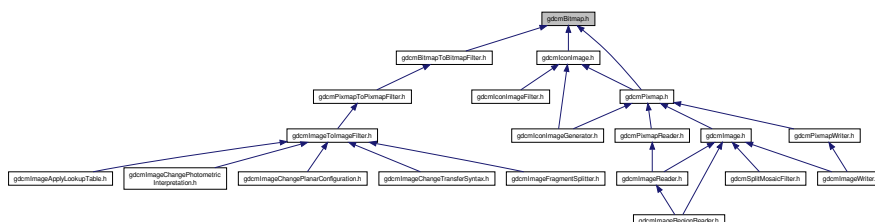
## 28.25 gdcmBitmap.h File Reference

```
#include "gdcmObject.h"
#include "gdcmCurve.h"
#include "gdcmDataElement.h"
#include "gdcmLookupTable.h"
#include "gdcmOverlay.h"
#include "gdcmPhotometricInterpretation.h"
#include "gdcmPixelFormat.h"
#include "gdcmSmartPointer.h"
#include "gdcmTransferSyntax.h"
#include <vector>
```

Include dependency graph for `gdcmBitmap.h`:



This graph shows which files directly or indirectly include this file:





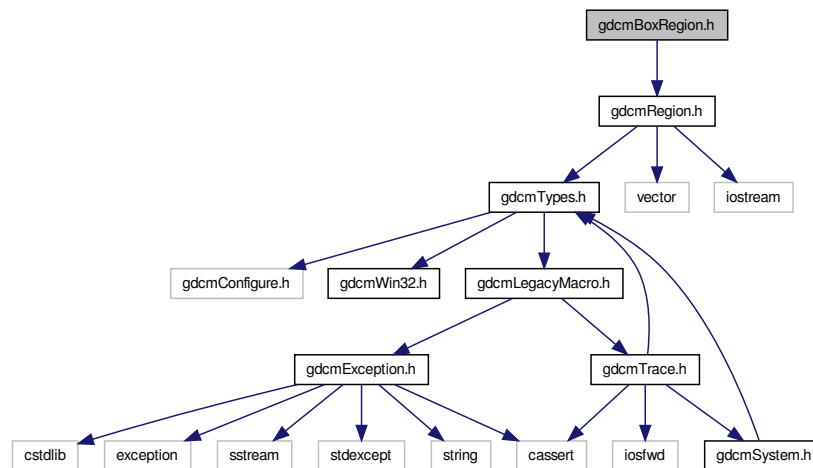
## Namespaces

- [gdcm](#)

## 28.27 gdcmBoxRegion.h File Reference

```
#include "gdcmRegion.h"
```

Include dependency graph for gdcmBoxRegion.h:



## Classes

- class [gdcm::BoxRegion](#)

*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)*

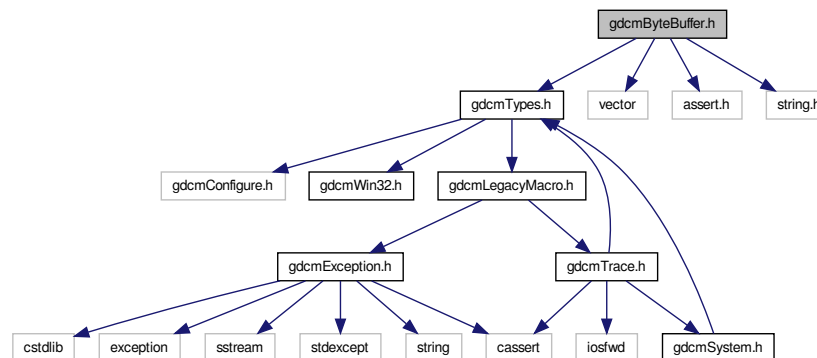
## Namespaces

- [gdcm](#)

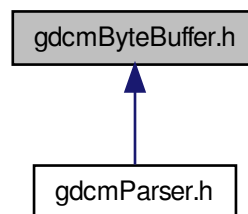
## 28.28 gdcmByteBuffer.h File Reference

```
#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`

## 28.29 gdcmByteSwap.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmSwapCode.h"
#include "gdcmByteSwap.txx"

```

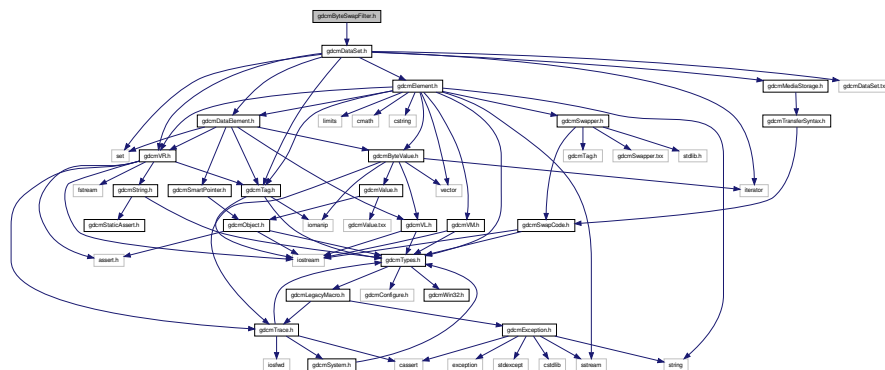
```

graph TD
    gdcmByteSwap.h --> gdcmByteSwap.h
    gdcmByteSwap.h --> gdcmSwapCode.h
    gdcmByteSwap.h --> gdcmByteSwap.box
    gdcmByteSwap.h --> stdlib.h
    gdcmSwapCode.h --> gdcmTypes.h
    gdcmSwapCode.h --> iostream
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmException.h
    gdcmTypes.h --> gdcmTrace.h
    gdcmTypes.h --> gdcmSystem.h
    gdcmLegacyMacro.h --> gdcmByteSwap.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmException.h --> cstdlb
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
    gdcmException.h --> cassert
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h
  
```

- class `gdcm::ByteSwap< T >`  
*ByteSwap*.

- **gdcm**

```
#include "gdcmDataSet.h"
Include dependency graph for gdcmByteSwapFilter.h:
```



- class `gdcm::ByteSwapFilter`

*ByteSwapFilter* In place byte-swapping of a dataset *FIXME: FL status ??*

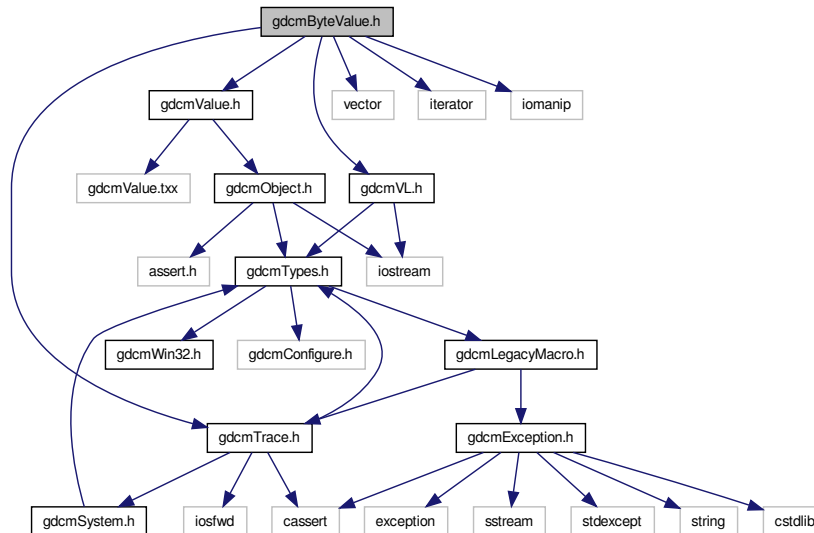
## Namespaces

- [gdc](#)

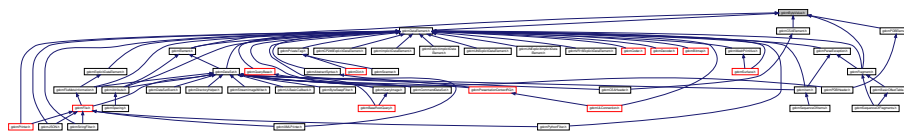
## 28.31 gdcByteValue.h File Reference

```
#include "gdcValue.h"
#include "gdcTrace.h"
#include "gdcVL.h"
#include <vector>
#include <iterator>
#include <iomanip>
```

Include dependency graph for gdcByteValue.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::ByteValue](#)  
Class to represent binary value (array of bytes)



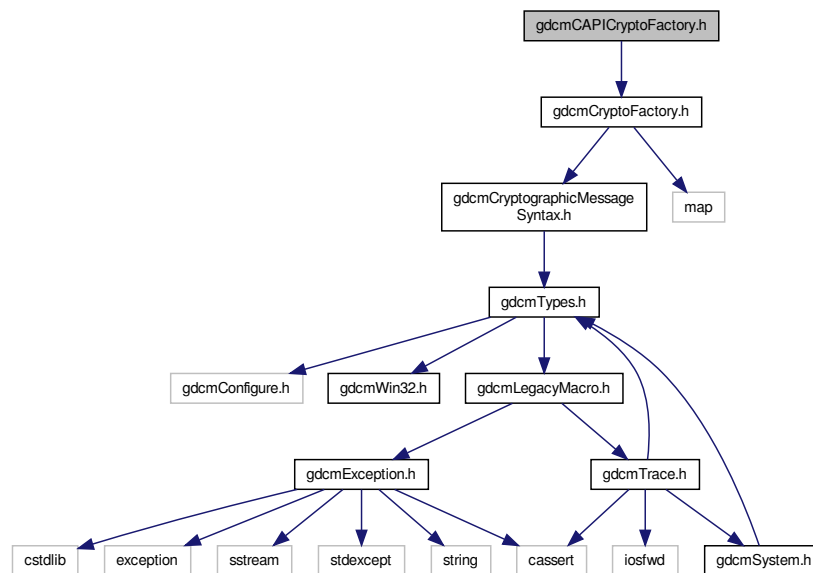
## Namespaces

- [gdcm](#)

## 28.32 gdcmCAPICryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
```

Include dependency graph for gdcmCAPICryptoFactory.h:



## Classes

- class [gdcm::CAPICryptoFactory](#)

## Namespaces

- [gdcm](#)

## 28.33 gdcmCAPICryptographicMessageSyntax.h File Reference

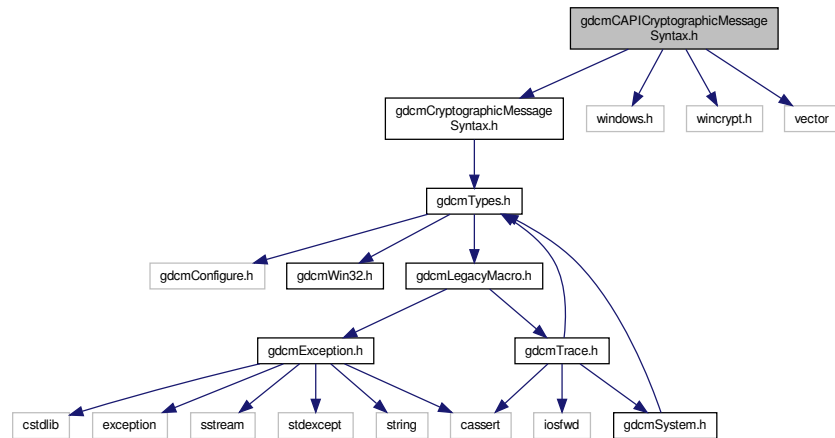
```
#include "gdcmCryptographicMessageSyntax.h"
```

```
#include <windows.h>
```

```
#include <wincrypt.h>
```

```
#include <vector>
```

Include dependency graph for `gdcmCAPICryptographicMessageSyntax.h`:



## Classes

- class [gdcm::CAPICryptographicMessageSyntax](#)

## Namespaces

- [gdcm](#)

## 28.34 gdcmCEchoMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```

- class `gdcm::network::CEchoRQ`  
*CEchoRQ* this file defines the messages for the cecho action.
- class `gdcm::network::CEchoRSP`  
*CEchoRSP* this file defines the messages for the cecho action.

- gdc
- gdc::network

```
#include "gdcmBaseCompositeMessage.h"
#include "gdcmBaseRootQuery.h"
```



- class `gdcm::network::CMoveCancelRq`
- class `gdcm::network::CMoveRQ`  
*`CMoveRQ` this file defines the messages for the cmove action.*
- class `gdcm::network::CMoveRSP`  
*`CMoveRSP` this file defines the messages for the cmove action.*

- `gdcm`
- `gdcm::network`

```
#include "gdcmCoder.h"
#include "gdcmDecoder.h"
```

```

graph TD
    gdcmCodec.h --> gdcmDecoder.h
    gdcmCodec.h --> gdcmCoder.h
    gdcmDecoder.h --> gdcmDataElement.h
    gdcmCoder.h --> gdcmDataElement.h
    gdcmDataElement.h --> gdcmByteValue.h
    gdcmDataElement.h --> gdcmSmartPointer.h
    gdcmDataElement.h --> set
    gdcmDataElement.h --> gdcmVoxel.h
    gdcmDataElement.h --> gdcmVoxel.h
    gdcmDataElement.h --> gdcmString.h
    gdcmDataElement.h --> gdcmStaticAssert.h
    gdcmDataElement.h --> gdcmTypes.h
    gdcmDataElement.h --> gdcmTrace.h
    gdcmDataElement.h --> gdcmException.h
    gdcmByteValue.h --> iterator
    gdcmByteValue.h --> gdcmValue.h
    gdcmByteValue.h --> vector
    gdcmValue.h --> gdcmValueTox
    gdcmSmartPointer.h --> gdcmObject.h
    gdcmSmartPointer.h --> gdcmTag.h
    gdcmSmartPointer.h --> gdcmVoxel.h
    gdcmSmartPointer.h --> gdcmVoxel.h
    gdcmSmartPointer.h --> gdcmString.h
    gdcmSmartPointer.h --> gdcmStaticAssert.h
    gdcmSmartPointer.h --> gdcmTypes.h
    gdcmSmartPointer.h --> gdcmTrace.h
    gdcmSmartPointer.h --> gdcmException.h
    gdcmObject.h --> assert.h
    gdcmObject.h --> iomanip
    gdcmObject.h --> iosstream
    gdcmTag.h --> iosstream
    gdcmVoxel.h --> iosstream
    gdcmVoxel.h --> gdcmString.h
    gdcmVoxel.h --> gdcmStaticAssert.h
    gdcmVoxel.h --> gdcmTypes.h
    gdcmVoxel.h --> gdcmTrace.h
    gdcmVoxel.h --> gdcmException.h
    gdcmString.h --> gdcmStaticAssert.h
    gdcmString.h --> gdcmTypes.h
    gdcmString.h --> gdcmTrace.h
    gdcmString.h --> gdcmException.h
    gdcmStaticAssert.h --> gdcmTypes.h
    gdcmStaticAssert.h --> gdcmTrace.h
    gdcmStaticAssert.h --> gdcmException.h
    gdcmTypes.h --> gdcmConfigure.h
    gdcmTypes.h --> gdcmWin32.h
    gdcmTypes.h --> gdcmLegacyMacro.h
    gdcmTypes.h --> gdcmTrace.h
    gdcmTypes.h --> gdcmException.h
    gdcmConfigure.h --> gdcmTrace.h
    gdcmWin32.h --> gdcmTrace.h
    gdcmLegacyMacro.h --> gdcmTrace.h
    gdcmLegacyMacro.h --> gdcmException.h
    gdcmTrace.h --> iosfwd
    gdcmTrace.h --> gdcmSystem.h
    gdcmTrace.h --> cassert
    gdcmTrace.h --> cstdlib
    gdcmTrace.h --> exception
    gdcmTrace.h --> sstream
    gdcmTrace.h --> stdexcept
    gdcmTrace.h --> string
    gdcmException.h --> iosfwd
    gdcmException.h --> gdcmSystem.h
    gdcmException.h --> cassert
    gdcmException.h --> cstdlib
    gdcmException.h --> exception
    gdcmException.h --> sstream
    gdcmException.h --> stdexcept
    gdcmException.h --> string
  
```

[illegible]

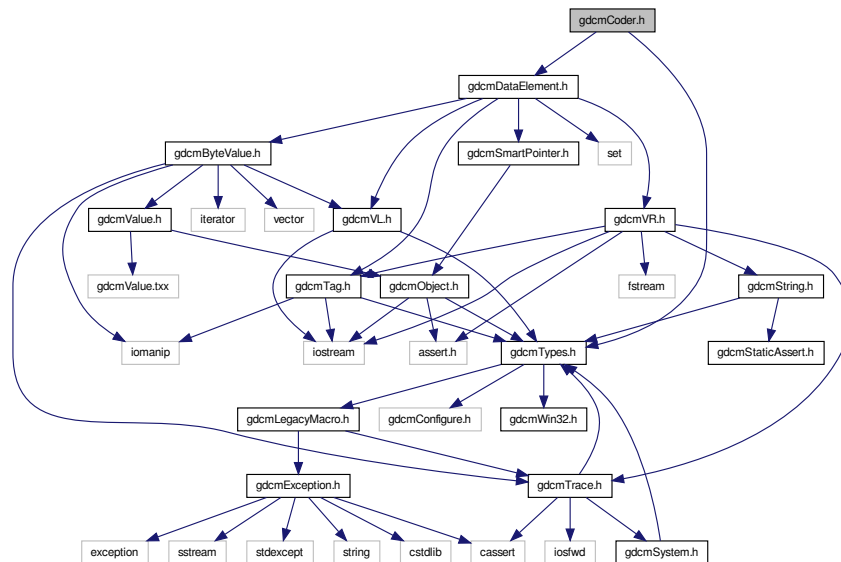
- class `gdcm::Codec`

## Namespaces

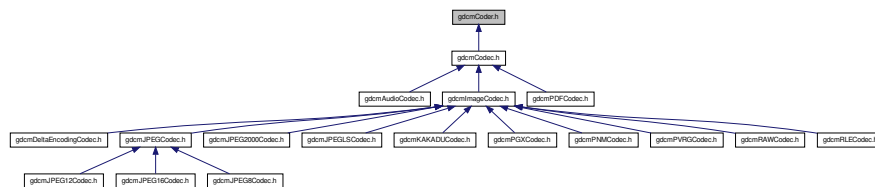
- **gdcm**

```
#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.*

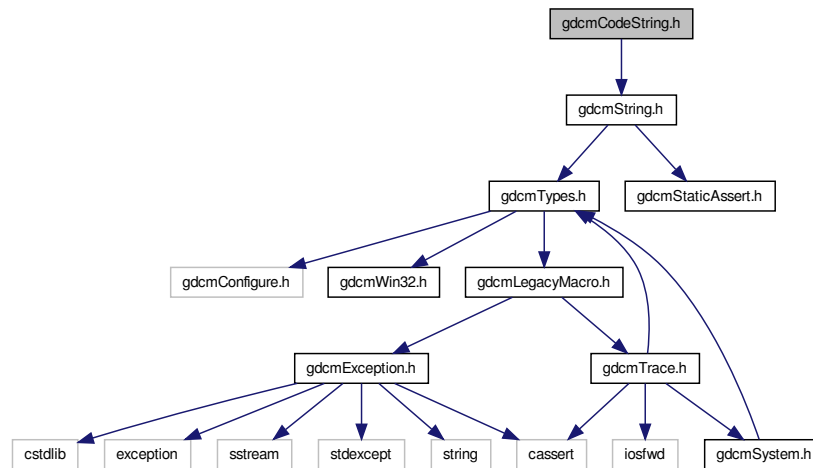
## Namespaces

- [gdcm](#)

## 28.39 gdcmCodeString.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for `gdcmCodeString.h`:



## Classes

- class `gdcm::CodeString`

*`CodeString` This is an implementation of DICOM [VR](#): CS The ctor will properly Trim so that operator== is correct.*

## Namespaces

- `gdcm`

## Functions

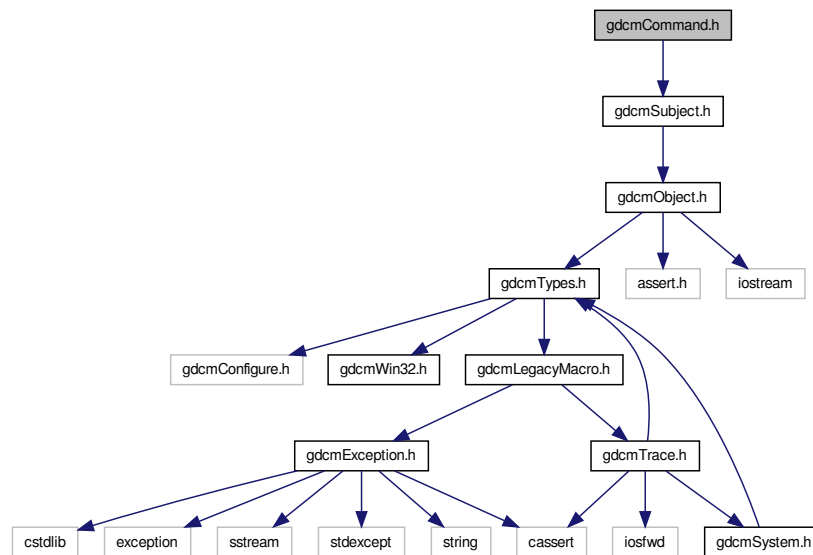
- bool `gdcm::operator!=` (const CodeString &ref, const CodeString &cs)
- std::ostream & `gdcm::operator<<` (std::ostream &os, const CodeString &str)
- bool `gdcm::operator==` (const CodeString &ref, const CodeString &cs)

## 28.40 gdcmCommand.h File Reference

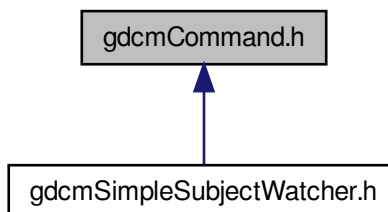
```
#include "gdcmSubject.h"
```



Include dependency graph for gdcCommand.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::Command](#)  
*Command superclass for callback/observer methods.*
- class [gdc::MemberCommand< T >](#)  
*Command subclass that calls a pointer to a member function.*
- class [gdc::SimpleMemberCommand< T >](#)  
*Command subclass that calls a pointer to a member function.*



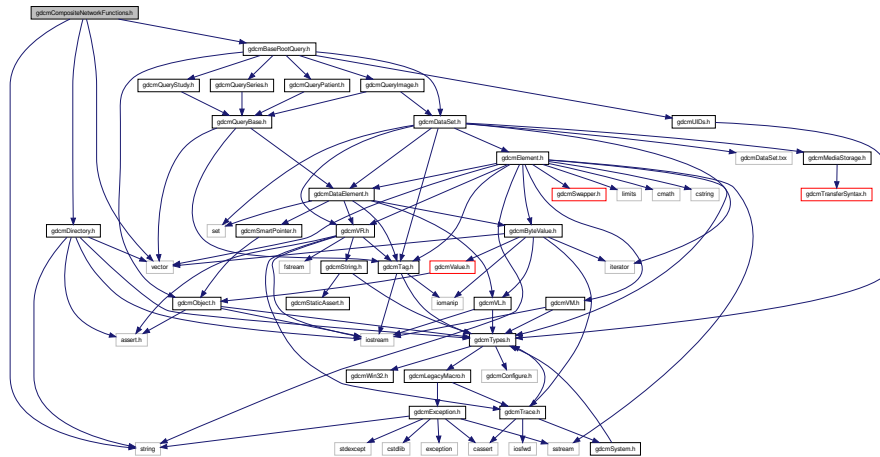
[illegible]

- class `gdcm::network::CompositeMessageFactory`

## Namespaces

- `gdcm`
- `gdcm::network`

```
#include "gdcmDirectory.h"
#include "gdcmBaseRootQuery.h"
#include <vector>
#include <string>
```



- class `gdcm::CompositeNetworkFunctions`

- 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:

*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:

- **gdcm**

### Classes

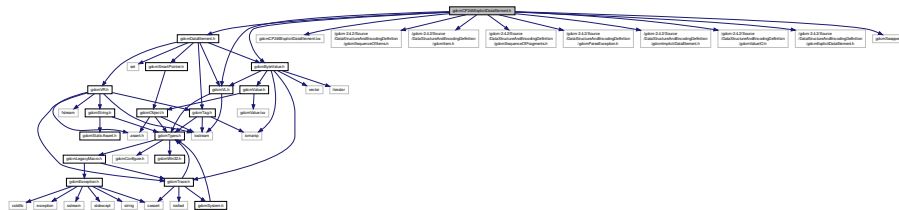
- class `gdcm::ConstCharWrapper`

- Do not use me.*

- **gdcm**

## 28.46 gdcmCP246ExplicitDataElement.h File Reference

```
#include "gdcMDataElement.h"
#include "gdcMCP246ExplicitDataElement.txx"
Include dependency graph for gdcMCP246ExplicitDataElement.h:
```



## Classes

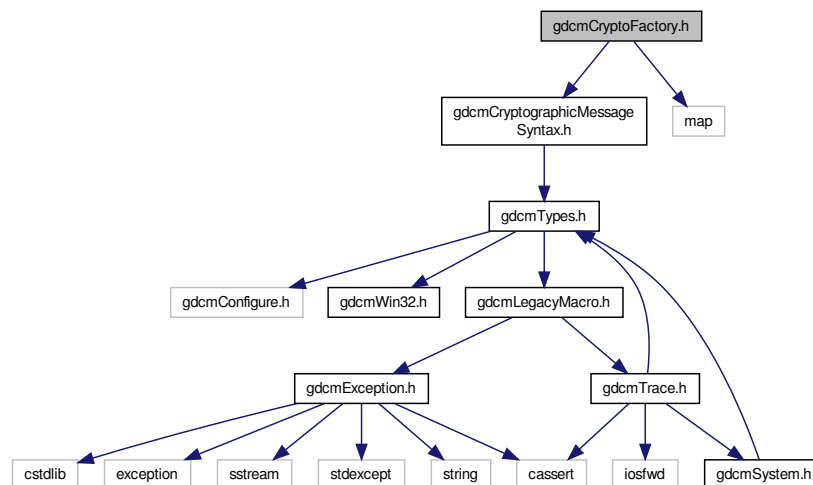
- class `gdc::CP246ExplicitDataElement`  
*Class to read/write a [DataElement](#) as CP246Explicit Data [Element](#).*

## Namespaces

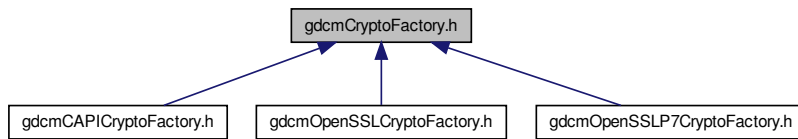
- **gdcm**

## 28.47 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 `gdcrypto::CryptoFactory`

*Class to do handle the crypto factory.*

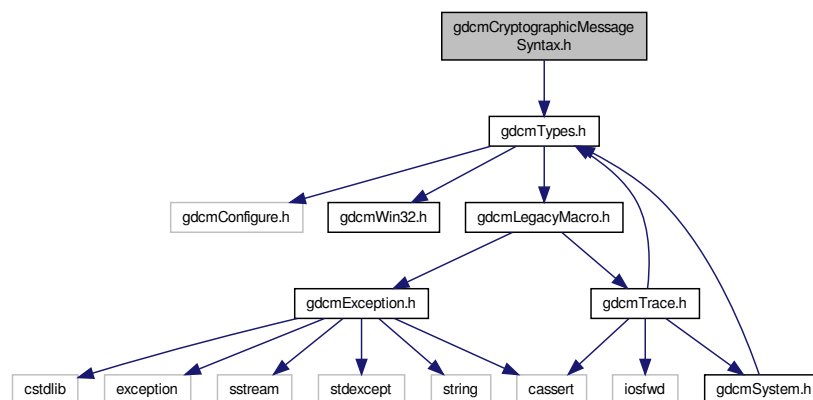
## Namespaces

- `gdcrypto`

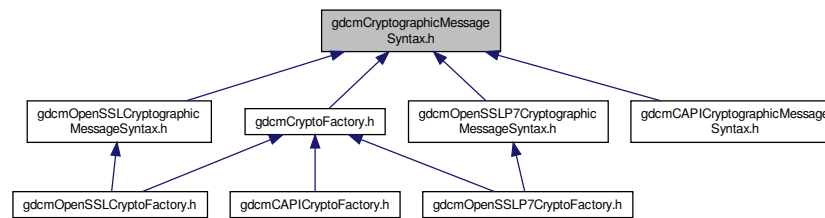
## 28.48 gdcryptographicMessageSyntax.h File Reference

```
#include "gdcryptoTypes.h"
```

Include dependency graph for `gdcryptographicMessageSyntax.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::CryptographicMessageSyntax`

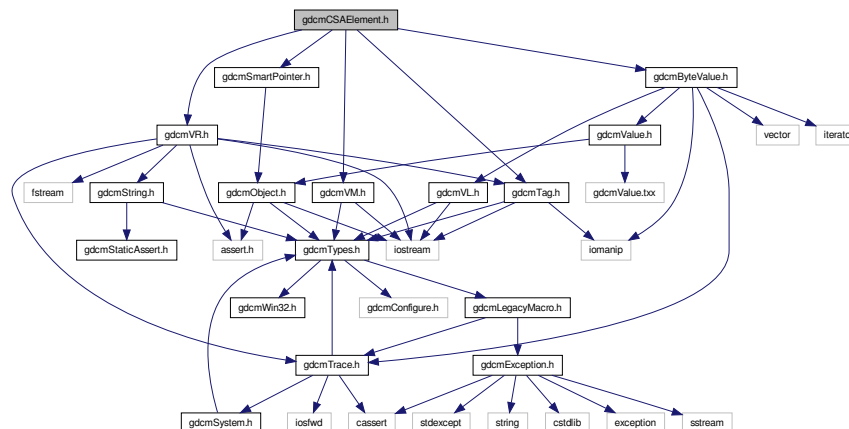
## Namespaces

- **gdcm**

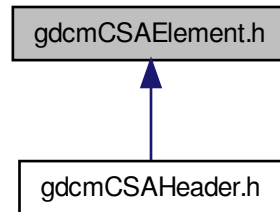
## 28.49 gdcMCSAElement.h File Reference

```
#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 [gdcms::CSAElement](#)

*Class to represent a CSA [Element](#).*

## Namespaces

- [gdcms](#)

## Functions

- `std::ostream & gdcms::operator<< (std::ostream &os, const CSAElement &val)`

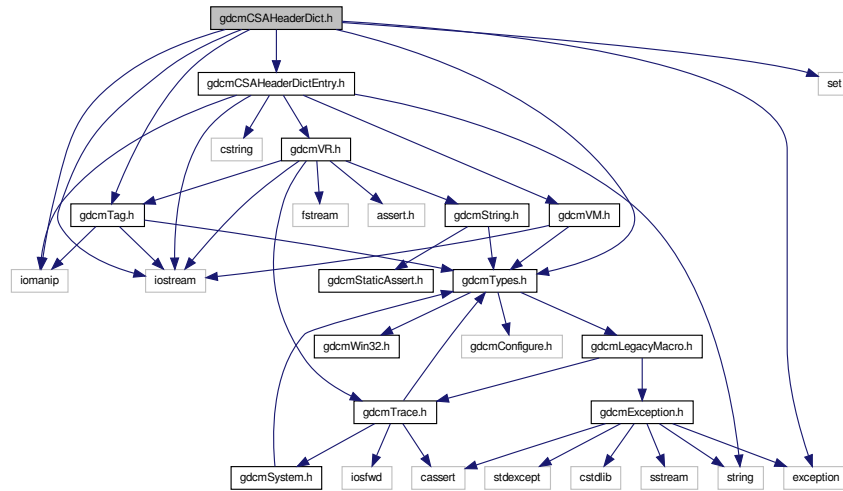
## 28.50 gdcmsAHeader.h File Reference

```
#include "gdcmsTypes.h"
#include "gdcmsDataSet.h"
#include "gdcmsCSAElement.h"
```

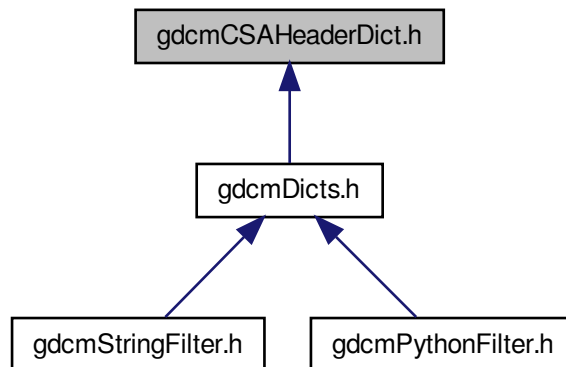




Include dependency graph for `gdcmCSAHeaderDict.h`:



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](#)

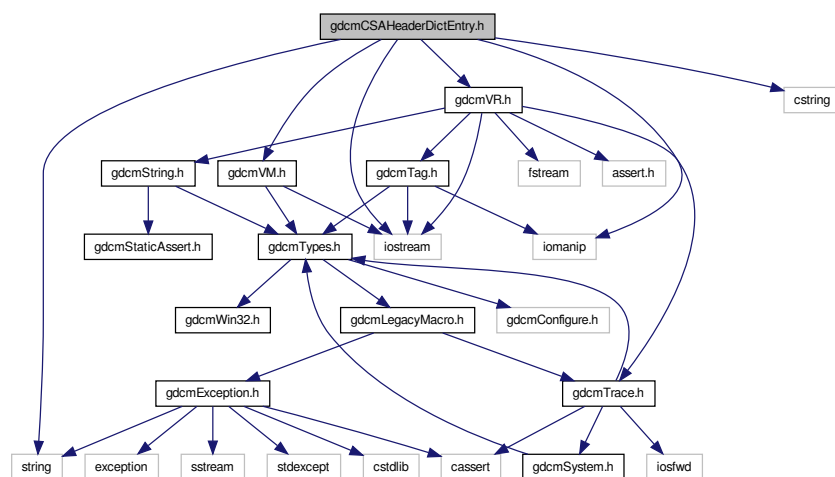
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDict &val)`

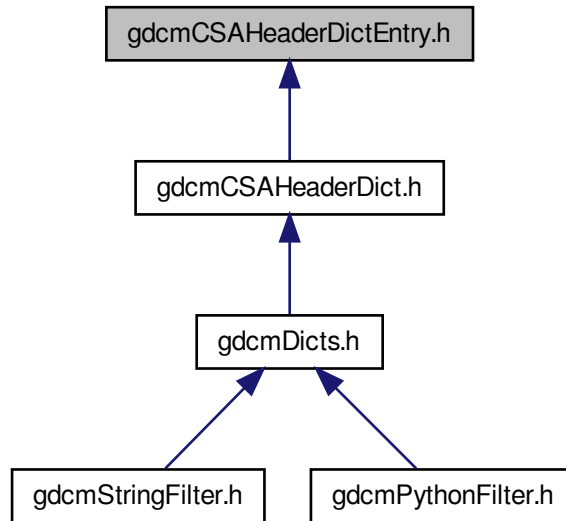
## 28.52 gdcmCSAHeaderDictEntry.h File Reference

```
#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](#). 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.*

## Namespaces

- [gdcm](#)

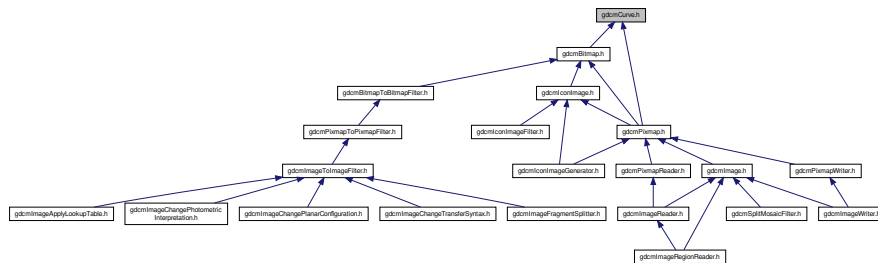
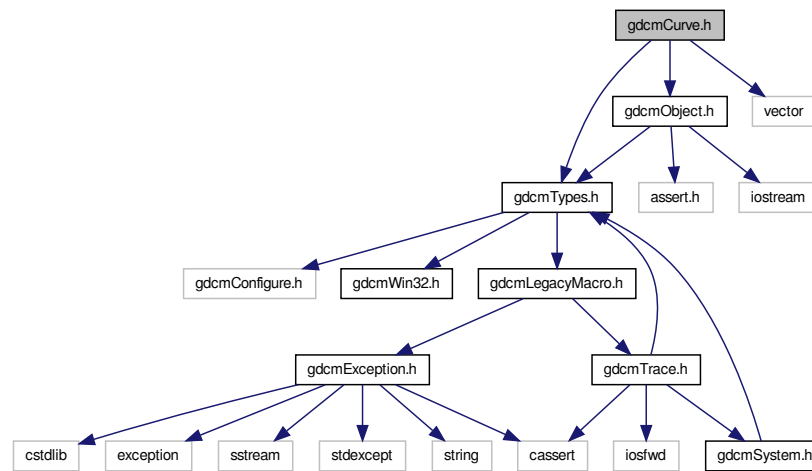
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const CSAHeaderDictEntry &val)`

## 28.53 gdcmCStoreMessages.h File Reference

```
#include "gdcmBaseCompositeMessage.h"
```





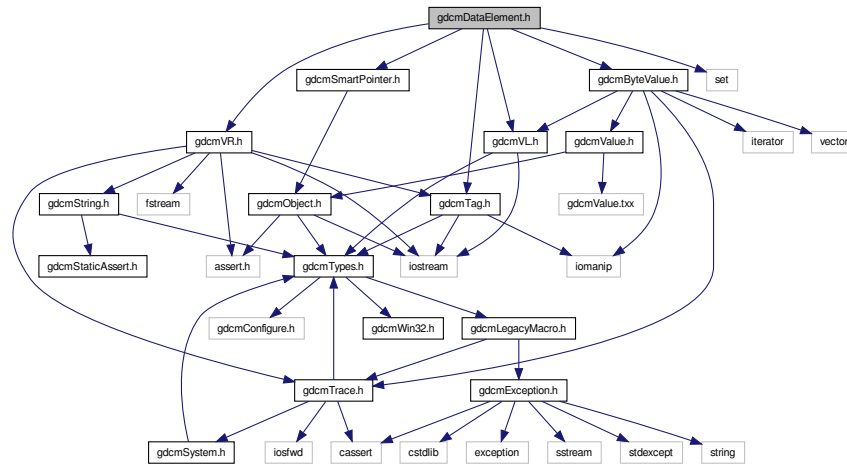
- class `gdc::Curve`

- **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

- `gdcm`

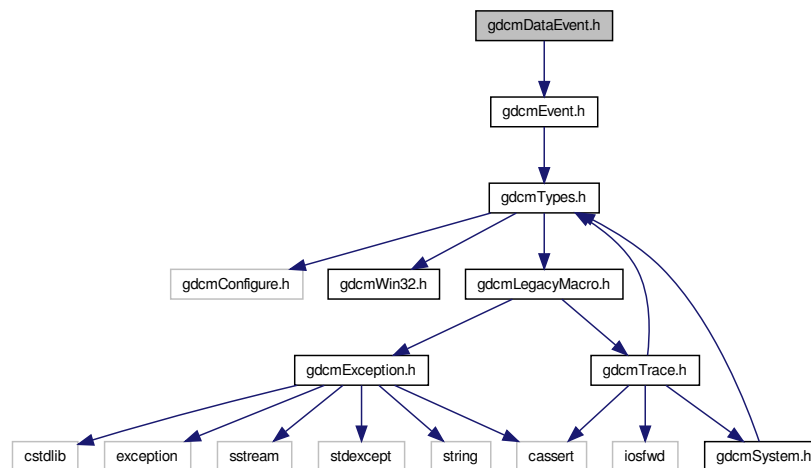
## Functions

- bool `gdcm::operator!=` (const DataElement &lhs, const DataElement &rhs)
- std::ostream & `gdcm::operator<<` (std::ostream &os, const DataElement &val)

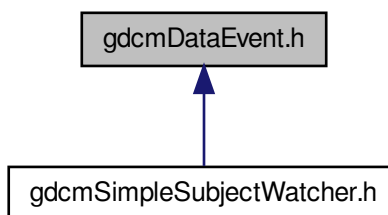
## 28.56 gdcmDataEvent.h File Reference

```
#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`



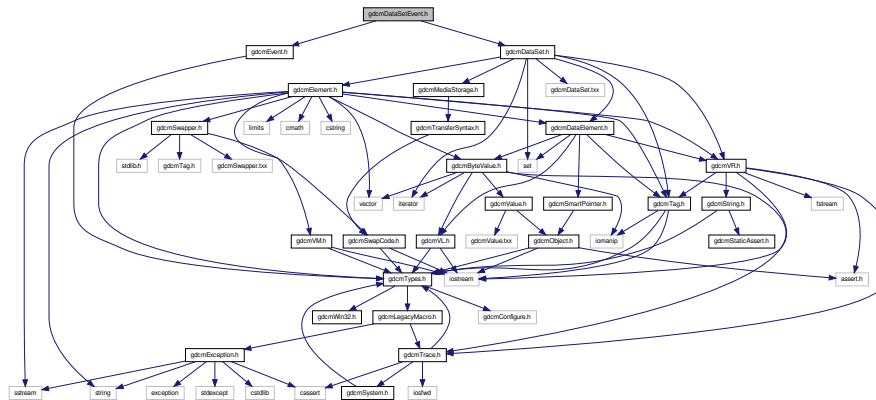


## 28.58 gdcmDataSetEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmDataSet.h"
```

Include dependency graph for `gdcmDataSetEvent.h`:



## Classes

- class `gdcm::DataSetEvent`

**DataSetEvent** Special type of event triggered during the **DataSet** store/move process.

## Namespaces

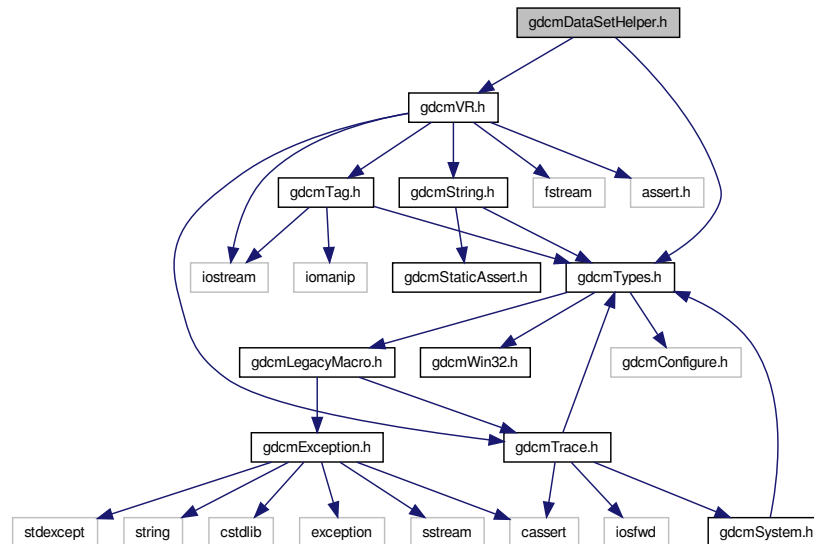
- **gdcm**

## 28.59 gdcmDataSetHelper.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmmVR.h"
```

Include dependency graph for gdcmDataSetHelper.h:



## Classes

- class [gdcm::DataSetHelper](#)

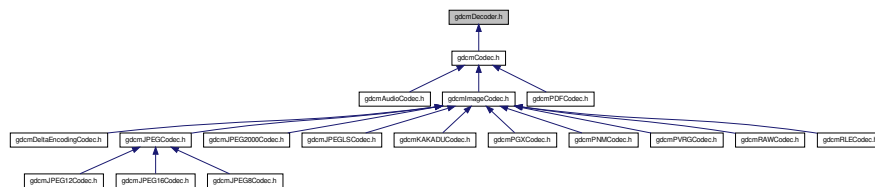
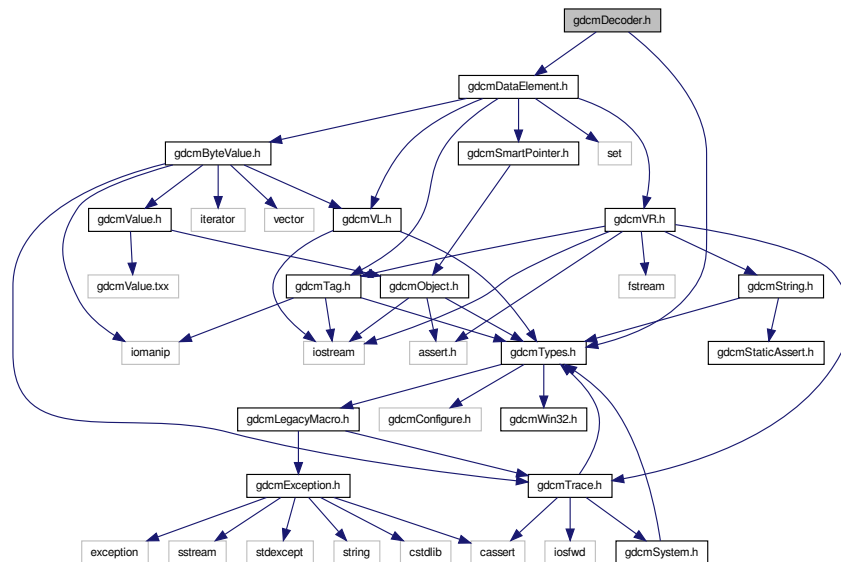
*[DataSetHelper](#) (internal class, not intended for user level)*

## Namespaces

- [gdcm](#)

## 28.60 gdcmDecoder.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataElement.h"
```



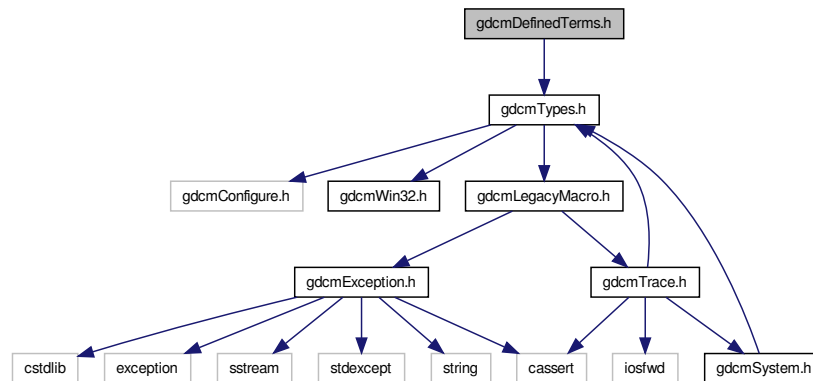
- class `gdcm::Decoder`  
*Decoder.*

- **gdcm**

```
#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.*

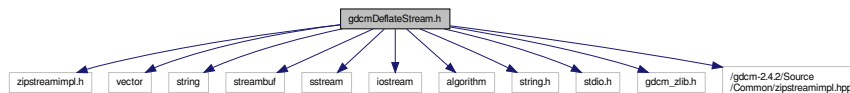
## Namespaces

- [gdcm](#)

## 28.62 gdcmDeflateStream.h File Reference

```
#include "zipstreamimpl.h"
```

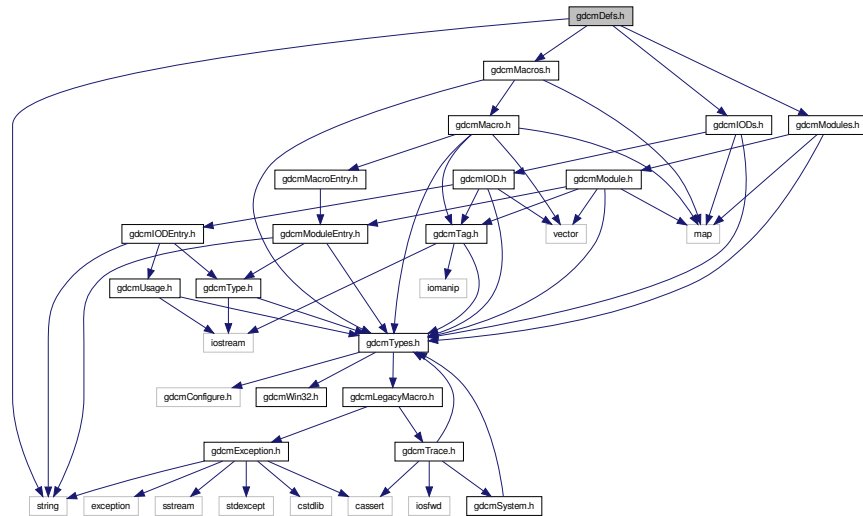
Include dependency graph for gdcmDeflateStream.h:



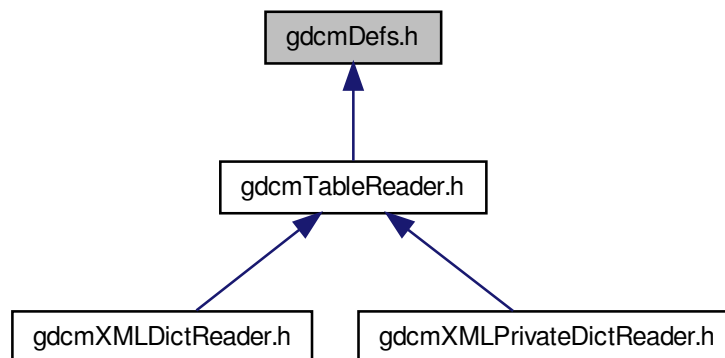
## 28.63 gdcmDefs.h File Reference

```
#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

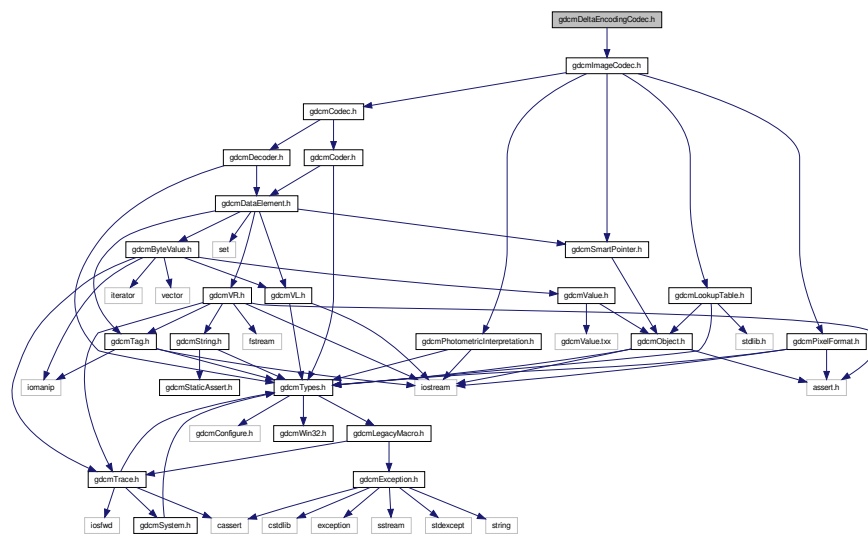
- class [gdcm::Defs](#)

*FIXME I do not like the name 'Defs'.*

- **gdcm**

- **gdcm**

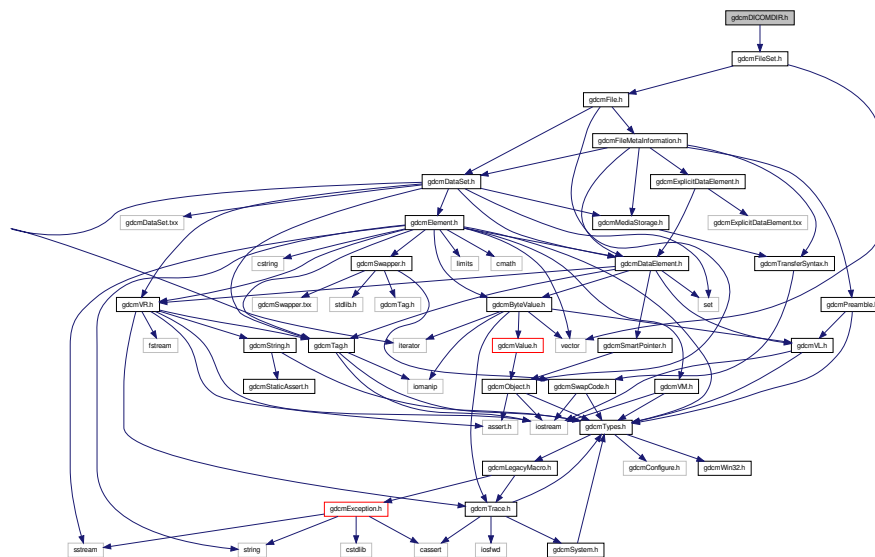
Include dependency graph for gdcMDeltaEncodingCodec.h:



- class `gdcm::DeltaEncodingCodec`

- **gdcm**

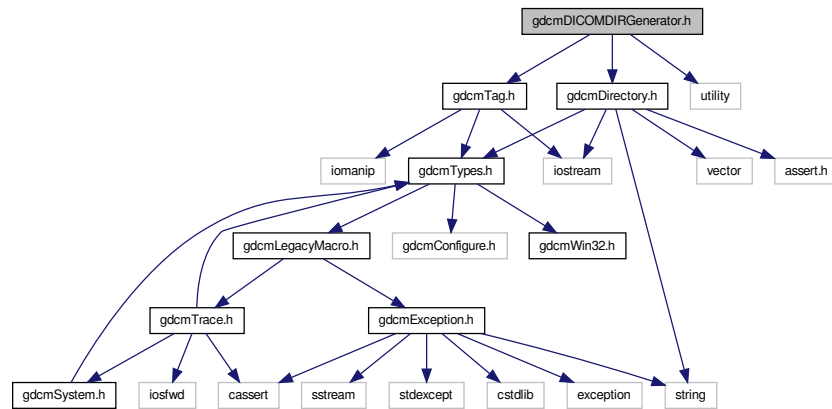
Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen



---



Include dependency graph for gdcmDICOmdirGenerator.h:



## Classes

- class [gdcm::DICOmdirGenerator](#)

*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.

## Namespaces

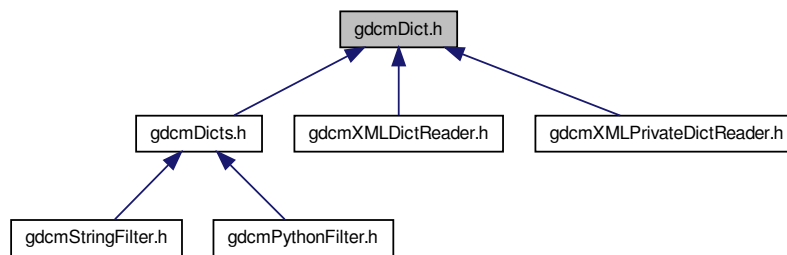
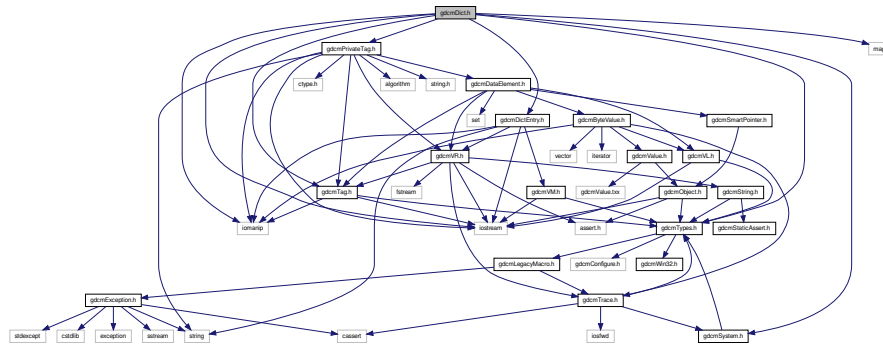
- [gdcm](#)

## 28.67 gdcmDict.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmPrivateTag.h"
#include "gdcmDictEntry.h"
#include "gdcmsystem.h"
#include <iostream>
#include <iomanip>
#include <map>

```



- class `gdcmm::Dict`  
*Class to represent a map of `DictEntry`.*
- class `gdcmm::PrivateDict`  
*Private `Dict`.*

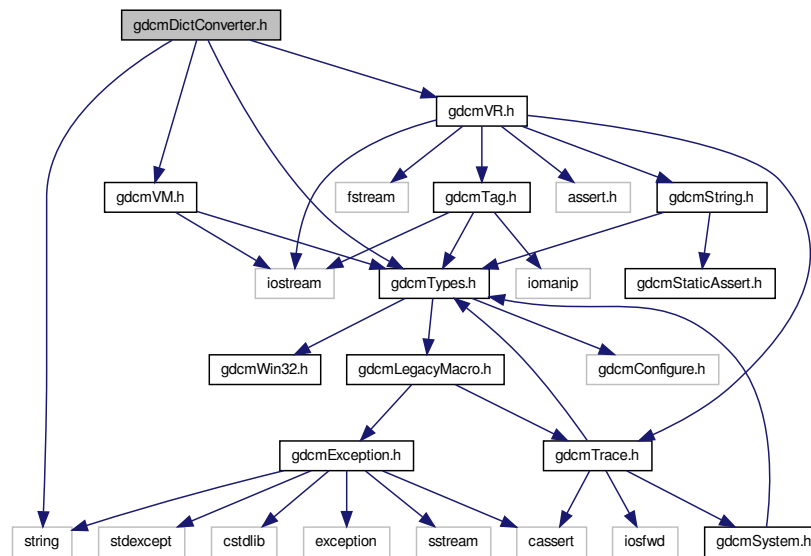
- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Dict &val)`
- `std::ostream & gdcmm::operator<< (std::ostream &os, const PrivateDict &val)`

## 28.68 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:*

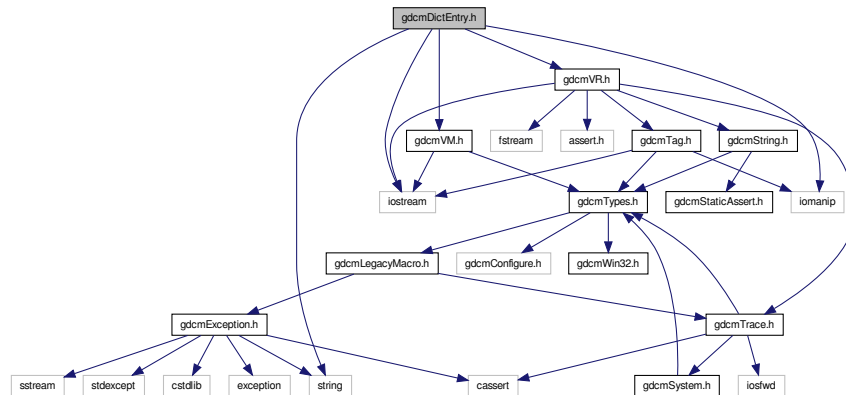
### Namespaces

- [gdcm](#)

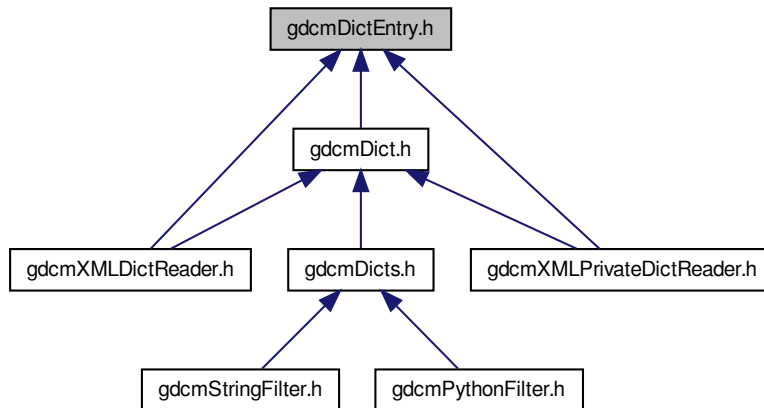
## 28.69 gdcmDictEntry.h File Reference

```
#include "gdcmVR.h"
#include "gdcmVM.h"
#include <string>
#include <iostream>
#include <iomanip>
```

Include dependency graph for `gdcDictEntry.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::DictEntry](#)

*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 [gdc::Tag](#) to the needed information.*

## Namespaces

- [gdc](#)

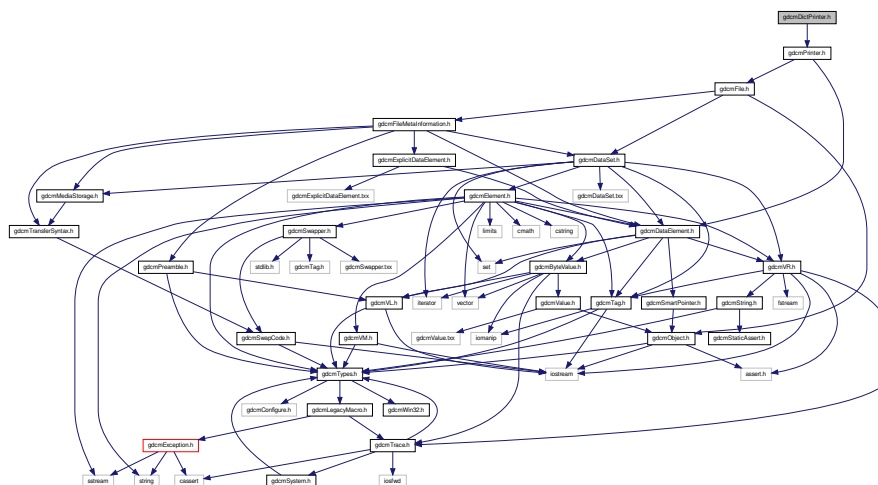
## Functions

- `std::ostream & gdcmm::operator<< (std::ostream &os, const DictEntry &val)`

## 28.70 gdcmDictPrinter.h File Reference

```
#include "gdcmPrinter.h"
```

Include dependency graph for gdcDictPrinter.h:



## Classes

- class `gdcm::DictPrinter`

*DictPrinter* class.

## Namespaces

- gdc

## 28.71 gdcmDicts.h File Reference

```
#include "gdcmDict.h"
```

```
#include "gdcmCSAHeaderDict.h"
```

```
#include <string>
```

```
graph BT; gdcStringFilter.h --> gdcDicts.h; gdcPythonFilter.h --> gdcDicts.h
```

- class `gdcmm::Dicts`  
*Class to manipulate the sum of knowledge (all the dict user load)*

- gdc

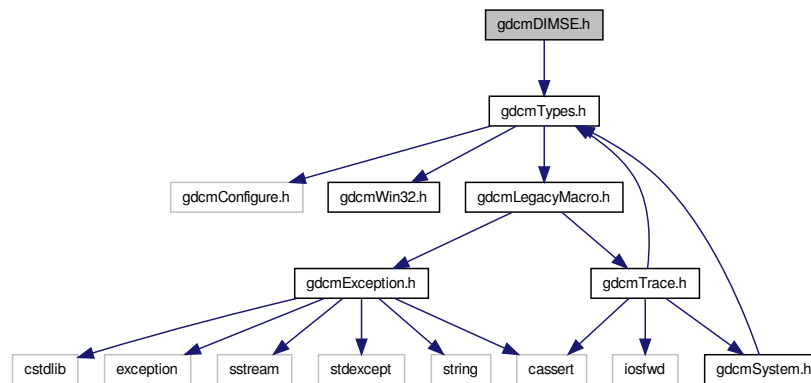
- `std::ostream & gdcmm::operator<< (std::ostream &os, const Dicts &d)`

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen

## 28.73 gdcmDIMSE.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDIMSE.h:



### Classes

- class [gdcm::network::CEchoRQ](#)

*[CEchoRQ](#) this file defines the messages for the cecho action.*

- class [gdcm::network::CEchoRSP](#)

*[CEchoRSP](#) this file defines the messages for the cecho action.*

- class [gdcm::network::CFind](#)

- class [gdcm::network::DIMSE](#)

*[DIMSE](#) PS 3.7 - 2009 Annex E [Command Dictionary \(Normative\)](#) E.1 REGISTRY OF DICOM COMMAND ELEMENTS  
[Table E.1-1 COMMAND FIELDS \(PART 1\)](#)*

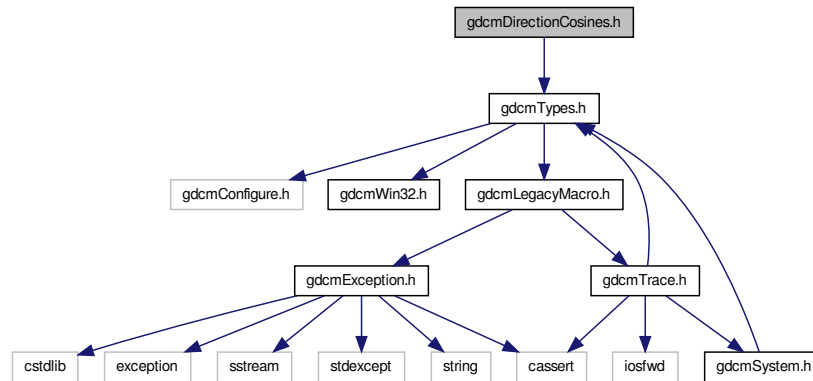
### Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.74 gdcmDirectionCosines.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmDirectionCosines.h`:



## Classes

- class `gdcm::DirectionCosines`  
class to handle `DirectionCosines`

## Namespaces

- `gdcm`

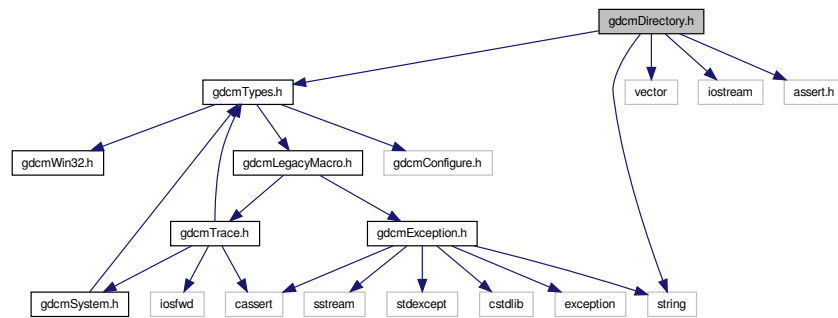
## 28.75 gdcmDirectory.h File Reference

```

#include "gdcmTypes.h"
#include <string>
#include <vector>
#include <iostream>
#include <assert.h>

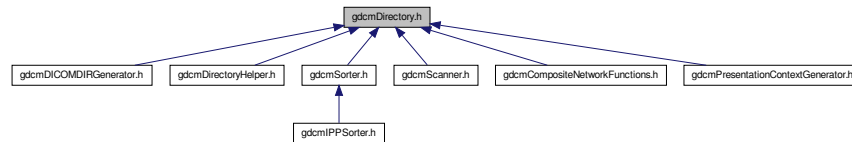
```

Include dependency graph for `gdcmDirectory.h`:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Directory](#)  
Class for manipulation directories.

## Namespaces

- [gdcm](#)

## Functions

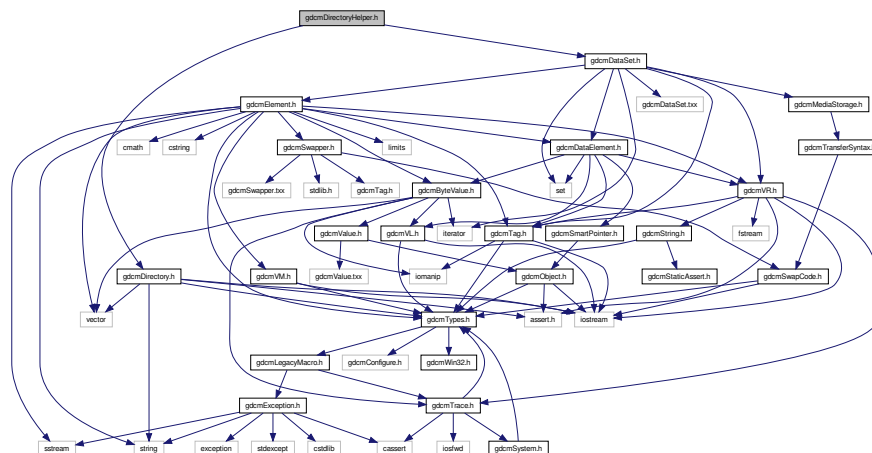
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const Directory &d)

## 28.76 gdcmDirectoryHelper.h File Reference

```
#include "gdcmDirectory.h"
```

```
#include "gdcmDataSet.h"
```

Include dependency graph for gdcmDirectoryHelper.h:



## Classes

- class [gdcm::DirectoryHelper](#)

*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.

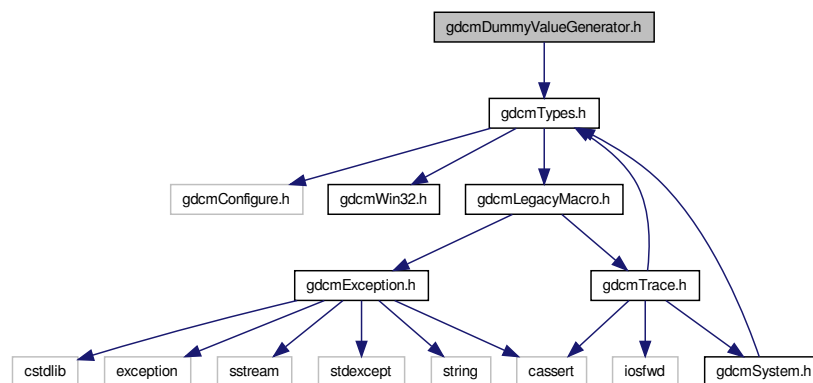
## Namespaces

- [gdcm](#)

## 28.77 gdcmDummyValueGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmDummyValueGenerator.h:



## Classes

- class [gdcm::DummyValueGenerator](#)

*Class for generating dummy value.*

## Namespaces

- [gdcm](#)

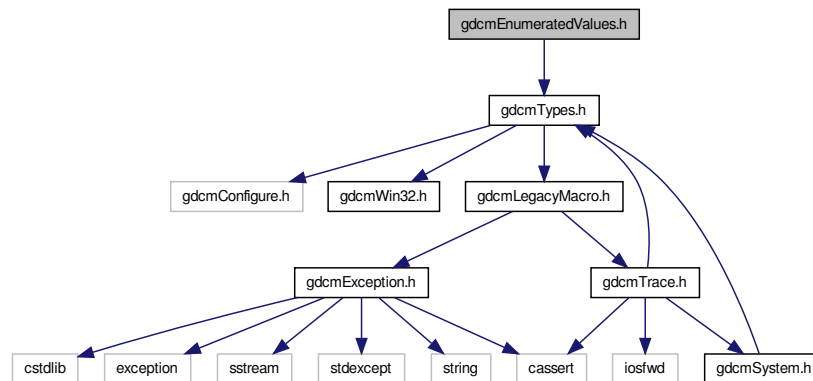
## 28.78 gdcmdump.man File Reference







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:

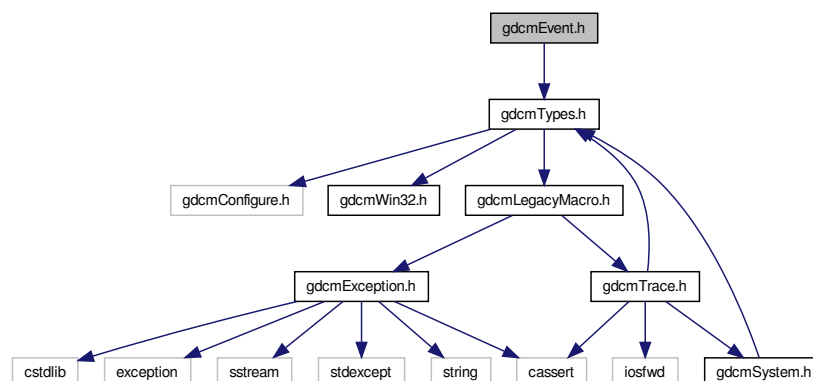
## Namespaces

- [gdcm](#)

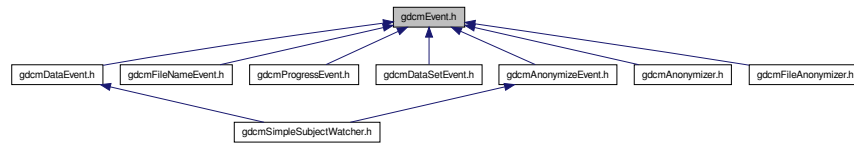
## 28.83 gdcmEvent.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmEvent.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::AbortEvent](#)
- class [gdcm::AnyEvent](#)
- class [gdcm::EndEvent](#)
- class [gdcm::Event](#)  
*superclass for callback/observer methods*
- class [gdcm::ExitEvent](#)
- class [gdcm::InitializeEvent](#)
- class [gdcm::IterationEvent](#)
- class [gdcm::ModifiedEvent](#)
- class [gdcm::NoEvent](#)
- class [gdcm::StartEvent](#)
- class [gdcm::UserEvent](#)

## Namespaces

- [gdc](#)

## Macros

- `#define gdcmEventMacro(classname, super)`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, Event &e)`  
*Generic inserter operator for [Event](#) and its subclasses.*

### 28.83.1 Macro Definition Documentation

#### 28.83.1.1 `#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 ::gdc::Event* e) const \
{ return dynamic_cast<const Self*>(e) ? true : false; } \
virtual ::gdc::Event* MakeObject() const \
{ return new Self; } \
classname(const Self&s) : super(s){}; \
private: \
void operator=(const Self&); \
}

```

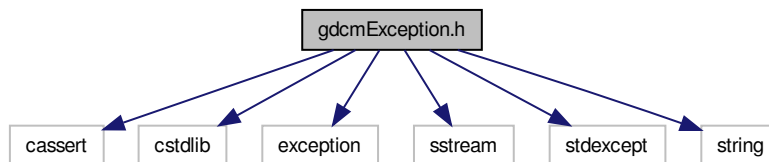
## 28.84 gdcException.h File Reference

```

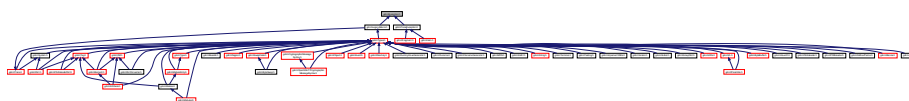
#include <cassert>
#include <cstdlib>
#include <exception>
#include <sstream>
#include <stdexcept>
#include <string>

```

Include dependency graph for gdcException.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdc::Exception](#)  
*Exception.*

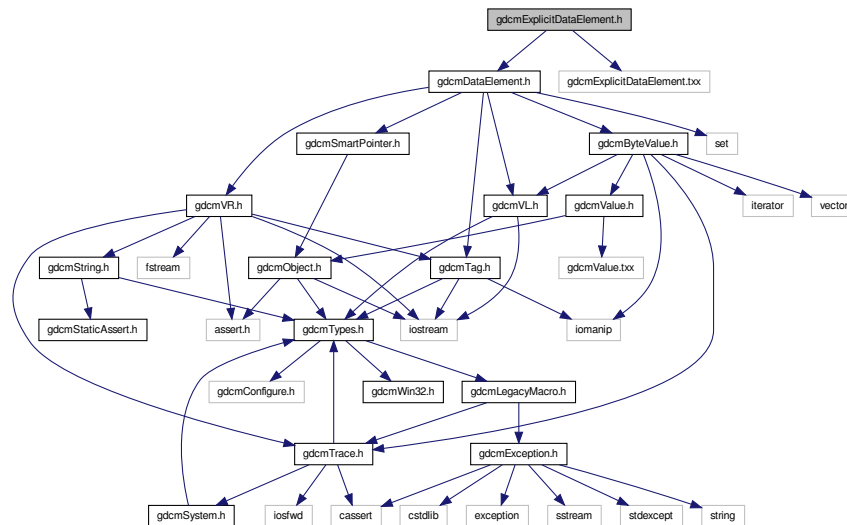
### Namespaces

- [gdc](#)

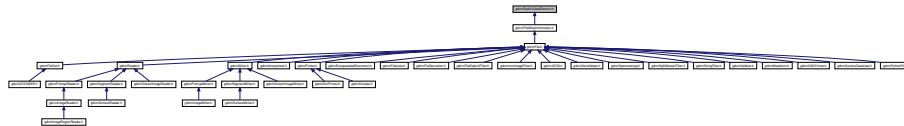


## 28.85 gdcmExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmExplicitDataElement.txx"
Include dependency graph for gdcmExplicitDataElement.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

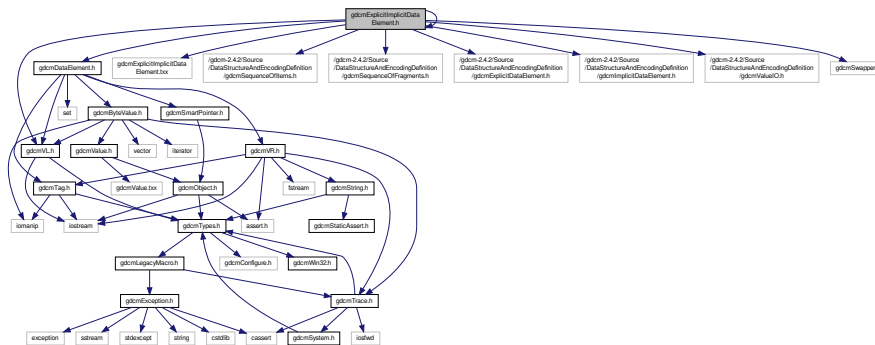
- class [gdcm::ExplicitDataElement](#)  
Class to read/write a [DataElement](#) as *Explicit Data Element*.

### Namespaces

- [gdcm](#)

## 28.86 gdcmExplicitImplicitDataElement.h File Reference

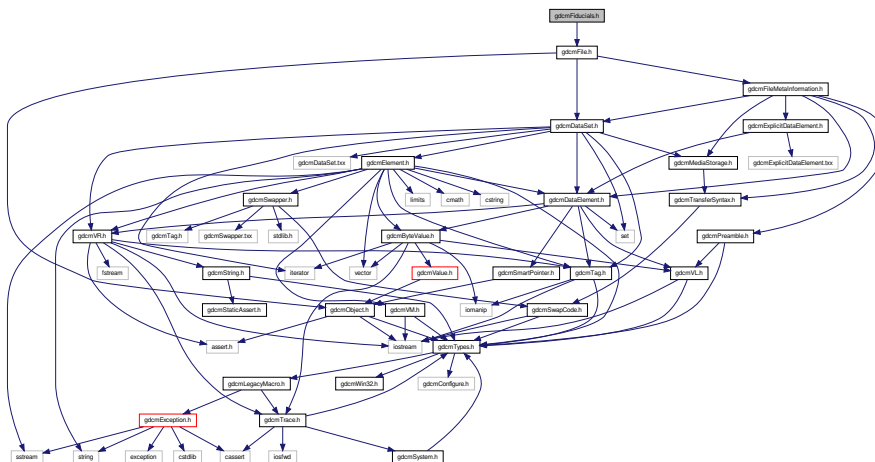
```
#include "gdcmDataElement.h"
#include "gdcmExplicitImplicitDataElement.txx"
```



- class `gdcmm::ExplicitImplicitDataElement`  
*Class to read/write a `DataElement` as ExplicitImplicit Data `Element`.*

- gdc

Include dependency graph for g



## Classes

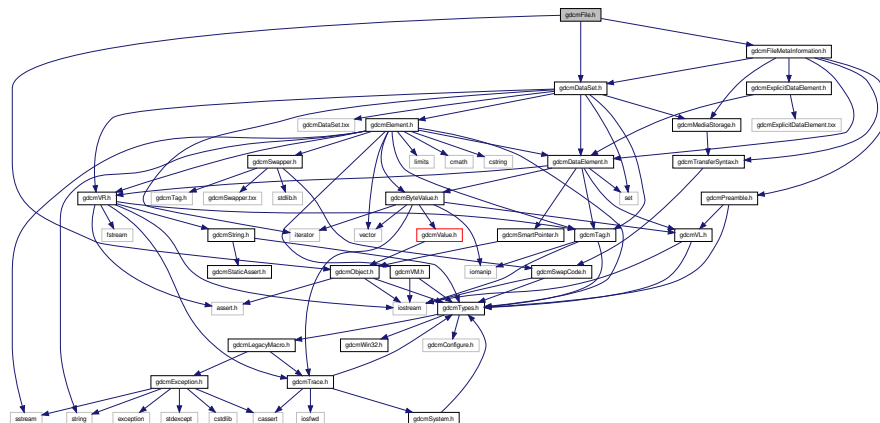
- class `gdcm::Fiducials`  
*Fiducials*.

## Namespaces

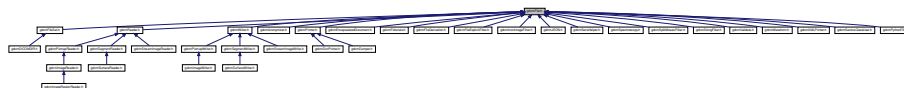
- **gdcm**

## 28.88 gdcmFile.h File Reference

```
#include "gdcmObject.h"
#include "gdcmDataSet.h"
#include "gdcmFileMetaInformation.h"
Include dependency graph for gdcmFile.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::File`

a DICOM [File](#) 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.

## Namespaces

- **gdcm**

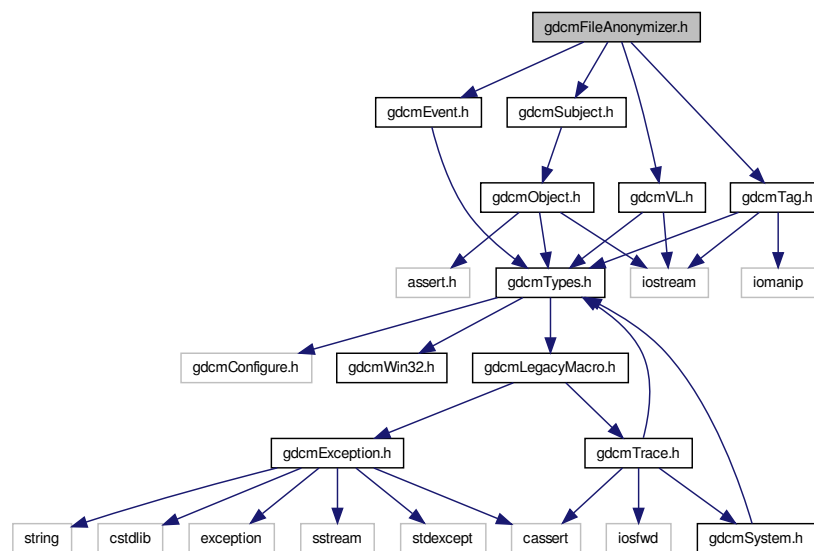
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const File &val)`

## 28.89 `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.*

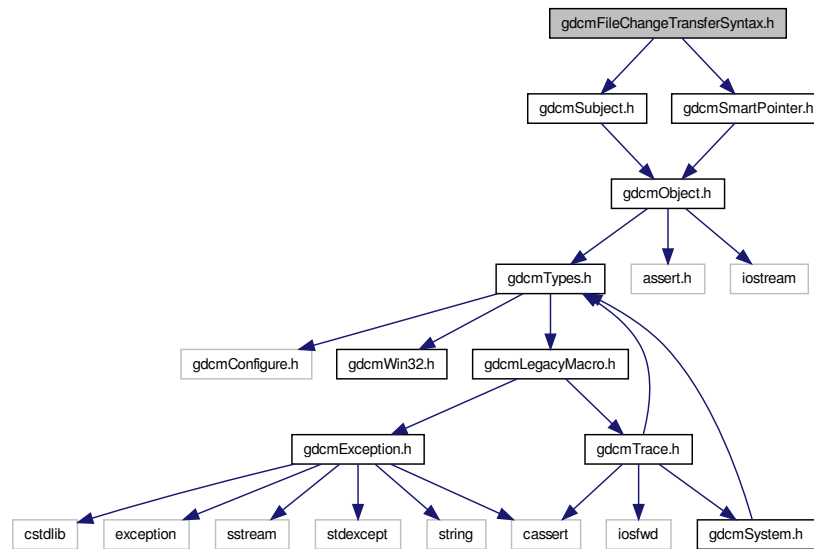
## Namespaces

- `gdcm`

## 28.90 `gdcmFileChangeTransferSyntax.h` File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
```

Include dependency graph for gdcmFileChangeTransferSyntax.h:



## Classes

- class [gdcm::FileChangeTransferSyntax](#)

*FileChangeTransferSyntax.*

## Namespaces

- [gdcm](#)

## 28.91 gdcmFileDerivation.h File Reference

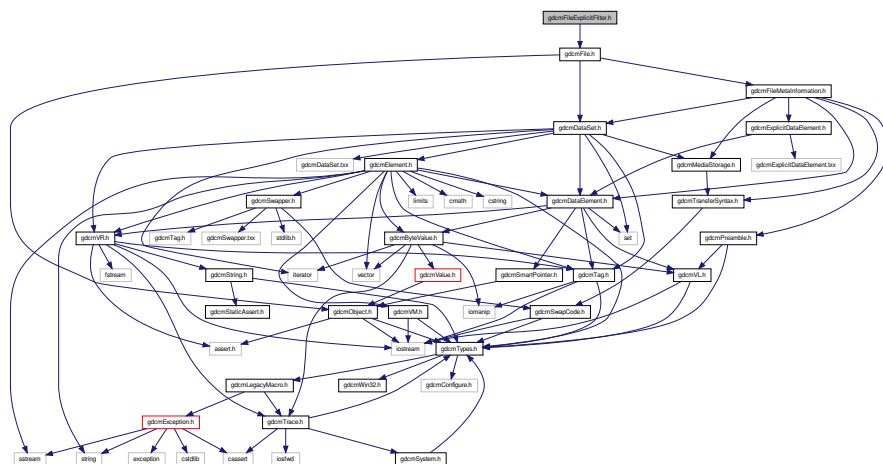
```
#include "gdcmFile.h"
```

- class `gdcm::FileDerivation`

## Namespaces

- ## 28.92 gdcmFileExplicitFilter.h File Reference

Include dependency graph for `gdcmFileExplicitFilter.h`:





## Namespaces

- [gdcm](#)

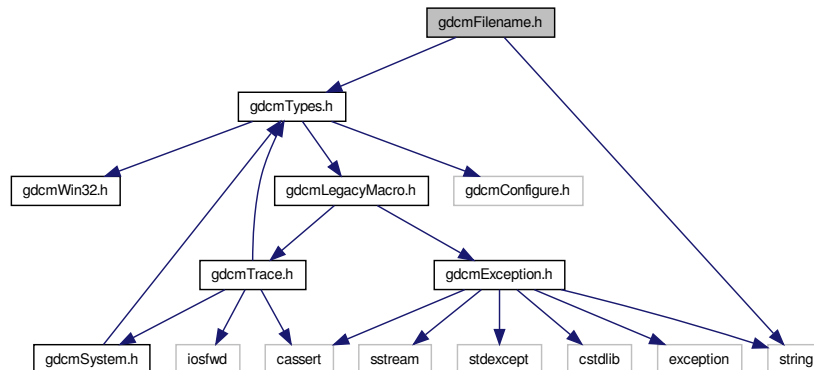
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const FileMetaInformation &val)`

## 28.94 [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

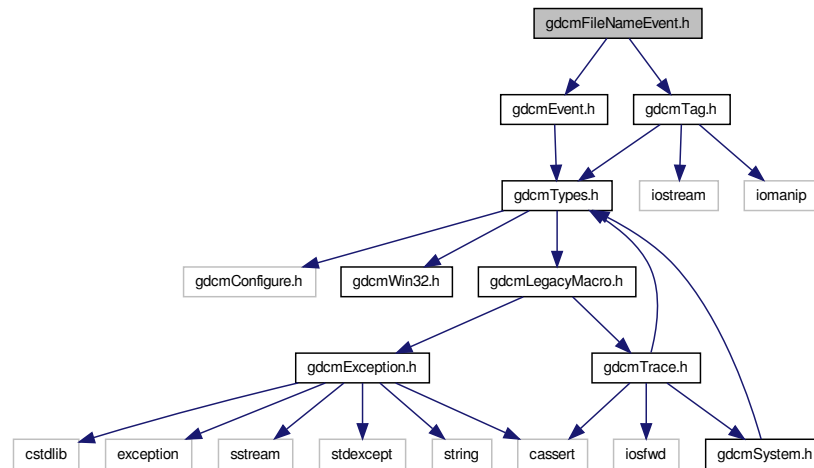
- [gdcm](#)

## 28.95 [gdcmFileNameEvent.h](#) File Reference

```
#include "gdcmEvent.h"
#include "gdcmTag.h"
```



Include dependency graph for gdcmFileNameEvent.h:



## Classes

- class [gdcm::FileNameEvent](#)

*FileNameEvent* Special type of event triggered during processing of *FileSet*.

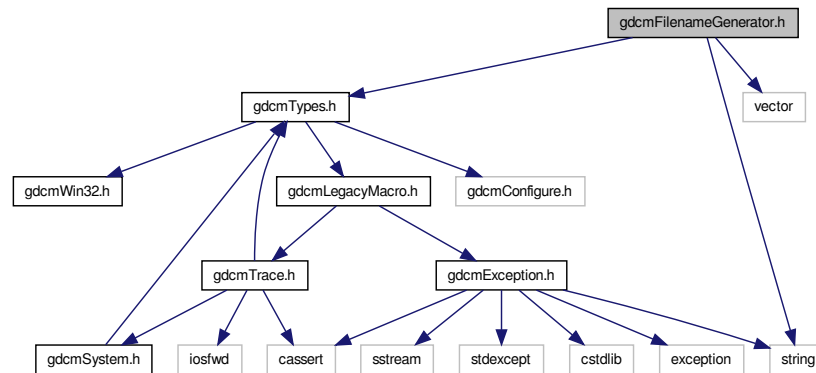
## Namespaces

- [gdcm](#)

## 28.96 gdcmFilenameGenerator.h File Reference

```
#include "gdcmTypes.h"
#include <string>
#include <vector>
```

Include dependency graph for `gdcmlFilenameGenerator.h`:



## Classes

- class `gdcml::FilenameGenerator`

*FilenameGenerator.*

## Namespaces

- `gdcml`

## 28.97 gdcmlFileSet.h File Reference

```
#include "gdcmlFile.h"
#include <vector>
```

```
graph BT; gdcmDICOMDIR.h --> gdcmFileSet.h
```

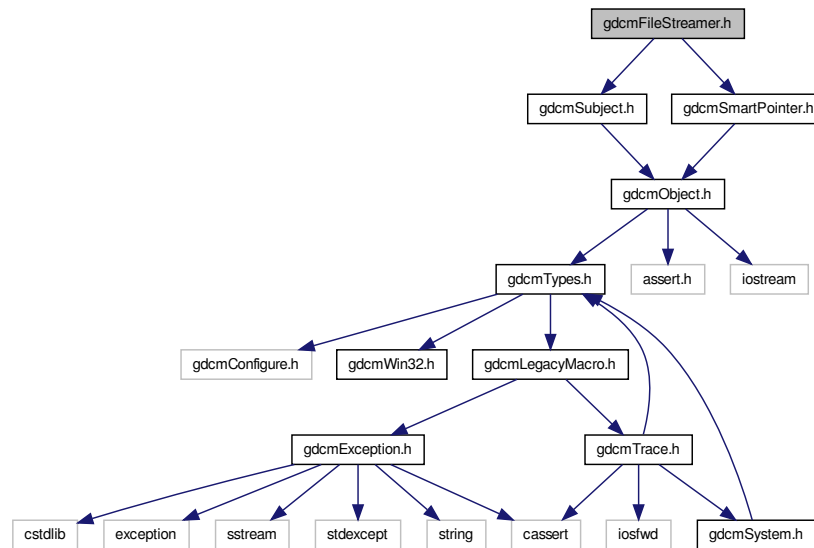
- class `gdcm::FileSet`

- gdc

- `std::ostream & gdcm::operator<< (std::ostream &os, const FileSet &f)`

## 28.98 gdcmFileStreamer.h File Reference

```
#include "gdcmSubject.h"
#include "gdcmSmartPointer.h"
Include dependency graph for gdcmFileStreamer.h:
```



### Classes

- class [gdcm::FileStreamer](#)

*[FileStreamer](#) This class let a user create a massive DICOM [DataSet](#) from a template DICOM file, by appending chunks of data.*

### Namespaces

- [gdcm](#)

## 28.99 gdcmFindPatientRootQuery.h File Reference

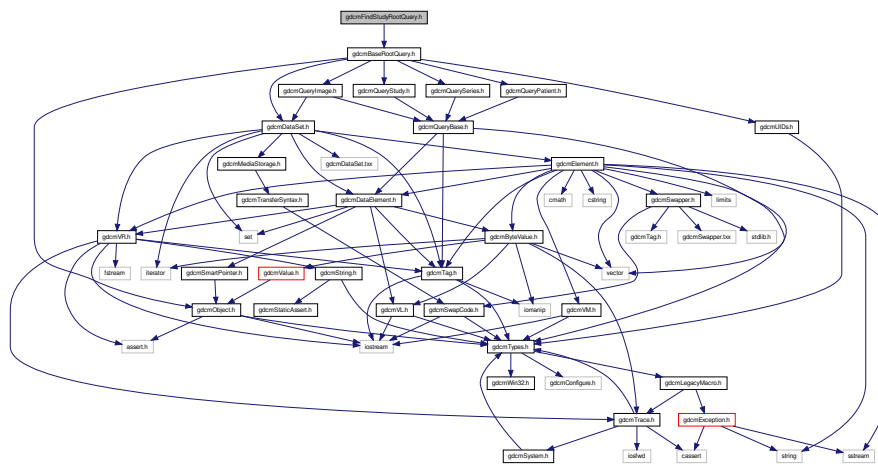
```
#include "gdcmBaseRootQuery.h"
```

```
graph BT
    A[gdcmFindPatientRootQuery.h] --> B[gdcmMovePatientRootQuery.h]
```

- class `gdcm::FindPatientRootQuery`

- **gdcm**

```
#include "gdcmBaseRootQuery.h"
```

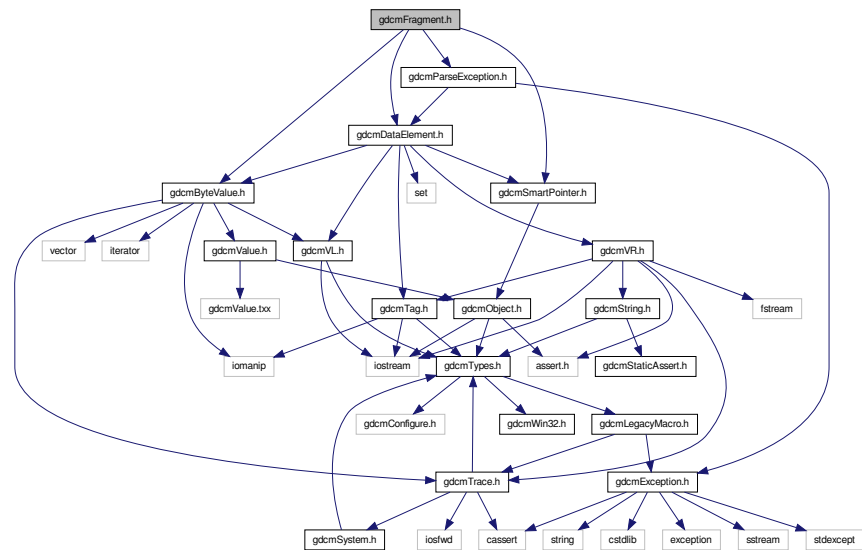


- class `gdcm::FindStudyRootQuery`

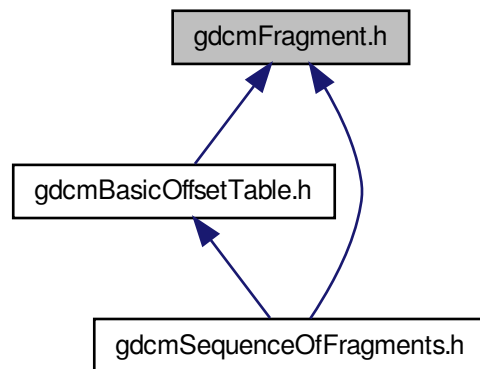
- **gdcm**

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen

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

- [gdc](#)

## Functions

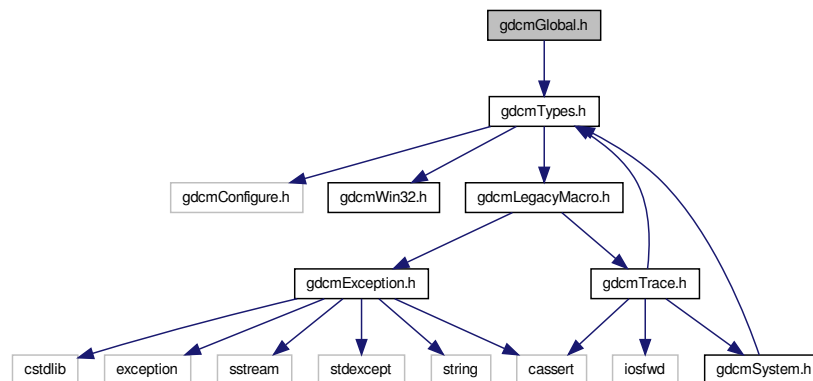
- `std::ostream & gdc::operator<< (std::ostream &os, const Fragment &val)`

## 28.102 gdcmgendir.man File Reference

## 28.103 gdcGlobal.h File Reference

```
#include "gdcTypes.h"
```

Include dependency graph for gdcGlobal.h:



## Classes

- class [gdc::Global](#)  
*Global.*

## Namespaces

- [gdc](#)

## Functions

- `std::ostream & gdc::operator<< (std::ostream &os, const Global &g)`

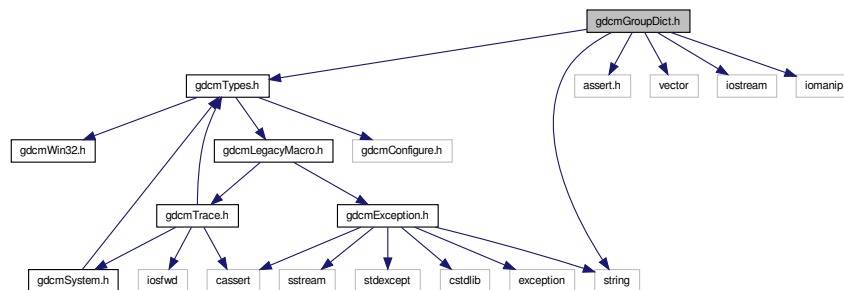
## Variables

- static Global [gdc::GlobalInstance](#)



## 28.104 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](#)

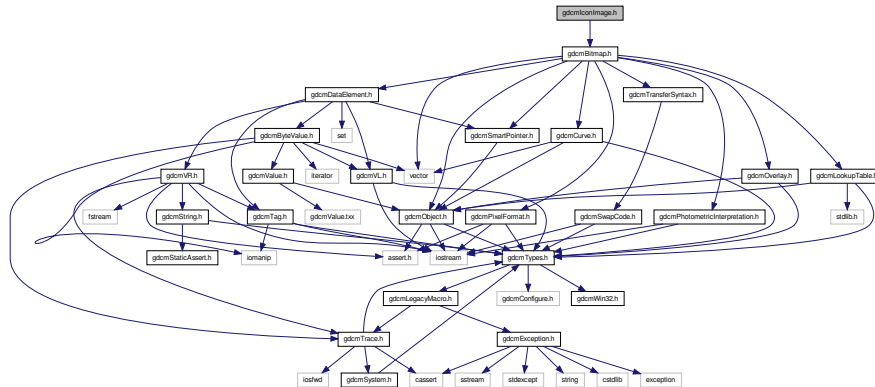
### Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const GroupDict &_val)`

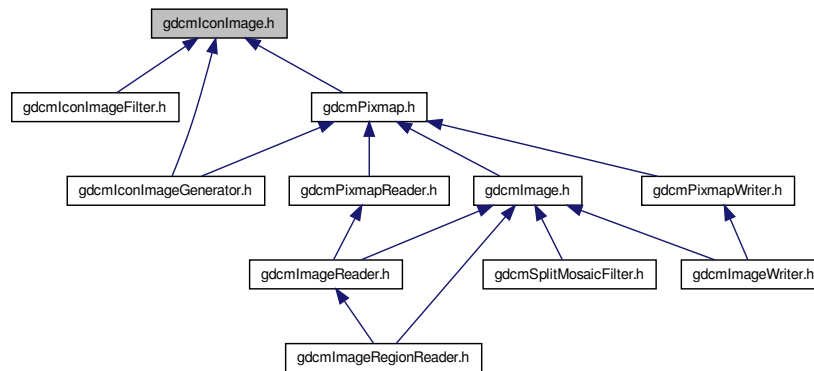
## 28.105 gdcmIconImage.h File Reference

```
#include "gdcmBitmap.h"
```

Include dependency graph for `gdcmIconImage.h`:



This graph shows which files directly or indirectly include this file:



## Namespaces

- [gdcm](#)

## Typedefs

- typedef Bitmap [gdcm::IconImage](#)

## 28.106 gdcmIconImageFilter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmIconImage.h"
```

- class `gdcm::IconImageFilter`

## Namespaces

- **gdcm**

```
#include "gdcmPixmap.h"
#include "gdcmIconImage.h"
```



[illegible]

```

graph TD
    gdcmImageReader[gdcmImageReader.h] --> gdcmImage[gdcmImage.h]
    gdcmImageWriter[gdcmImageWriter.h] --> gdcmImage
    gdcmSplitMosaicFilter[gdcmSplitMosaicFilter.h] --> gdcmImage
    gdcmImage --> gdcmImageRegionReader[gdcmImageRegionReader.h]
    gdcmImage --> gdcmImageReader
  
```

- class `gdcm::Image`

## Namespaces

- ## 28.109 gdcmlImageApplyLookupTable.h File Reference

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen



[illegible]

- class `gdcm::ImageChangePhotometricInterpretation`

*ImageChangePhotometricInterpretation* class Class to change the Photometric Interpretation of an input DICOM.

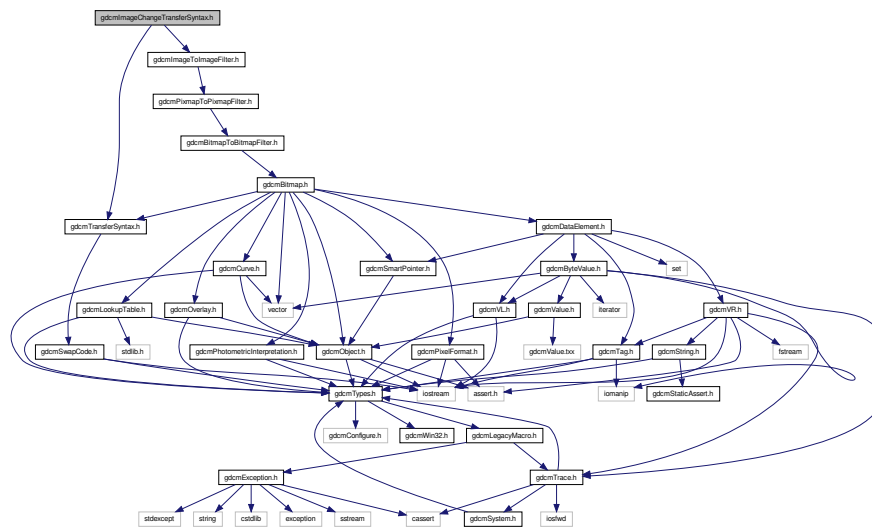
- **gdcm**

```
#include "gdcmImageToImageFilter.h"
```





Include dependency graph for gdcmImageChangeTransferSyntax.h:



## Classes

- class [gdcm::ImageChangeTransferSyntax](#)

*ImageChangeTransferSyntax* class Class to change the transfer syntax of an input DICOM.

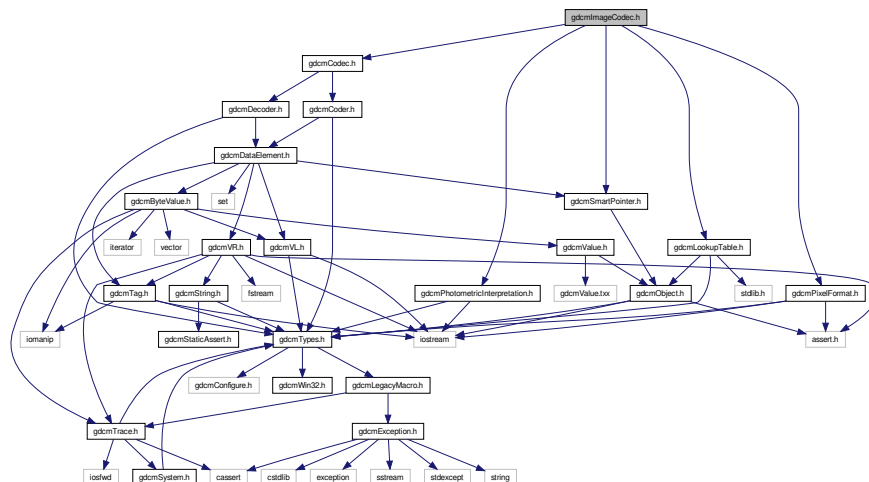
## Namespaces

- [gdcm](#)

## 28.113 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.*

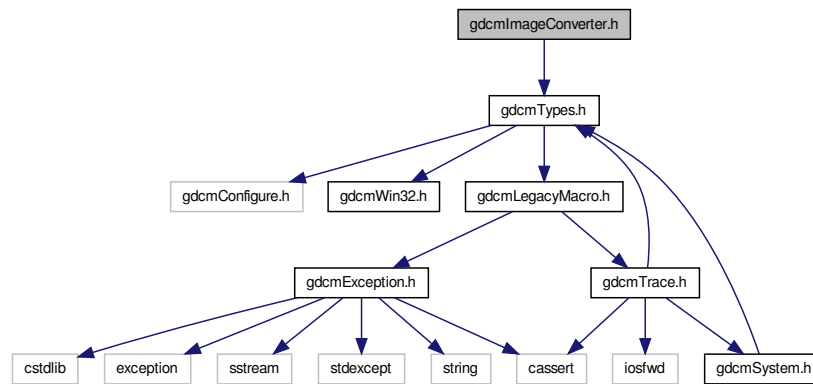
## Namespaces

- `gdcm`

## 28.114 gdcmImageConverter.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImageConverter.h:



## Classes

- class [gdcm::ImageConverter](#)

*Image Converter.*

## Namespaces

- [gdcm](#)

## 28.115 gdcmImageFragmentSplitter.h File Reference

```
#include "gdcmImageToImageFilter.h"
```

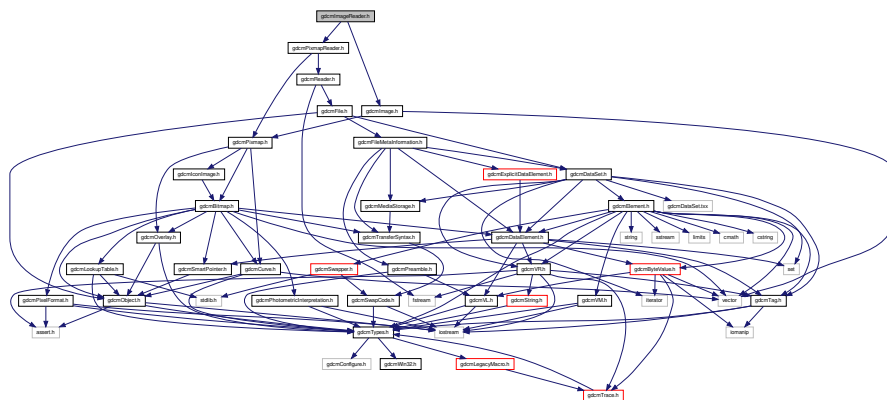


[illegible]

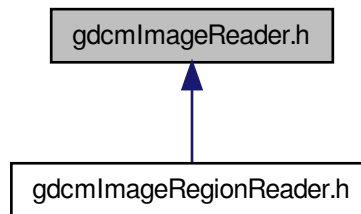
- class `gdcm::ImageHelper`  
*ImageHelper* (internal class, not intended for user level)

- **gdcm**

```
#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 [gdcml::ImageReader](#)

*[ImageReader](#)*.

## Namespaces

- [gdcml](#)

## 28.118 gdcmlImageRegionReader.h File Reference

```
#include "gdcmlImageReader.h"  
#include "gdcmlImage.h"  
#include "gdcmlRegion.h"
```

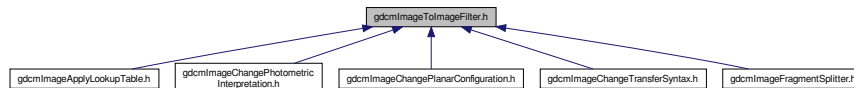
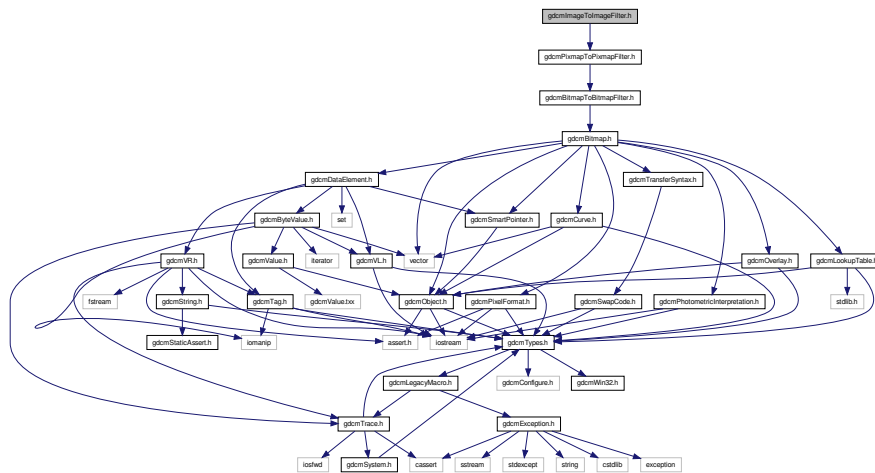
[illegible]

- class `gdcm::ImageRegionReader`

## Namespaces

- **gdcm**

```
#include "gdcmPixmapToPixmapFilter.h"
```



- class `gdcm::ImageToImageFilter`

- **gdcm**

```
#include "gdcmPixmapWriter.h"
#include "gdcmImage.h"
```

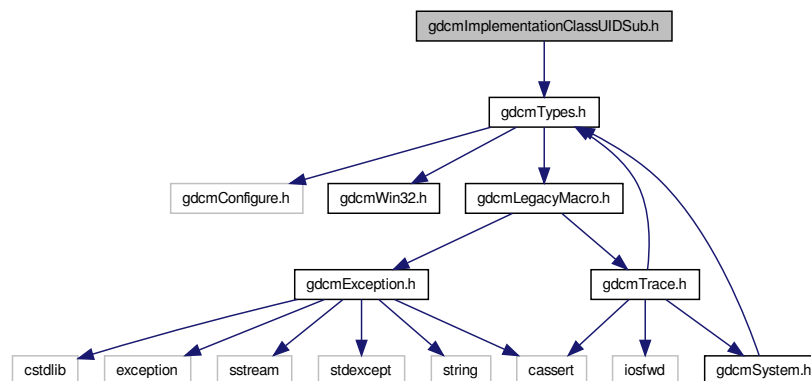


- class `gdcm::ImageWriter`  
*ImageWriter*.

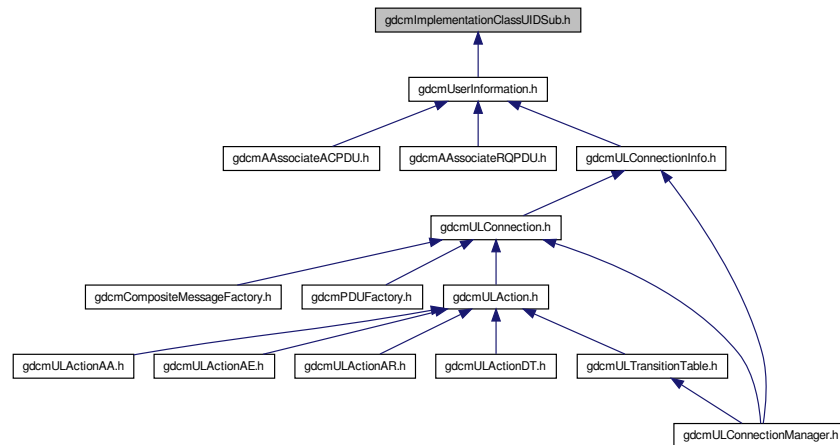
- **gdcm**

## 28.122 gdcmlImplementationClassUIDSub.h File Reference

Include dependency graph for `gdcmlImplementationClassUIDSub.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ImplementationClassUIDSub](#)

[ImplementationClassUIDSub](#) PS 3.7 Table D.3-1 IMPLEMENTATION CLASS UID SUB-ITEM FIELDS (A-ASSOCIATE↔RQ)

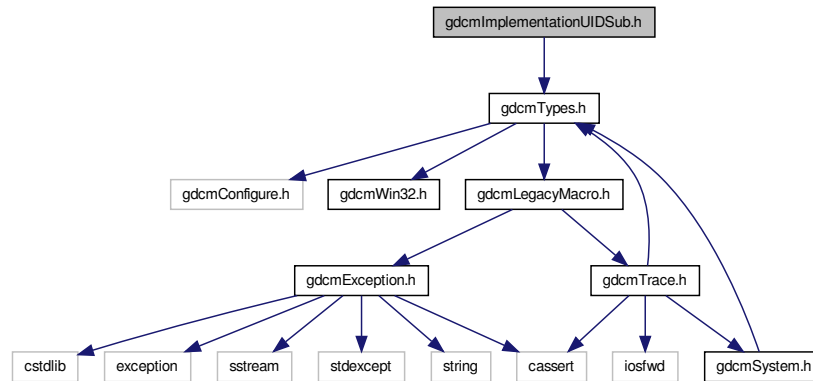
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.123 gdcmImplementationUIDSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmImplementationUIDSub.h:



## Classes

- class [gdcm::network::ImplementationUIDSub](#)

*ImplementationUIDSub Table D.3-2 IMPLEMENTATION UID SUB-ITEM FIELDS (A-ASSOCIATE-AC)*

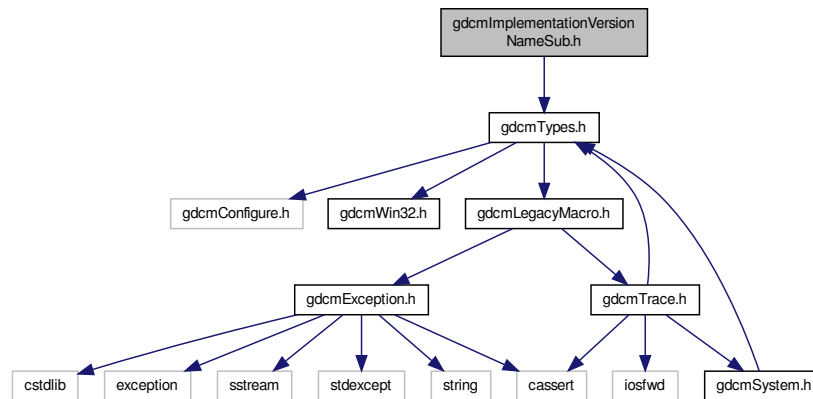
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

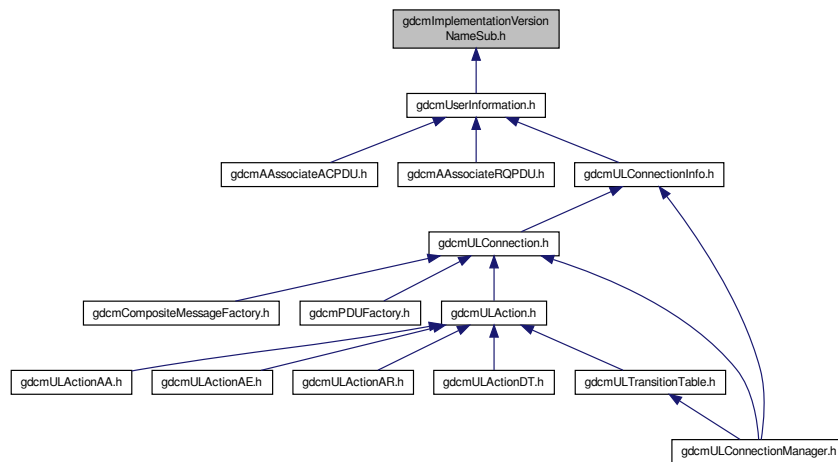
## 28.124 gdcmImplementationVersionNameSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmImplementationVersionNameSub.h`:



This graph shows which files directly or indirectly include this file:



## Classes

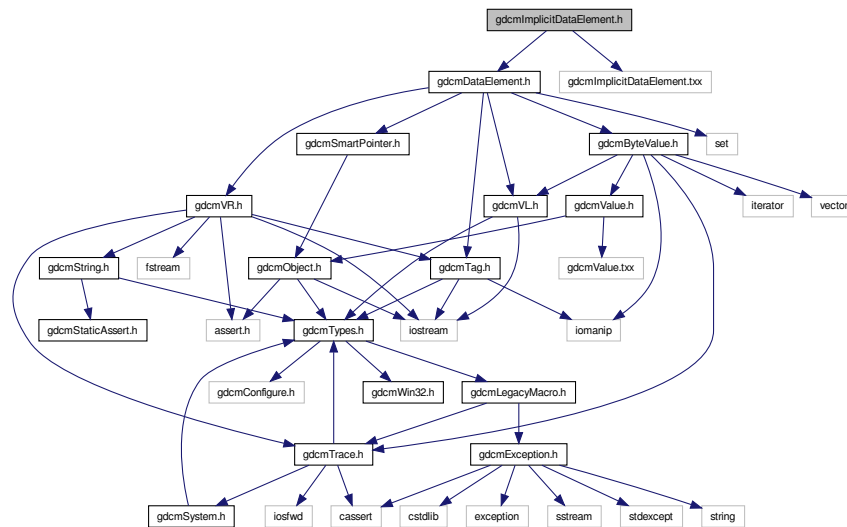
- class `gdcm::network::ImplementationVersionNameSub`  
*ImplementationVersionNameSub* Table D.3-3 IMPLEMENTATION VERSION NAME SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

## Namespaces

- `gdcm`
- `gdcm::network`

## 28.125 gdcmImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmImplicitDataElement.txx"
Include dependency graph for gdcmImplicitDataElement.h:
```



### Classes

- class [gdcm::ImplicitDataElement](#)

*Class to represent an Implicit VR Data Element.*

### Namespaces

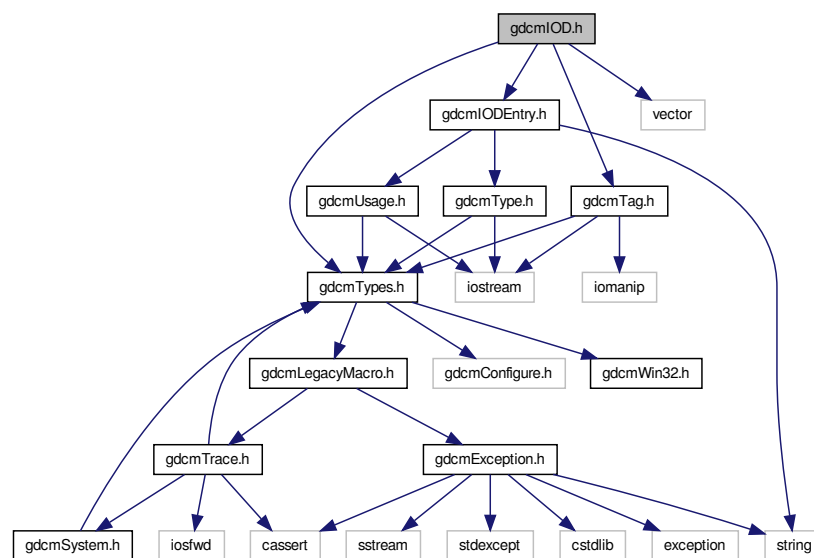
- [gdcm](#)

## 28.126 gdcminfo.man File Reference

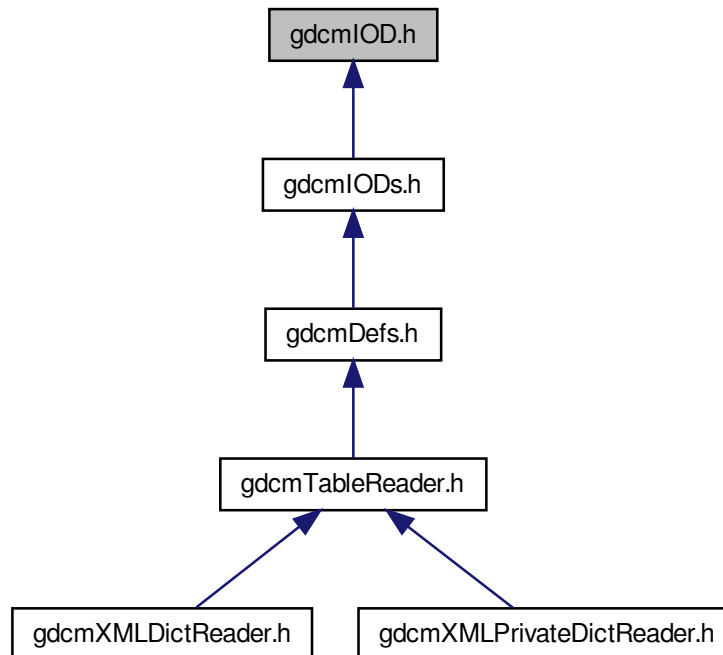
## 28.127 gdcmIOD.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmIODEntry.h"
#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](#)

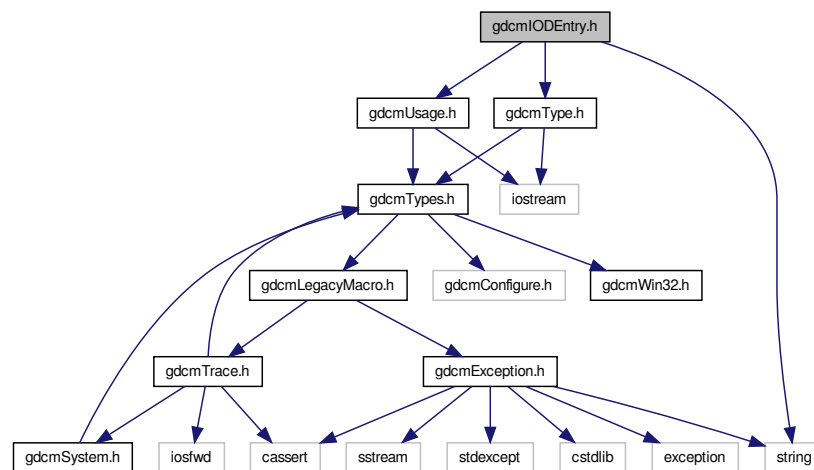
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IOD &_val)`

## 28.128 gdcmIODEntry.h File Reference

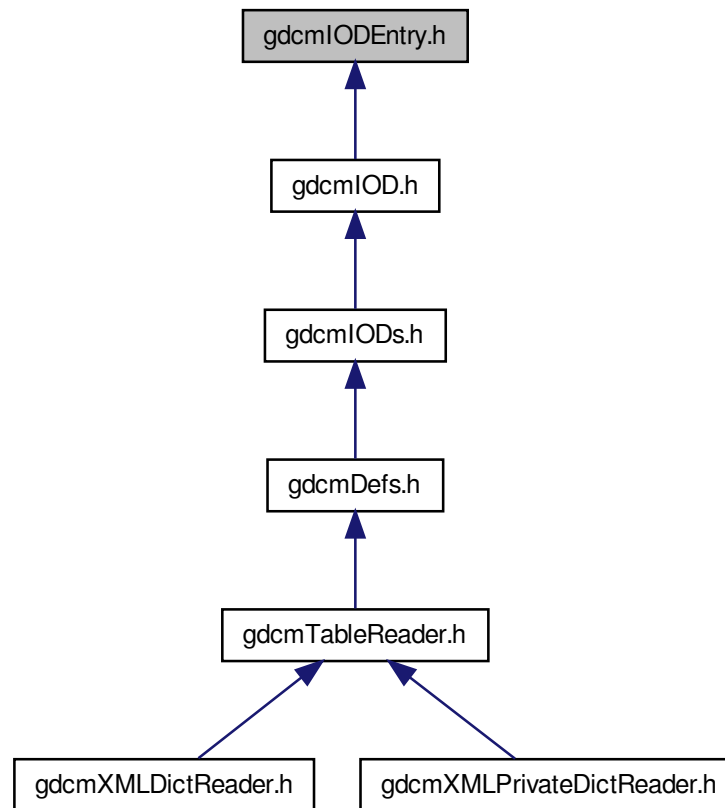
```
#include "gdcmUsage.h"  
#include "gdcmType.h"  
#include <string>
```

Include dependency graph for `gdcmIODEntry.h`:





This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::IODEntry](#)  
*Class for representing a [IODEntry](#).*

## Namespaces

- [gdcm](#)

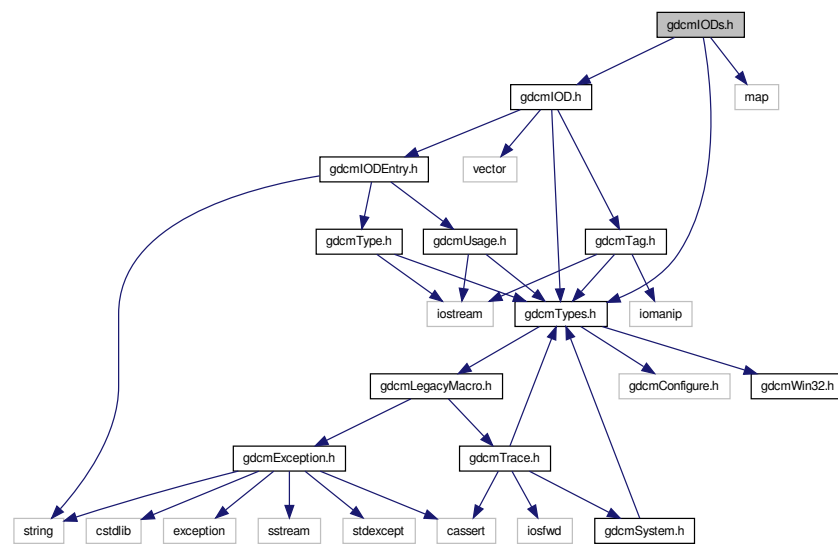
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODEntry &_val)`

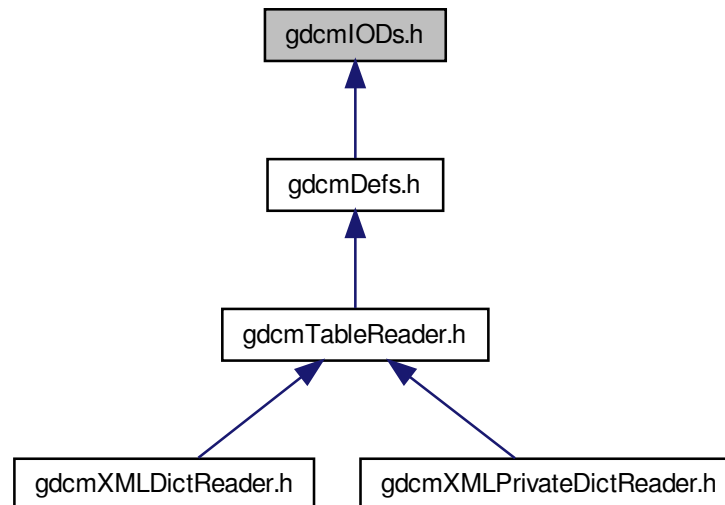
## 28.129 gdcmIODs.h File Reference

```
#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`

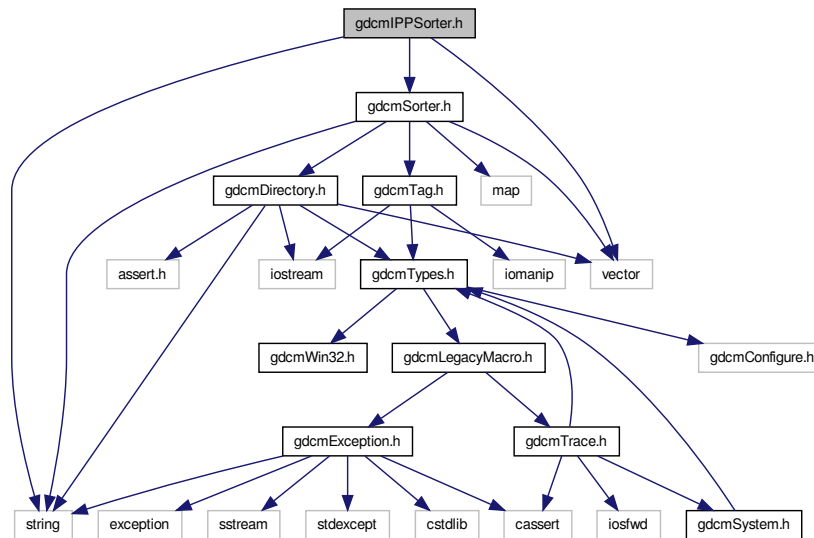
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const IODs &_val)`

## 28.130 gdcmIIPSorter.h File Reference

```
#include "gdcmSorter.h"  
#include <vector>  
#include <string>
```

Include dependency graph for `gdcmIPPSorter.h`:



## Classes

- class [gdcm::IPPSorter](#)

*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.

## Namespaces

- [gdcm](#)

## 28.131 gdcmItem.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmDataSet.h"
#include "gdcmParseException.h"
#include "gdcmSwapper.h"
#include "gdcmItem.txx"

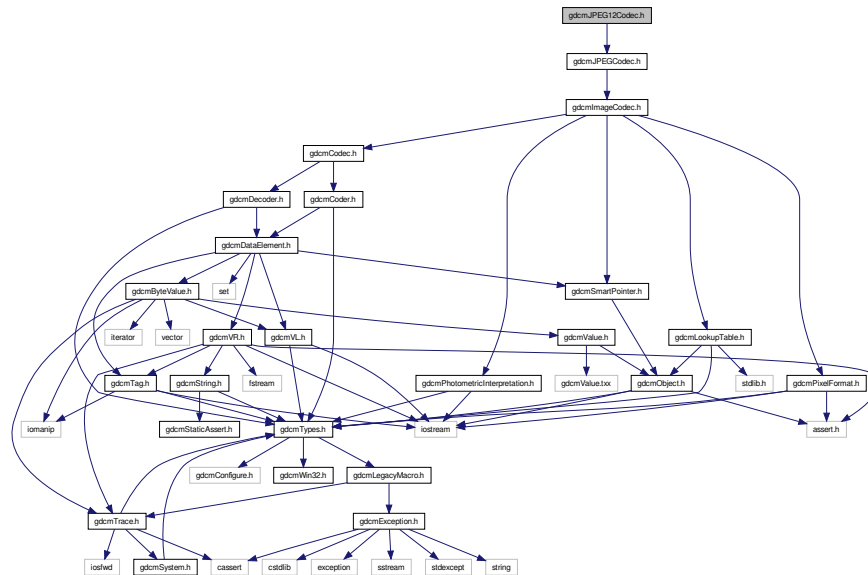
```



## 28.132 gdcmJPEG12Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

Include dependency graph for gdcmJPEG12Codec.h:



## Classes

- class [gdcm::JPEG12Codec](#)

*Class to do JPEG 12bits (lossy & lossless)*

## Namespaces

- [gdcm](#)

## 28.133 gdcmJPEG16Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```

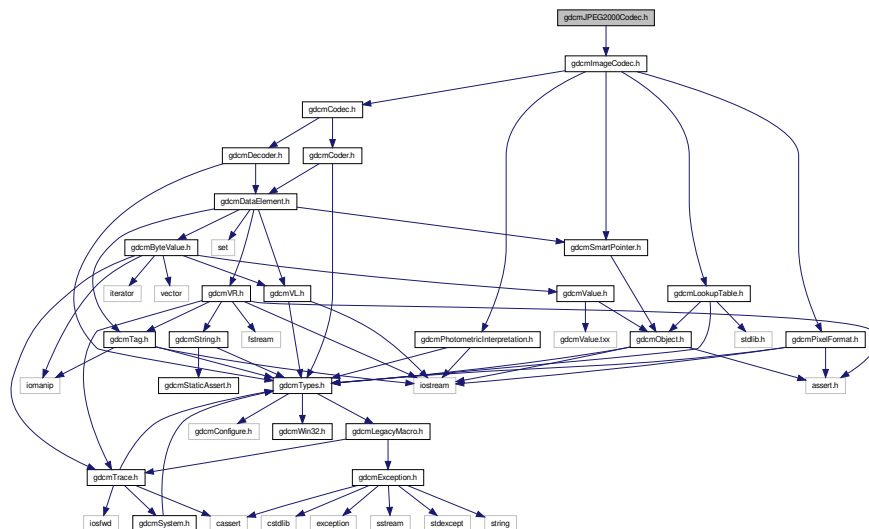
- class `gdcm::JPEG16Codec`

## Namespaces

- ## 28.134 gdcMJPEG2000Codec.h File Reference

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen

Include dependency graph for `gdcmJPEG2000Codec.h`:



## Classes

- class [gdcm::JPEG2000Codec](#)

*Class to do JPEG 2000.*

## Namespaces

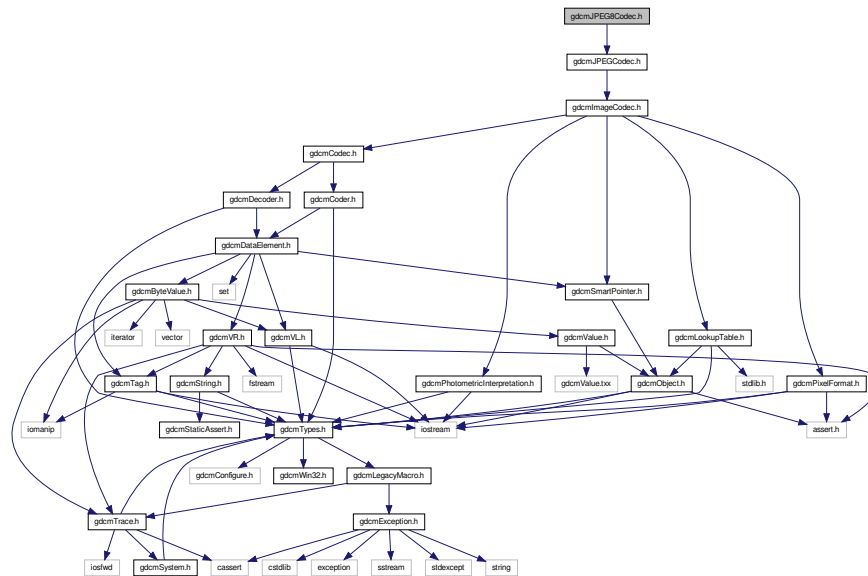
- [gdcm](#)

## 28.135 gdcmJPEG8Codec.h File Reference

```
#include "gdcmJPEGCodec.h"
```



Include dependency graph for gdcmJPEG8Codec.h:



## Classes

- class [gdcm::JPEG8Codec](#)

*Class to do JPEG 8bits (lossy & lossless)*

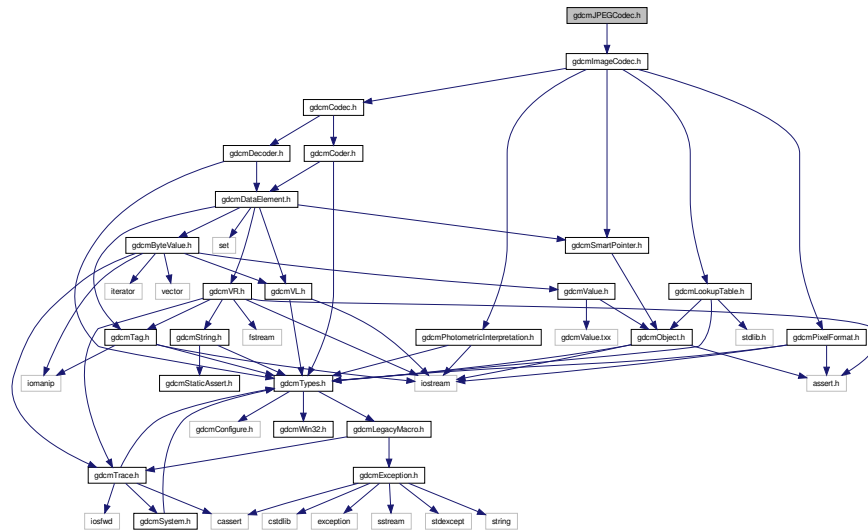
## Namespaces

- [gdcm](#)

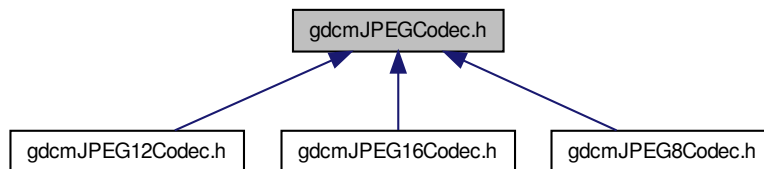
## 28.136 gdcmJPEGCodec.h File Reference

```
#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 Class to do JPEG (8bits, 12bits, 16bits lossy & lossless). It redispach in between the different codec implementation: [gdcm::JPEG8Codec](#), [gdcm::JPEG12Codec](#) & [gdcm::JPEG16Codec](#) It also support inconsistency in between DICOM header and JPEG compressed stream [ImageCodec](#) implementation for the JPEG case.*

## Namespaces

- [gdcm](#)

## 28.137 gdcmJPEGLSCodec.h File Reference

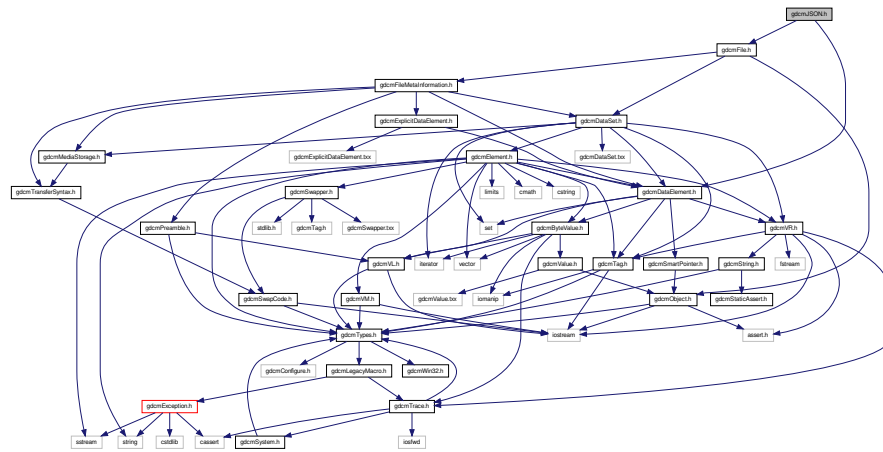
```
#include "gdcmImageCodec.h"
```

- class `gdcm::JPEGLSCodec`

## Namespaces

- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```



- class `gdcm::JSON`

- class `gdc::JSON`

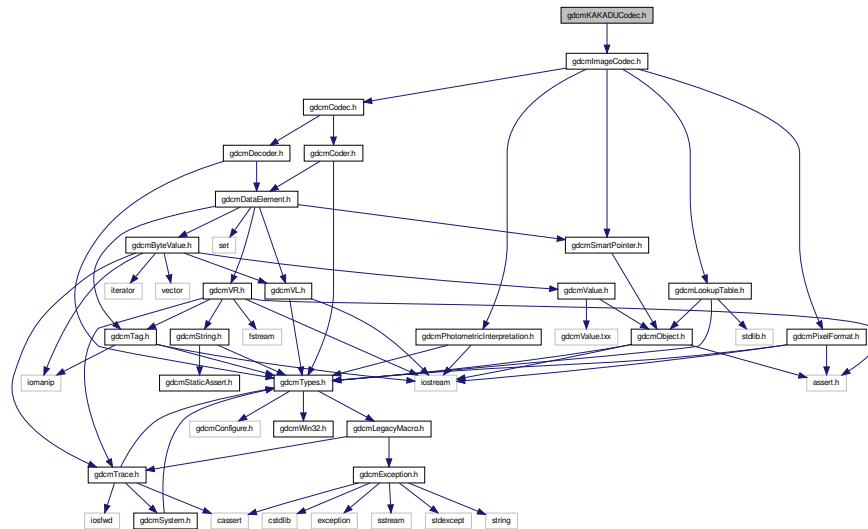
- **gdc**

- **gdcm**

```
#include "gdcmImageCodec.h"
```

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmKAKADUCodec.h:



## Classes

- class [gdcm::KAKADUCodec](#)

*KAKADUCodec.*

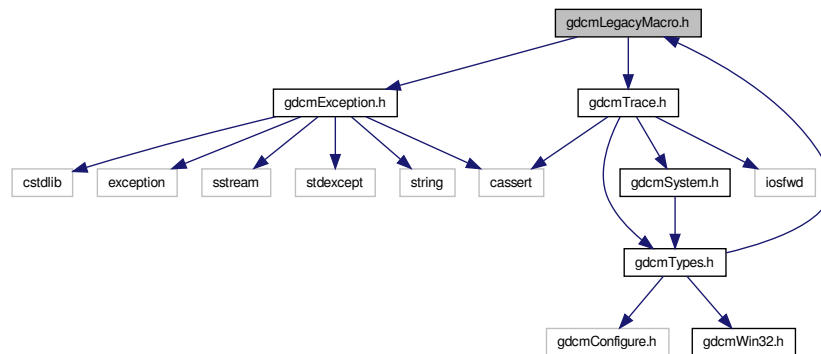
## Namespaces

- [gdcm](#)

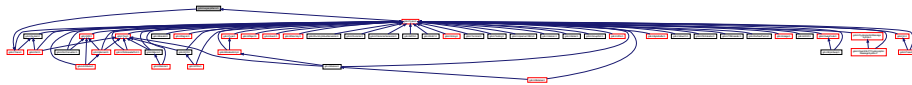
## 28.140 gdcmLegacyMacro.h File Reference

```
#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.")`

### 28.140.1 Macro Definition Documentation

28.140.1.1 `#define GDCM_LEGACY( method ) method;`

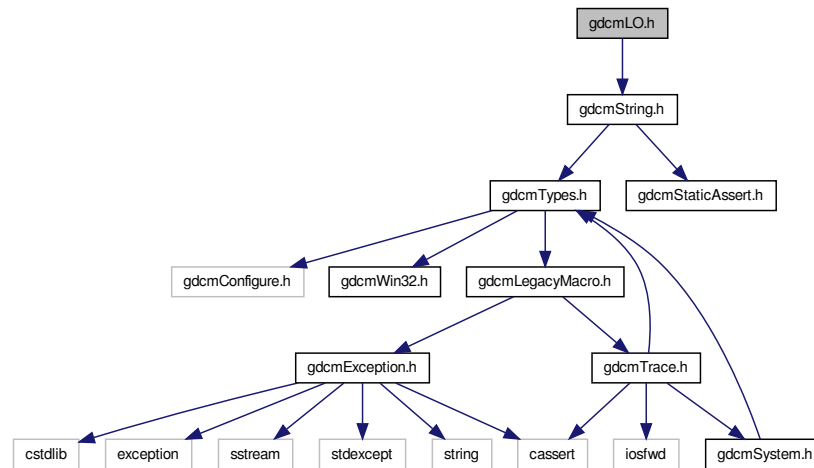
28.140.1.2 `#define GDCM_LEGACY_BODY( method, version ) gdcmWarningMacro(#method " was deprecated for " version " and will be removed in a future version.")`

28.140.1.3 `#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.")`

## 28.141 gdcmLO.h File Reference

```
#include "gdcmString.h"
```

Include dependency graph for gdcmLO.h:



## Classes

- class `gdcm::LO`

*LO.*

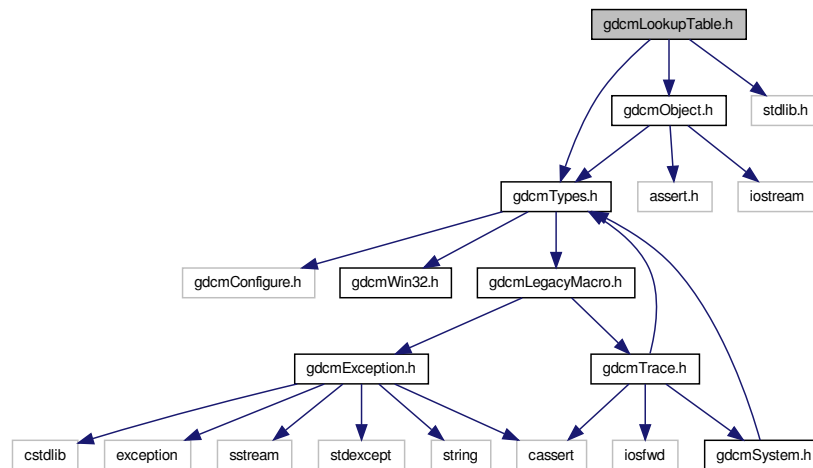
## Namespaces

- `gdcm`

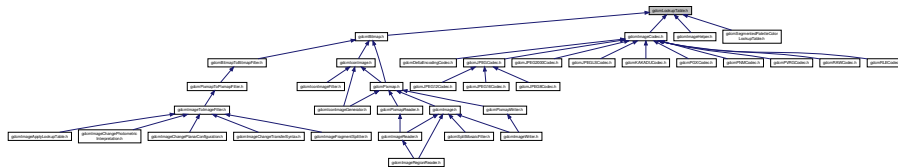
## 28.142 gdcmLookupTable.h File Reference

```
#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](#)

## 28.143 gdcmMacro.h File Reference

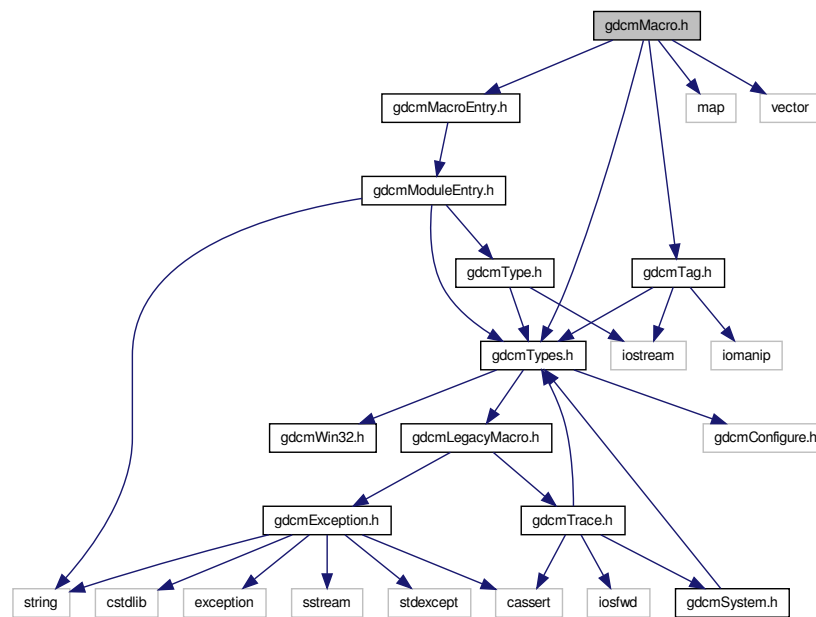
```

#include "gdcmTypes.h"
#include "gdcmTag.h"
#include "gdcmMacroEntry.h"
#include <map>
#include <vector>

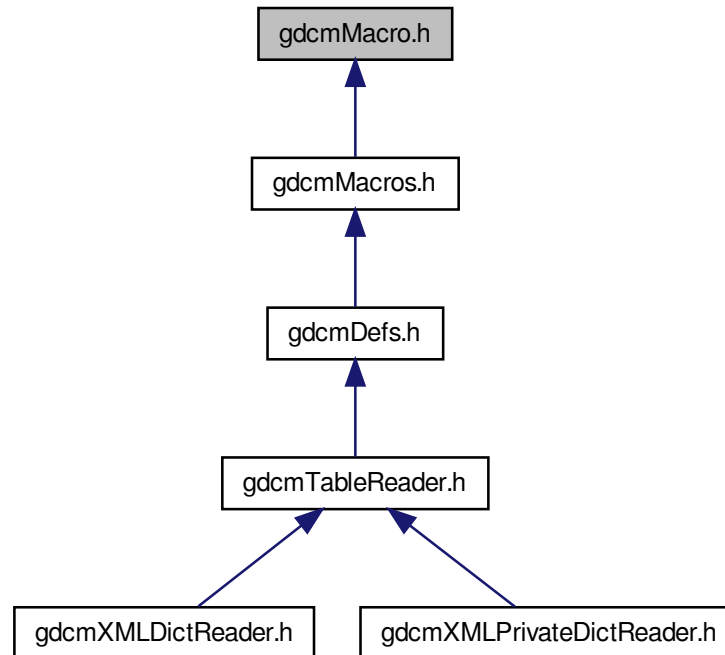
```



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](#)

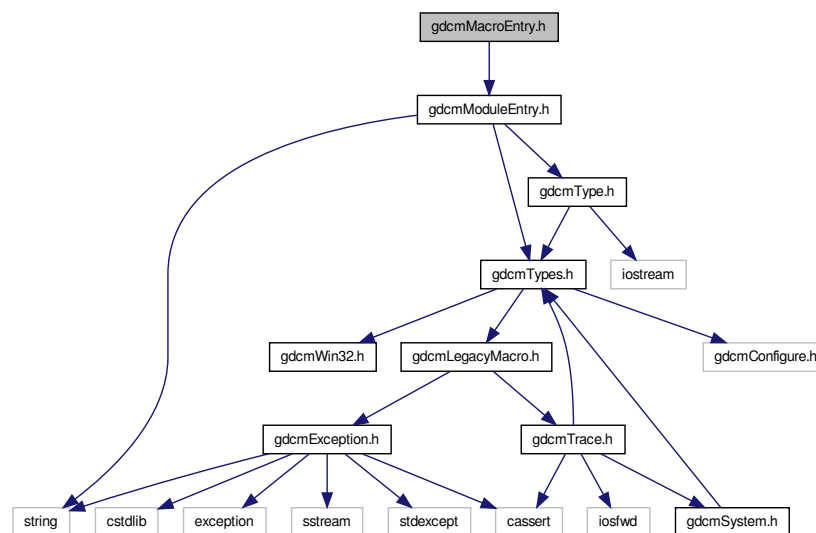
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macro &_val)`

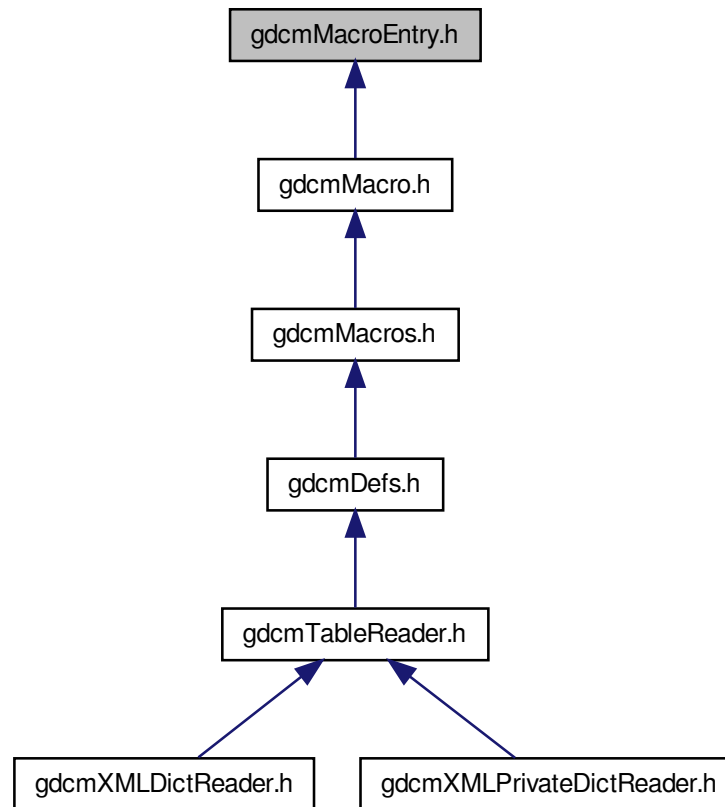
## 28.144 gdcMacroEntry.h File Reference

```
#include "gdcModuleEntry.h"
```

Include dependency graph for gdcmMacroEntry.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define GDCMMACROENTRY_H`

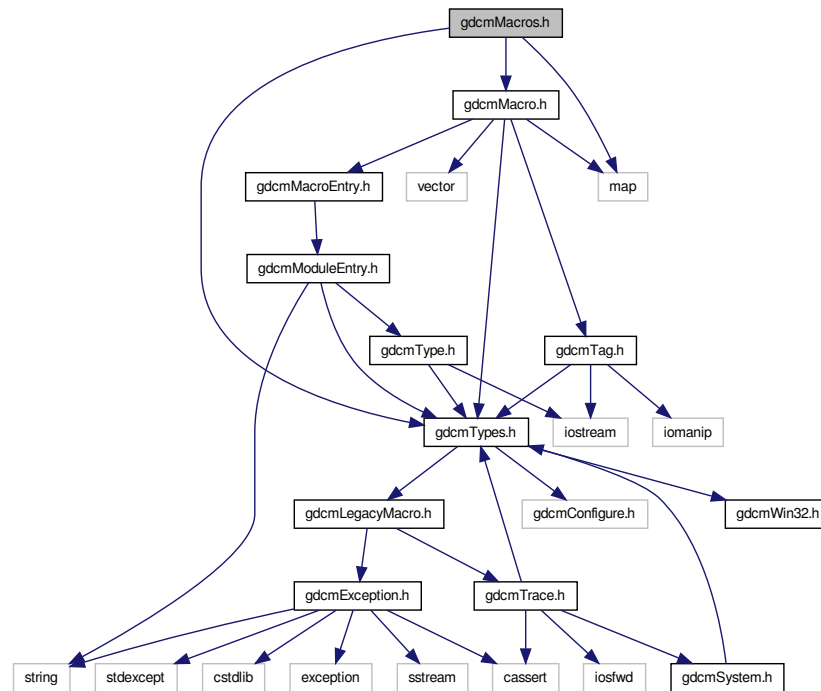
### 28.144.1 Macro Definition Documentation

#### 28.144.1.1 `#define GDCMMACROENTRY_H`

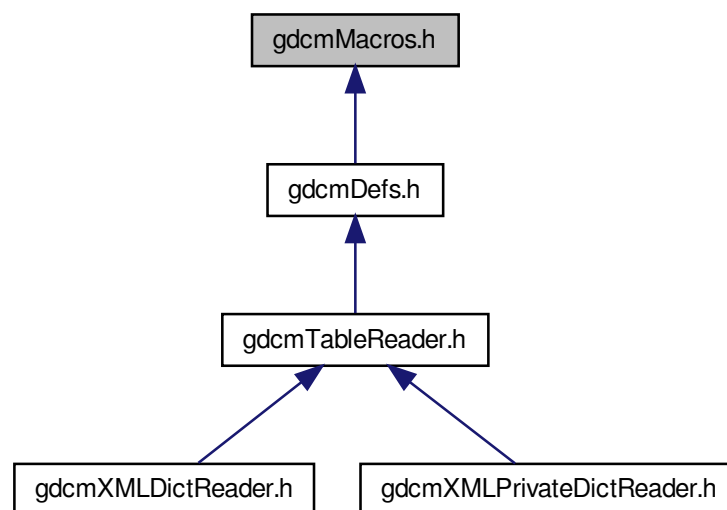
## 28.145 gdcmMacros.h File Reference

```
#include "gdcmTypes.h"  
#include "gdcmMacro.h"  
#include <map>
```

Include dependency graph for gdcMacros.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Macros](#)

*Class for representing a [Modules](#).*

## Namespaces

- [gdcm](#)

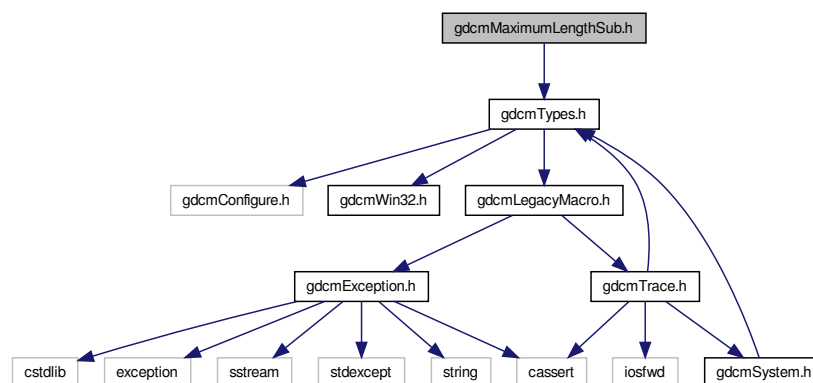
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Macros &_val)`

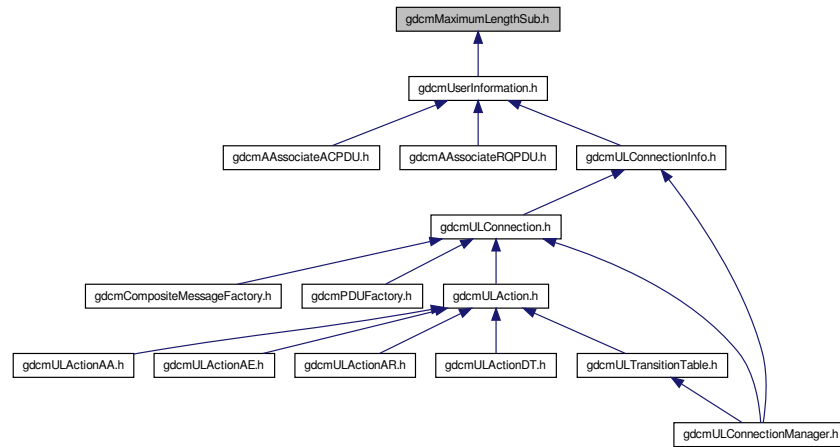
## 28.146 gdcmMaximumLengthSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmMaximumLengthSub.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcmd::network::MaximumLengthSub](#)

*MaximumLengthSub* Annex D Table D.1-1 MAXIMUM LENGTH SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

## Namespaces

- [gdcmd](#)
- [gdcmd::network](#)

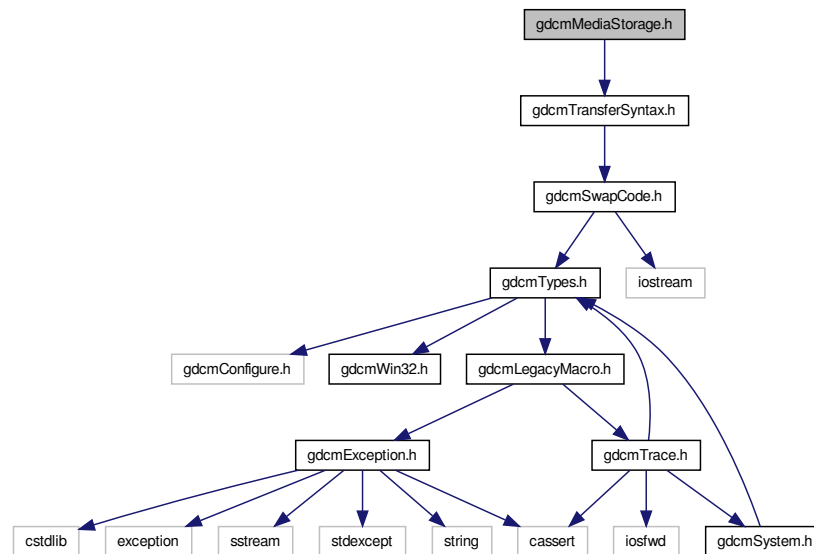
## 28.147 gdcmd5.h File Reference

```
#include "gdcmdTypes.h"
```

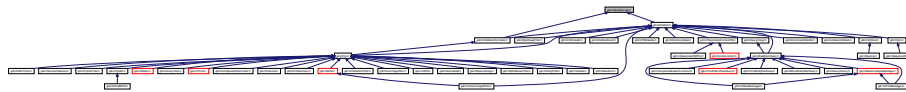




Include dependency graph for gdcmMediaStorage.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcm::MediaStorage`  
*MediaStorage.*

## Namespaces

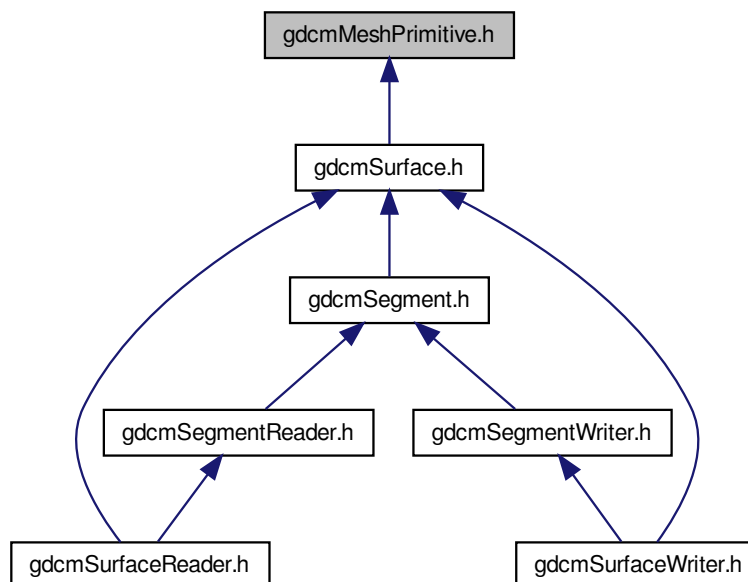
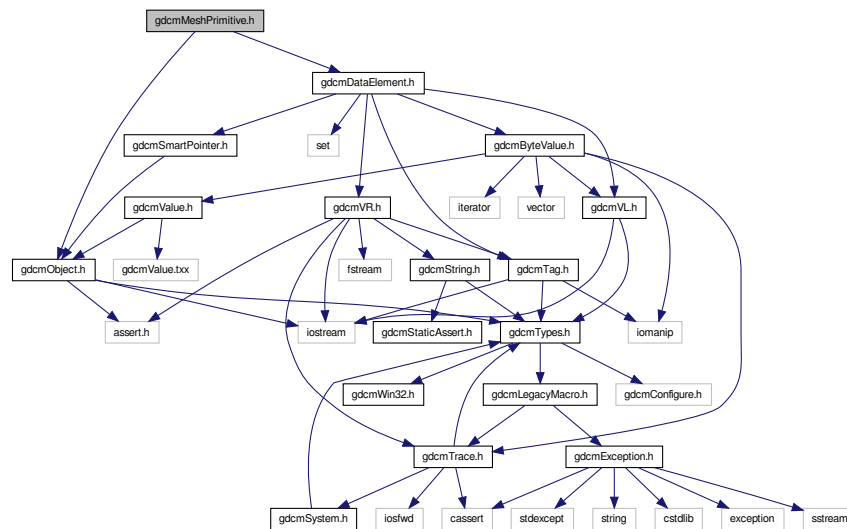
- `gdcm`

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const MediaStorage &ms)`

## 28.149 gdcmMeshPrimitive.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
```



## Classes

- class [gdcm::MeshPrimitive](#)

*This class defines surface mesh primitives. It is designed from surface mesh primitives macro.*

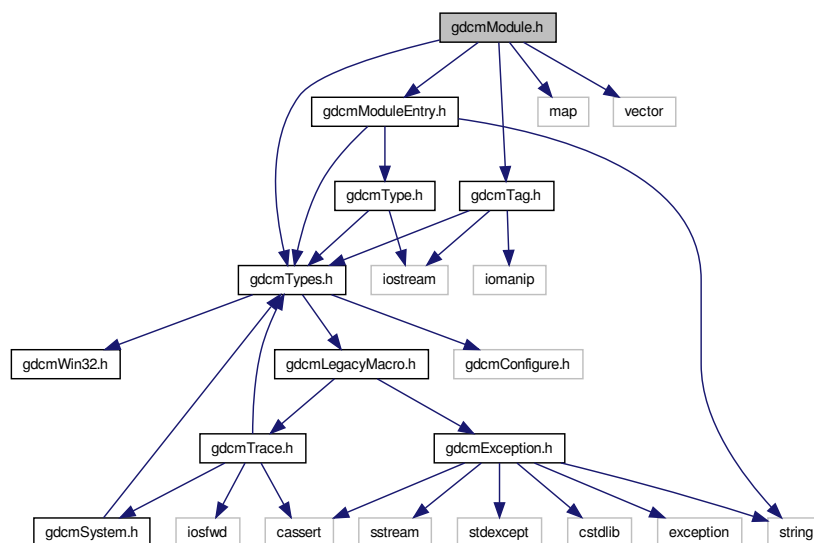
## Namespaces

- [gdcm](#)

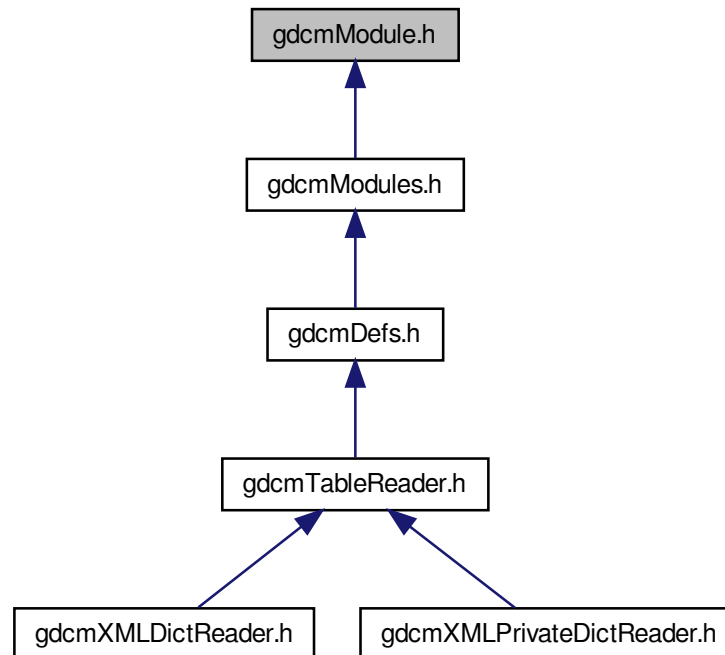
## 28.150 gdcmModule.h File Reference

```
#include "gdcmTypes.h"  
#include "gdcmTag.h"  
#include "gdcmModuleEntry.h"  
#include <map>  
#include <vector>
```

Include dependency graph for gdcmModule.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Module](#)  
*Class for representing a [Module](#).*

## Namespaces

- [gdc](#)

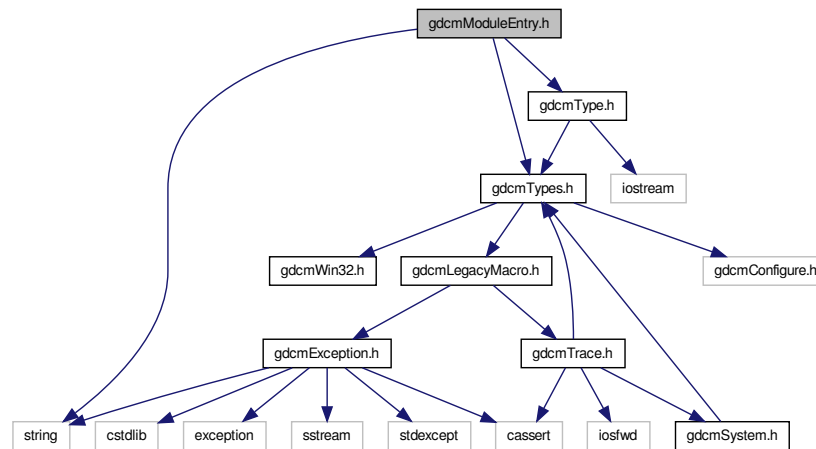
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Module &_val)`

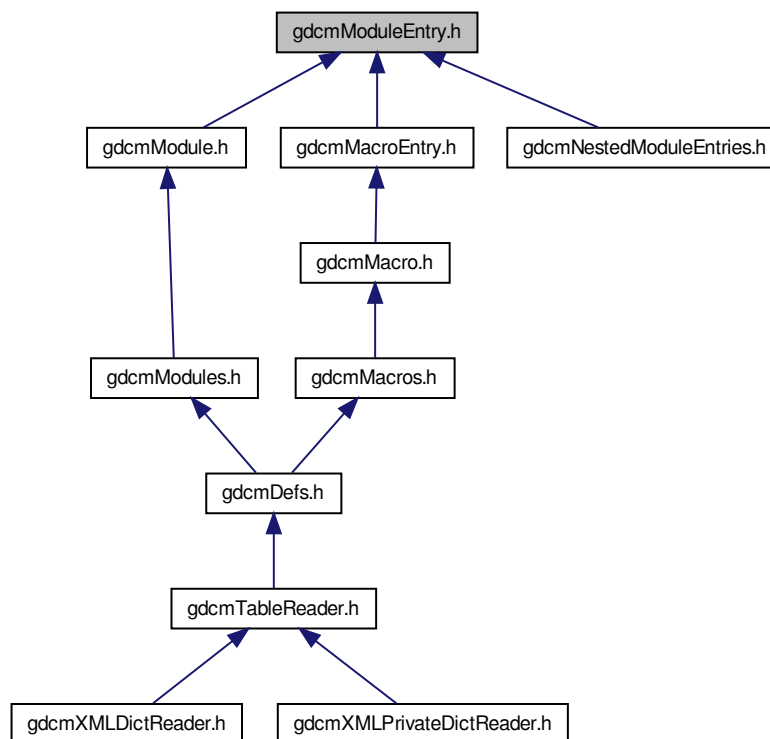
## 28.151 gdcModuleEntry.h File Reference

```
#include "gdcTypes.h"  
#include "gdcType.h"  
#include <string>
```

Include dependency graph for gdcModuleEntry.h:



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](#)

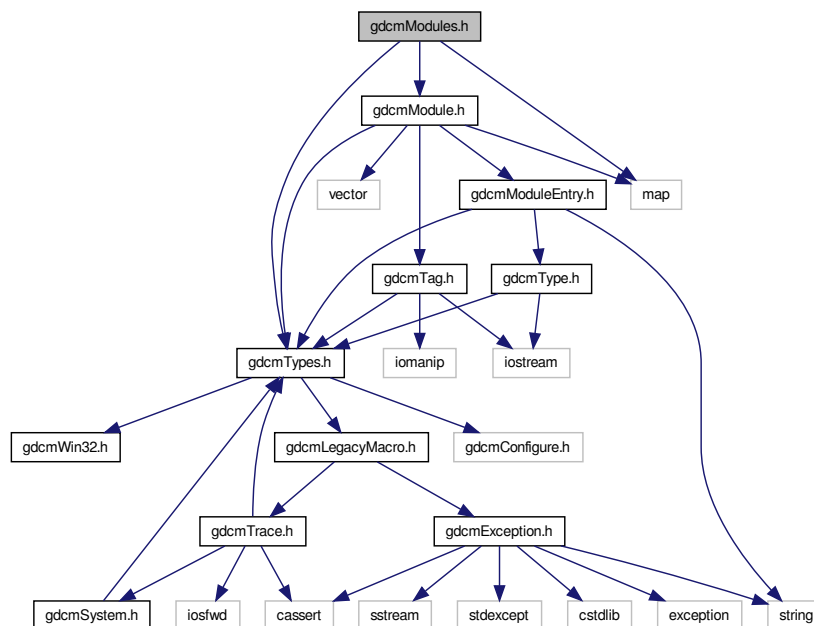
## Functions

- std::ostream & [gdcm::operator<<](#) (std::ostream &\_os, const ModuleEntry &\_val)

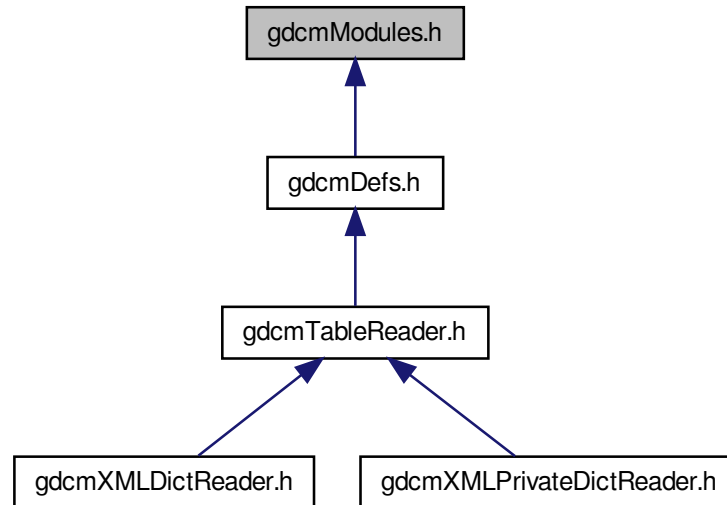
## 28.152 gdcmModules.h File Reference

```
#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:



## Classes

- class [gdcm::Modules](#)  
*Class for representing a [Modules](#).*

## Namespaces

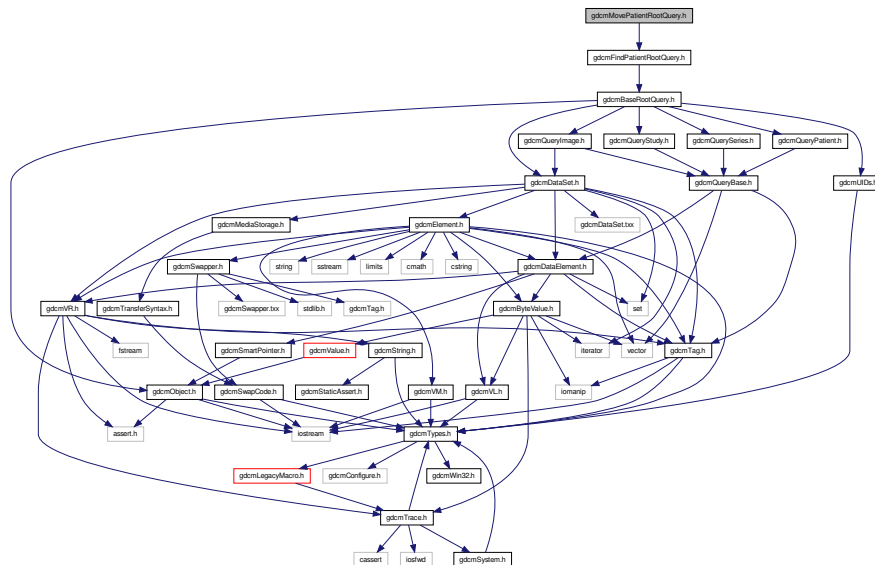
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Modules &_val)`

## 28.153 gdcmMovePatientRootQuery.h File Reference

```
#include "gdcmFindPatientRootQuery.h"
```



---



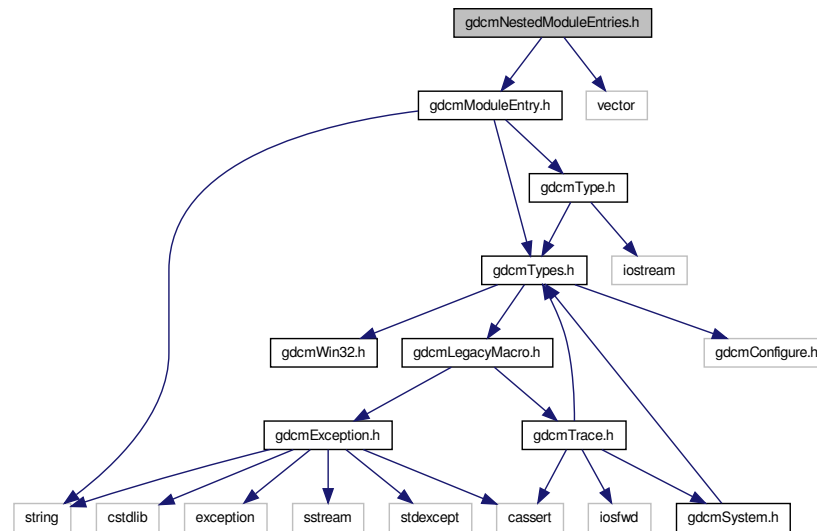
- class `gdcm::MoveStudyRootQuery`

*MoveStudyRootQuery* contains: the class which will produce a dataset for C-MOVE with study root.

- gdc

```
#include "gdcmModuleEntry.h"
#include <vector>
```

Include dependency graph for `gdcnNestedModuleEntries.h`:



## Classes

- class [gdcn::NestedModuleEntries](#)  
Class for representing a *NestedModuleEntries*.

## Namespaces

- [gdcn](#)

## Typedefs

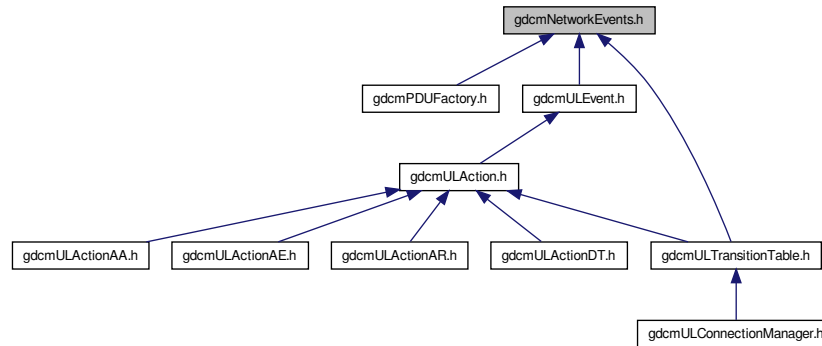
- typedef NestedModuleEntries [gdcn::NestedMacroEntries](#)

## Functions

- `std::ostream & gdcn::operator<< (std::ostream &_os, const NestedModuleEntries &_val)`

## 28.156 gdcNetworkEvents.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- `gdc`
- `gdc::network`

### Enumerations

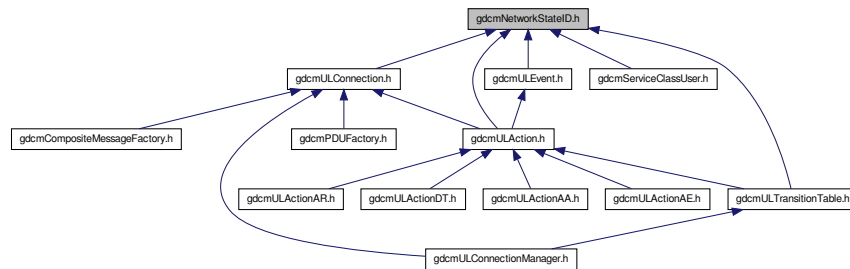
- `enum gdc::network::EEventID {`  
`gdc::network::eAASSOCIATERequestLocalUser = 0,`  
`gdc::network::eTransportConnConfirmLocal,`  
`gdc::network::eASSOCIATE_ACPDUreceived,`  
`gdc::network::eASSOCIATE_RJPDUreceived,`  
`gdc::network::eTransportConnIndicLocal,`  
`gdc::network::eAASSOCIATE_RQPDUreceived,`  
`gdc::network::eAASSOCIATEResponseAccept,`  
`gdc::network::eAASSOCIATEResponseReject,`  
`gdc::network::ePDATArequest,`  
`gdc::network::ePDATATFPDU,`  
`gdc::network::eARELEASERequest,`  
`gdc::network::eARELEASE_RQPDUReceivedOpen,`  
`gdc::network::eARELEASE_RPPDUReceived,`  
`gdc::network::eARELEASEResponse,`  
`gdc::network::eAABORTRequest,`  
`gdc::network::eAABORTPDUReceivedOpen,`  
`gdc::network::eTransportConnectionClosed,`  
`gdc::network::eARTIMTimerExpired,`  
`gdc::network::eUnrecognizedPDUReceived,`  
`gdc::network::eEventDoesNotExist }`

### Variables

- `const int gdc::network::cMaxEventID = eEventDoesNotExist`

## 28.157 gdcmlNetworkStateID.h File Reference

This graph shows which files directly or indirectly include this file:



### Namespaces

- [gdcml](#)
- [gdcml::network](#)

### Enumerations

- `enum gdcml::network::EStateID {  
gdcml::network::eStaDoesNotExist = 0,  
gdcml::network::eSta1Idle = 1,  
gdcml::network::eSta2Open = 2,  
gdcml::network::eSta3WaitLocalAssoc = 4,  
gdcml::network::eSta4LocalAssocDone = 8,  
gdcml::network::eSta5WaitRemoteAssoc = 16,  
gdcml::network::eSta6TransferReady = 32,  
gdcml::network::eSta7WaitRelease = 64,  
gdcml::network::eSta8WaitLocalRelease = 128,  
gdcml::network::eSta9ReleaseCollisionRqLocal = 256,  
gdcml::network::eSta10ReleaseCollisionAc = 512,  
gdcml::network::eSta11ReleaseCollisionRq = 1024,  
gdcml::network::eSta12ReleaseCollisionAcLocal = 2048,  
gdcml::network::eSta13AwaitingClose = 4096 }`

### Functions

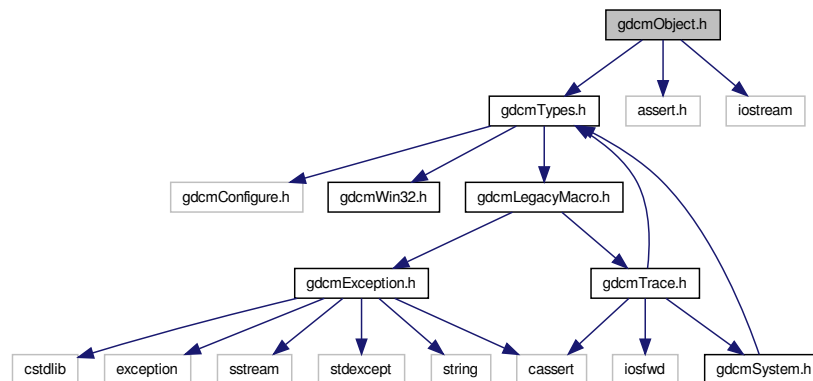
- `int gdcml::network::GetStateIndex (EStateID inState)`

### Variables

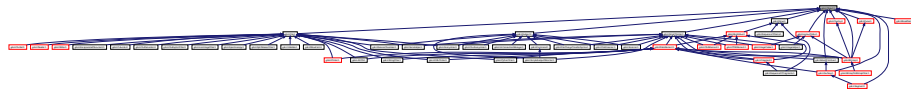
- `const int gdcml::network::cMaxStateID = 13`

## 28.158 gdcmObject.h File Reference

```
#include "gdcmTypes.h"
#include <assert.h>
#include <iostream>
Include dependency graph for gdcmObject.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [gdcm::Object](#)  
*Object.*
- class [gdcm::SmartPointer< ObjectType >](#)  
*Class for Smart Pointer.*

### Namespaces

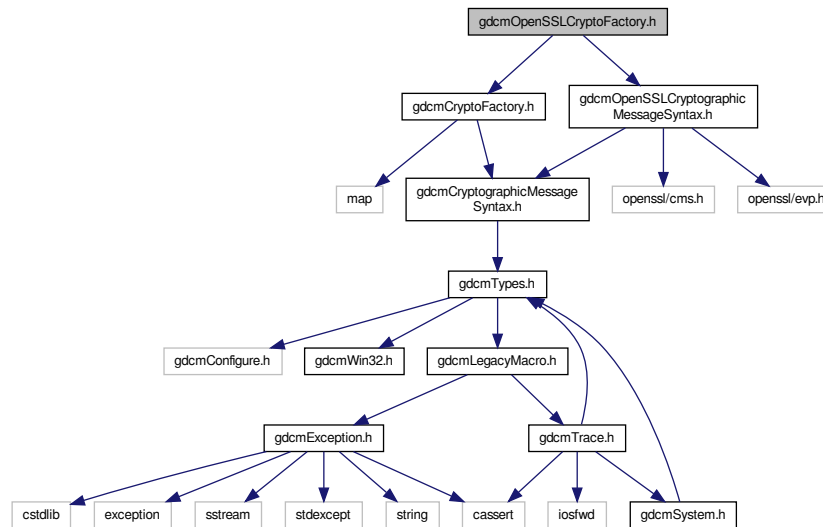
- [gdcm](#)

### Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Object &obj)`

## 28.159 gdcmOpenSSLCryptoFactory.h File Reference

```
#include "gdcmCryptoFactory.h"
#include "gdcmOpenSSLCryptographicMessageSyntax.h"
Include dependency graph for gdcmOpenSSLCryptoFactory.h:
```



### Classes

- class [gdcm::OpenSSLCryptoFactory](#)

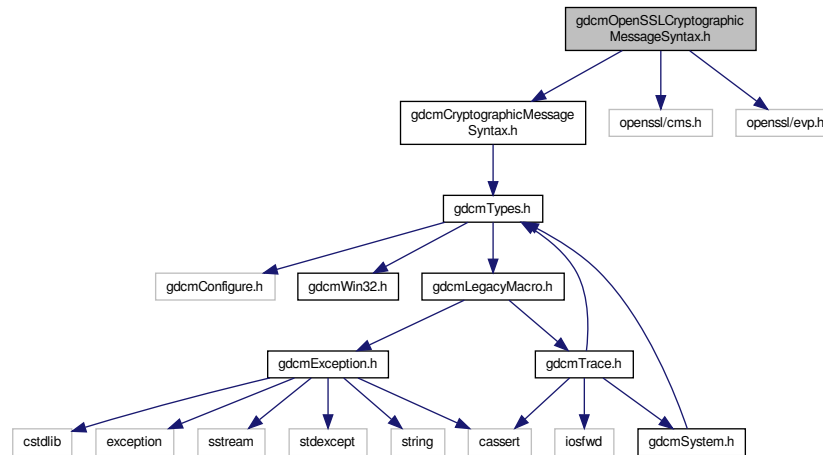
### Namespaces

- [gdcm](#)

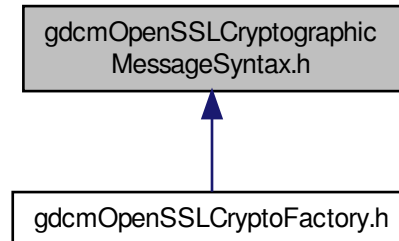
## 28.160 gdcmOpenSSLCryptographicMessageSyntax.h File Reference

```
#include "gdcmCryptographicMessageSyntax.h"
#include <openssl/cms.h>
#include <openssl/evp.h>
```

Include dependency graph for gdcOpenSSLCryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::OpenSSLCryptographicMessageSyntax](#)

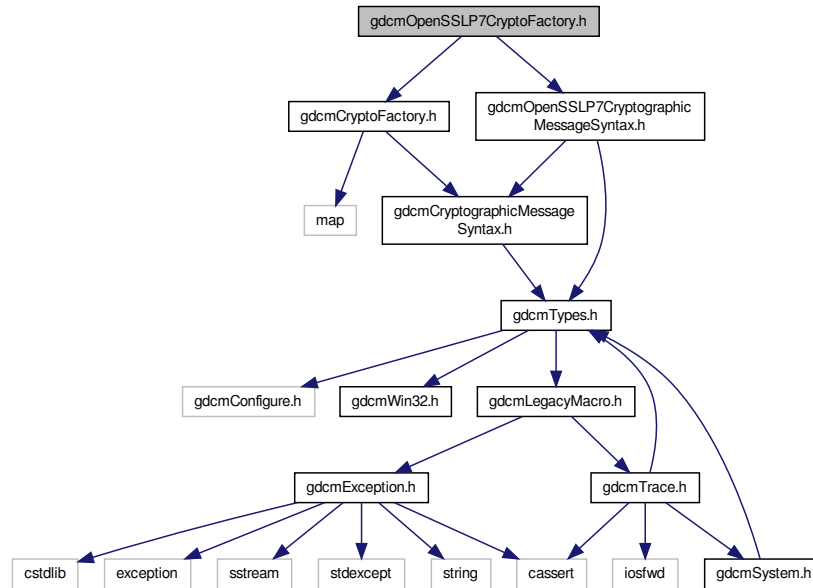
## Namespaces

- [gdc](#)

## 28.161 gdcOpenSSLP7CryptoFactory.h File Reference

```
#include "gdcCryptoFactory.h"
```

```
#include "gdcOpenSSL7CryptographicMessageSyntax.h"
Include dependency graph for gdcOpenSSL7CryptoFactory.h:
```



## Classes

- class [gdc::OpenSSL7CryptoFactory](#)

## Namespaces

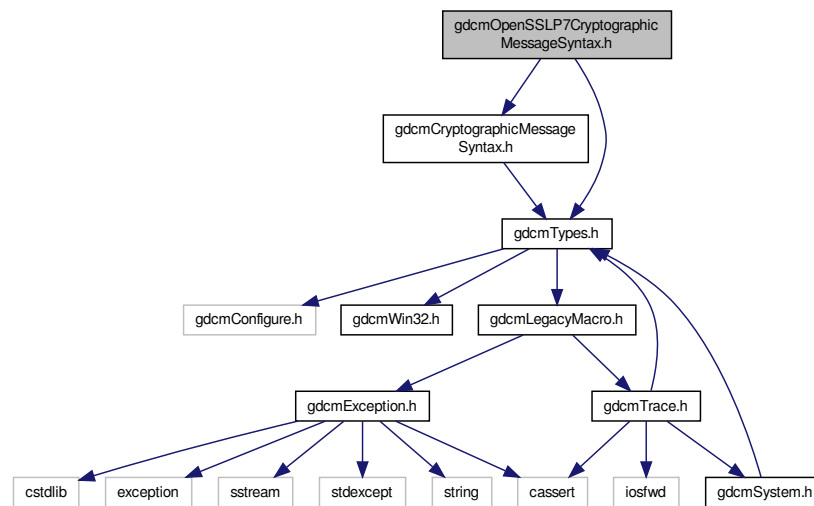
- [gdc](#)

## 28.162 gdcOpenSSL7CryptographicMessageSyntax.h File Reference

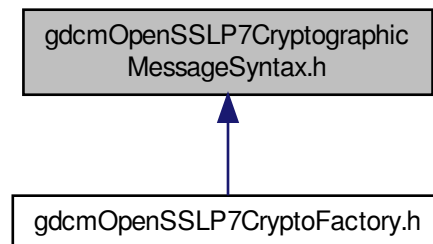
```
#include "gdcCryptographicMessageSyntax.h"
#include "gdcTypes.h"
```



Include dependency graph for gdcmOpenSSLP7CryptographicMessageSyntax.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::OpenSSLP7CryptographicMessageSyntax](#)

Class for [CryptographicMessageSyntax](#) encryption. This is just a simple wrapper around openssl PKCS7\_encrypt functionalities.

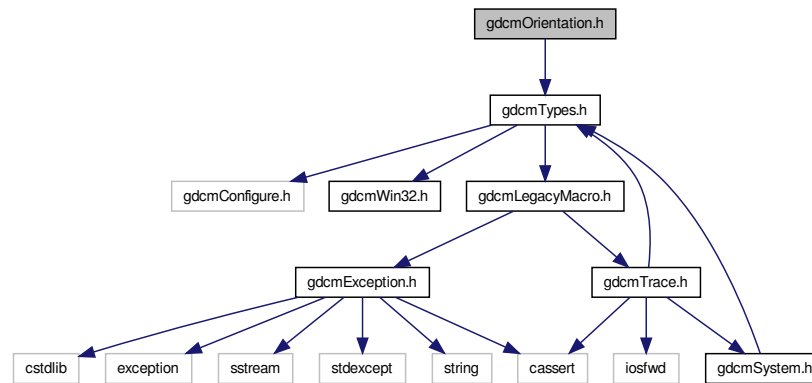
## Namespaces

- [gdcm](#)

## 28.163 gdcmOrientation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmOrientation.h:



### Classes

- class `gdcm::Orientation`  
*class to handle `Orientation`*

### Namespaces

- `gdcm`

### Functions

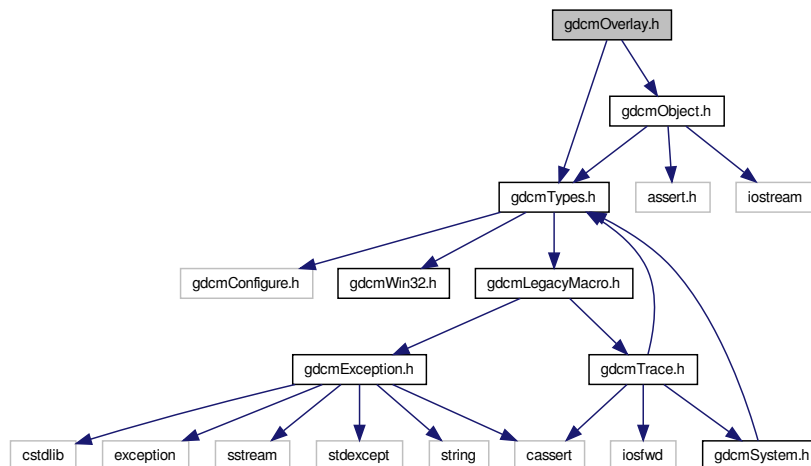
- `std::ostream & gdcm::operator<< (std::ostream &os, const Orientation &o)`

## 28.164 gdcmOverlay.h File Reference

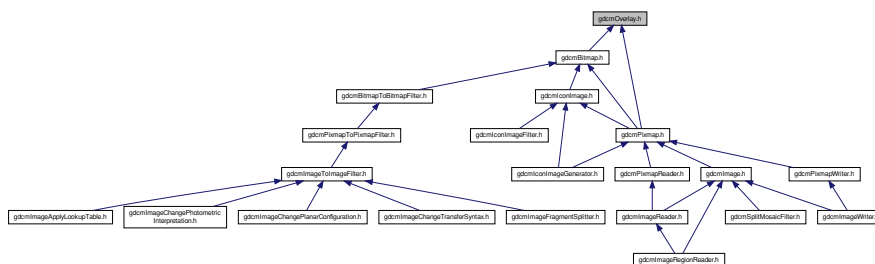
```
#include "gdcmTypes.h"
```

```
#include "gdcmObject.h"
```

```
#include "gdcmException.h"
```



## Classes



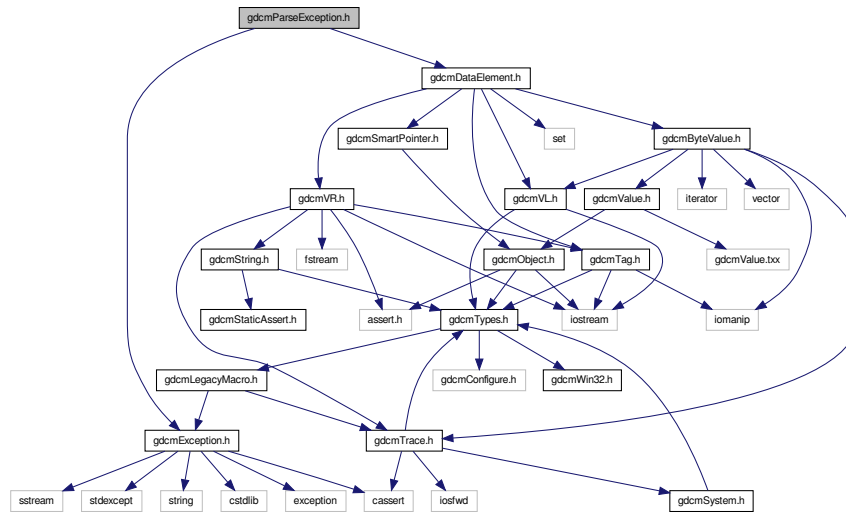
## 28.165 gdcmpap3.man File Reference

Generated on Fri Jul 18 2014 07:12:17 for GDCM by Doxygen

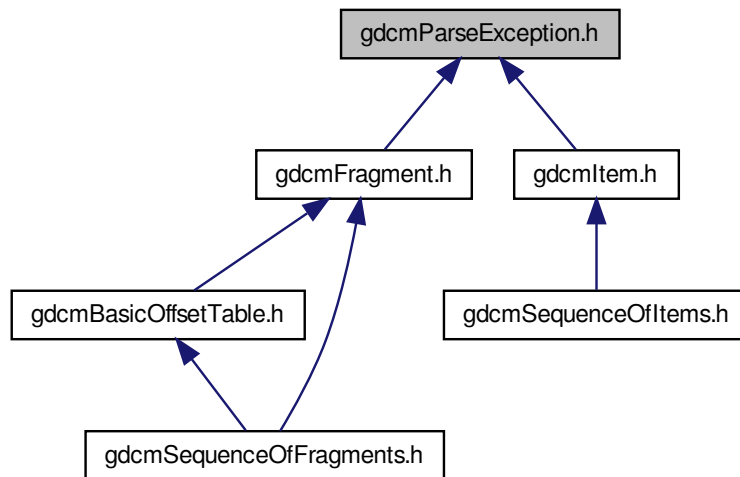
- **gdcm**

```
#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.

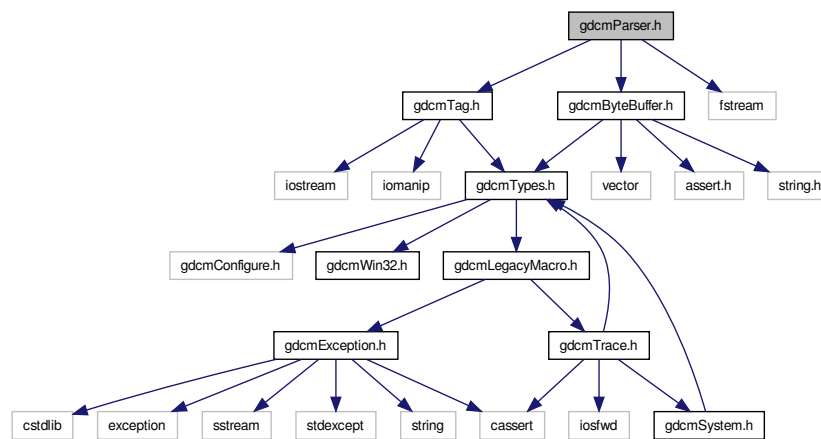
## Namespaces

- [gdcm](#)

## 28.167 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)*

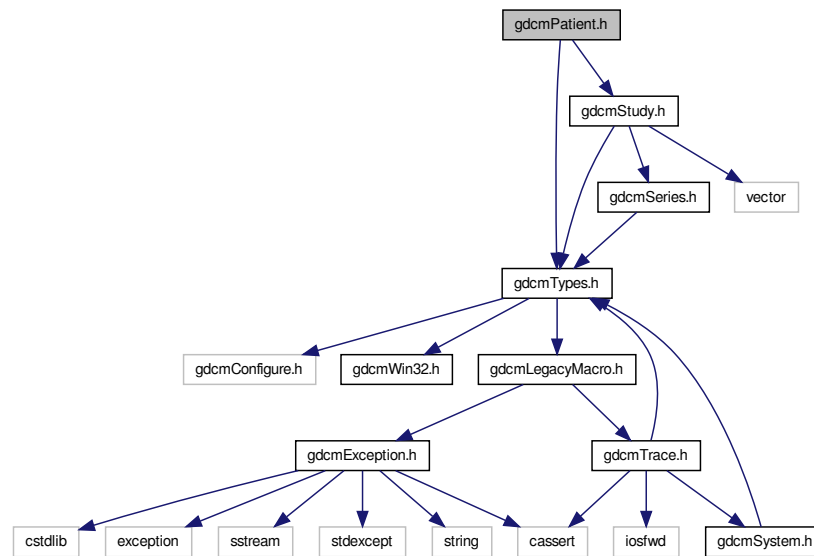
## Namespaces

- [gdcm](#)

## 28.168 gdcmPatient.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmStudy.h"
```

Include dependency graph for gdcmPidient.h:



## Classes

- class [gdcmPid::Patient](#)

See PS 3.3 - 2007 DICOM MODEL OF THE REAL-WORLD, p 54.

## Namespaces

- [gdcmPid](#)

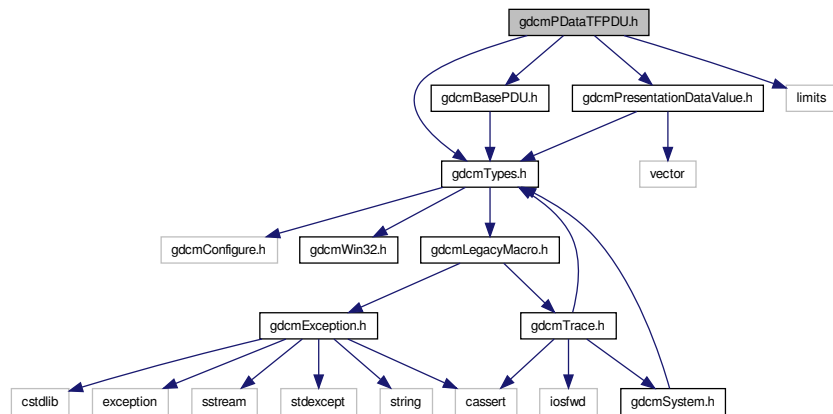
## 28.169 gdcmPidDataTFPDU.h File Reference

```

#include "gdcmPidTypes.h"
#include "gdcmPidPresentationDataValue.h"
#include "gdcmPidBasePDU.h"
#include <limits>

```

Include dependency graph for gdcmPDataTFPDU.h:



## Classes

- class [gdcm::network::PDataTFPDU](#)

*PDataTFPDU Table 9-22 P-DATA-TF PDU FIELDS.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.170 gdcmPDBelement.h File Reference

```

#include "gdcmTag.h"
#include "gdcmVM.h"
#include "gdcmVR.h"
#include "gdcmByteValue.h"
#include "gdcmSmartPointer.h"

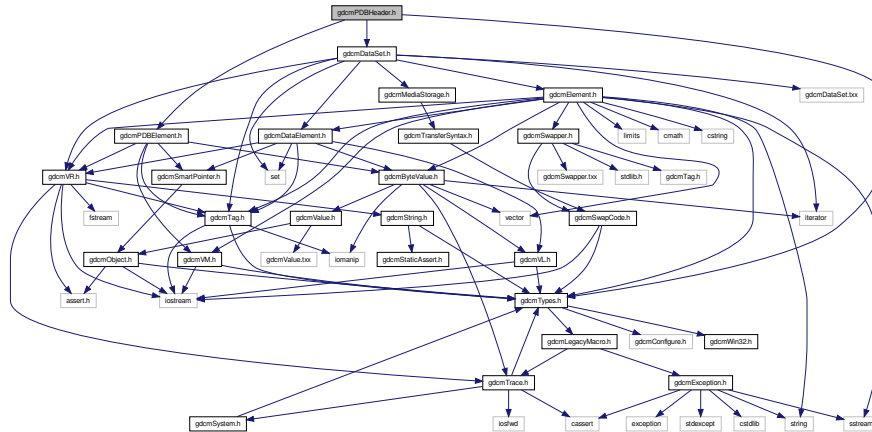
```





## 28.171 gdcmPDBHeader.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmDataSet.h"
#include "gdcmPDBElement.h"
Include dependency graph for gdcmPDBHeader.h:
```



### Classes

- class [gdcm::PDBHeader](#)  
Class for *PDBHeader*.

### Namespaces

- [gdcm](#)

### Functions

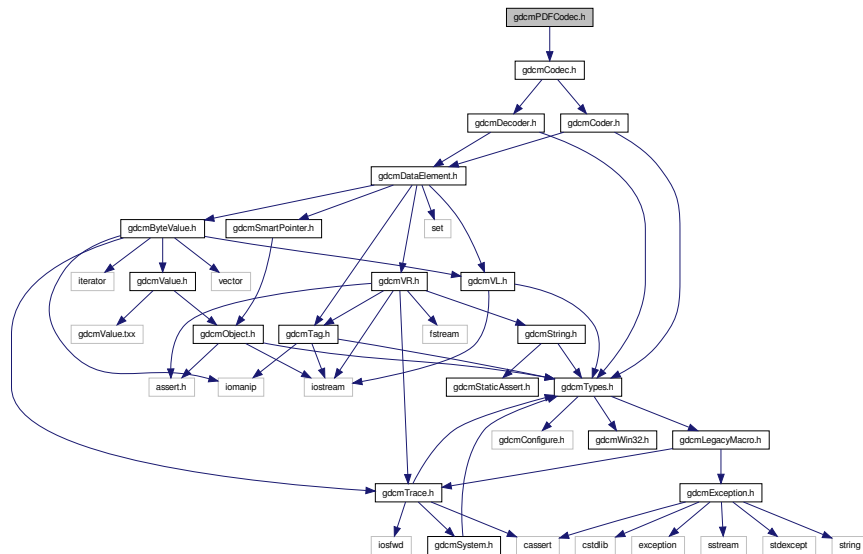
- `std::ostream & gdcm::operator<< (std::ostream &os, const PDBHeader &d)`

## 28.172 gdcmpdf.man File Reference

## 28.173 gdcmPDFCodec.h File Reference

```
#include "gdcmCodec.h"
```

Include dependency graph for `gdcmPDFCodec.h`:



## Classes

- class [gdcm::PDFCodec](#)

*PDFCodec* class.

## Namespaces

- [gdcm](#)

## 28.174 gdcmPDUFactory.h File Reference

```

#include "gdcmTypes.h"
#include "gdcmNetworkEvents.h"
#include "gdcmULConnection.h"
#include "gdcmPresentationDataValue.h"

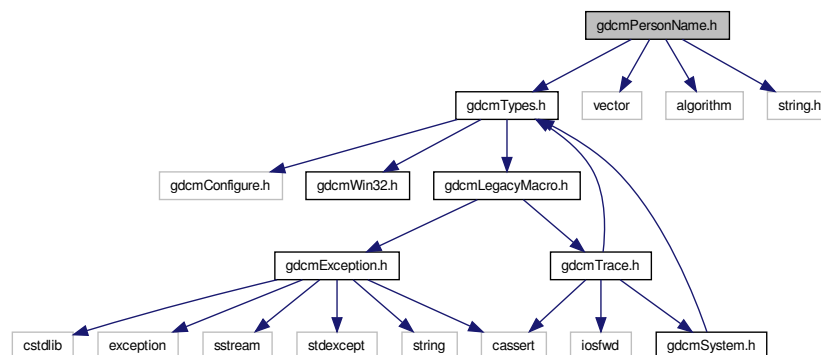
```

- class `gdcm::network::PDUFactory`

## Namespaces

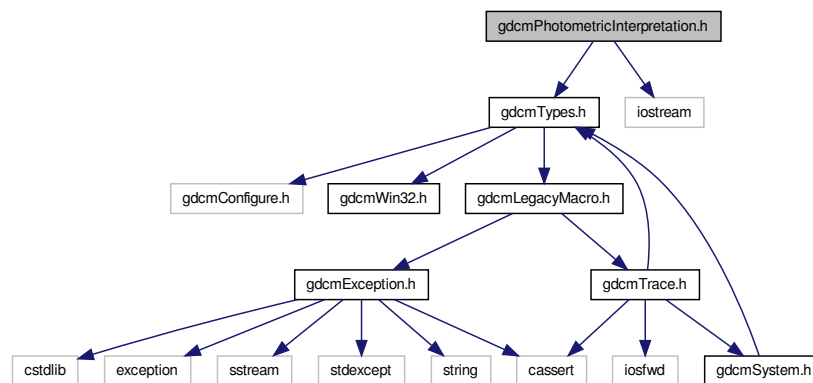
- ## 28.175 gdcmPersonName.h File Reference

Include dependency graph for `gdcmPersonName.h`:

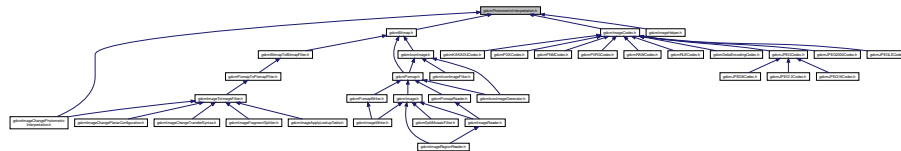




Include dependency graph for gdcmPhotometricInterpretation.h:



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)`

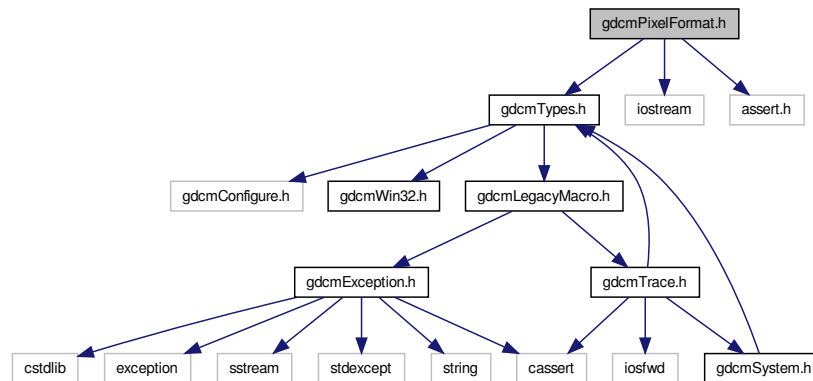
## 28.178 gdcmPixelFormat.h File Reference

```

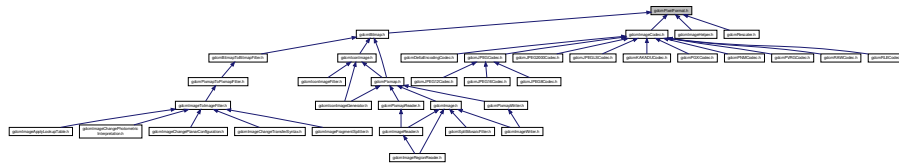
#include "gdcmTypes.h"
#include <iostream>
#include <assert.h>

```

Include dependency graph for `gdcmPidelFormat.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)`

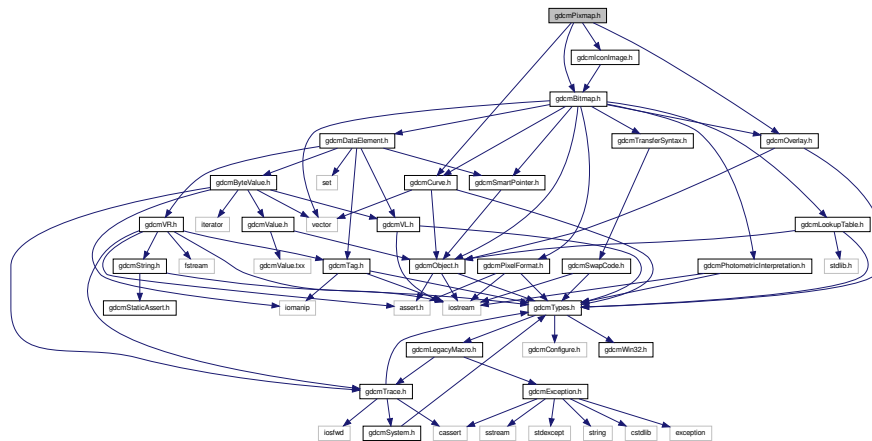
## 28.179 gdcmPidap.h File Reference

```

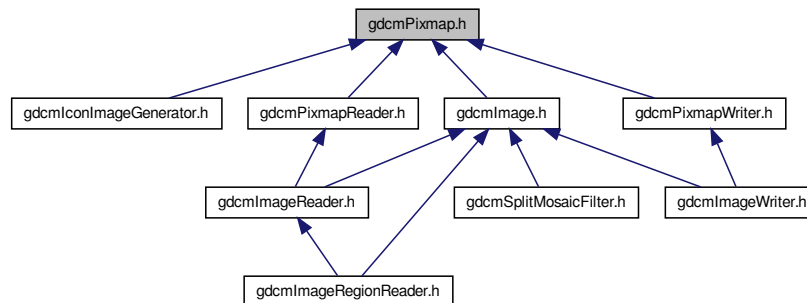
#include "gdcmPidap.h"
#include "gdcmPidap.h"
#include "gdcmPidap.h"
#include "gdcmPidap.h"

```

Include dependency graph for gdcmPixmap.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Pixmap](#)

*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)

## Namespaces

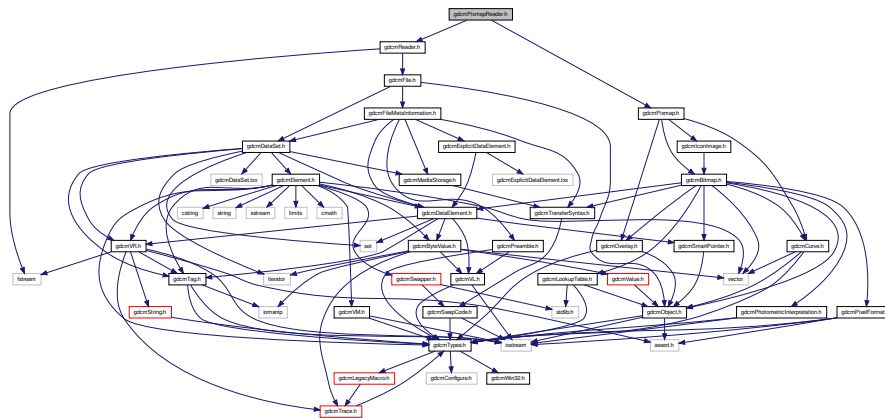
- [gdcm](#)

## 28.180 gdcmPixmapReader.h File Reference

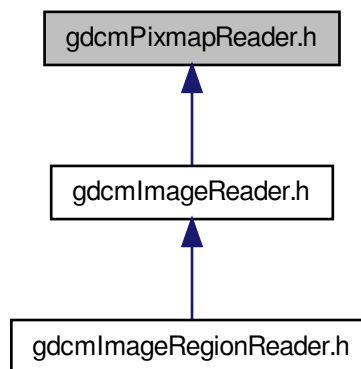
```
#include "gdcmReader.h"
```

```
#include "gdcmPixmap.h"
```

Include dependency graph for gdcmPixmapReader.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::PixmapReader](#)  
*PixmapReader.*

## Namespaces

- [gdcm](#)





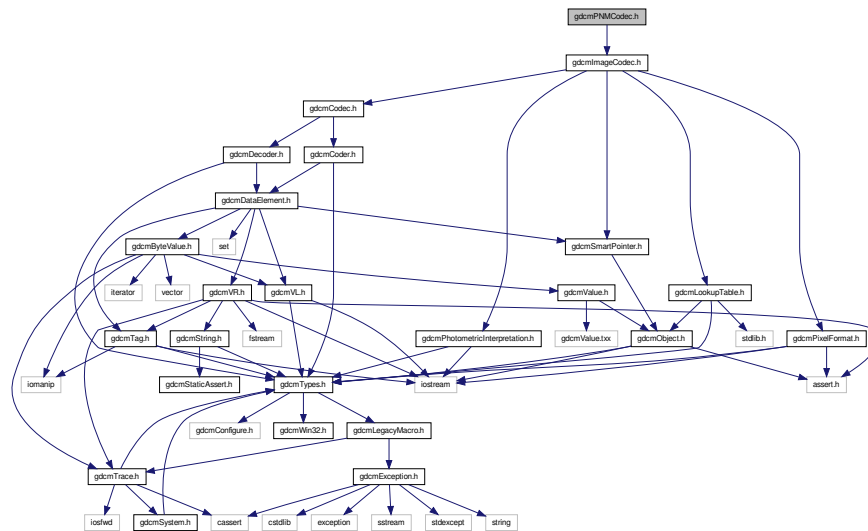
```
graph BT; gdcmImageWriter.h --> gdcmPixmapWriter.h
```

- class `gdcm::PixmapWriter`  
*PixmapWriter* This class will takes two inputs:

- **gdcm**

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPNMCodec.h:



## Classes

- class [gdcm::PNMCodec](#)

*Class to do PNM PNM is the Portable anymap file format. The main web page can be found at: <http://netpbm.sourceforge.net/>.*

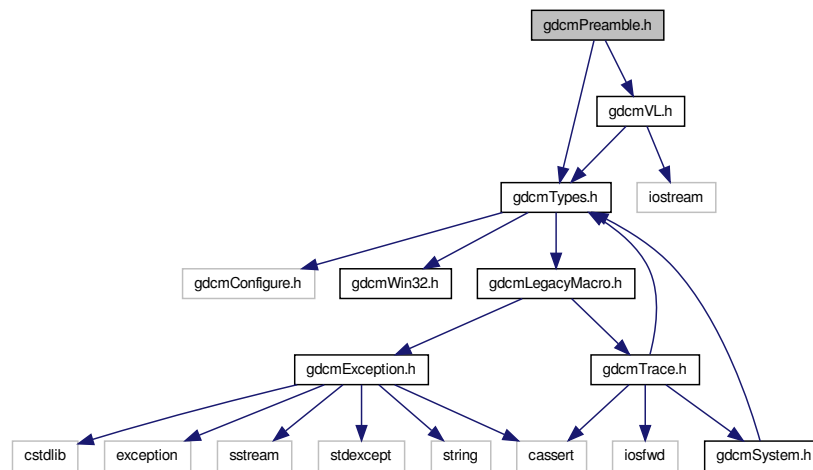
## Namespaces

- [gdcm](#)

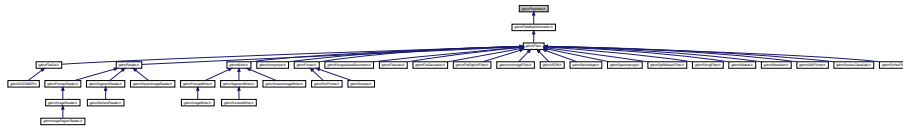
## 28.184 gdcmPreamble.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmVL.h"
```

Include dependency graph for `gdcnPreamble.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcn::Preamble](#)  
*DICOM Preamble (Part 10)*

## Namespaces

- [gdcn](#)

## Functions

- `std::ostream & gdcn::operator<< (std::ostream &os, const Preamble &val)`

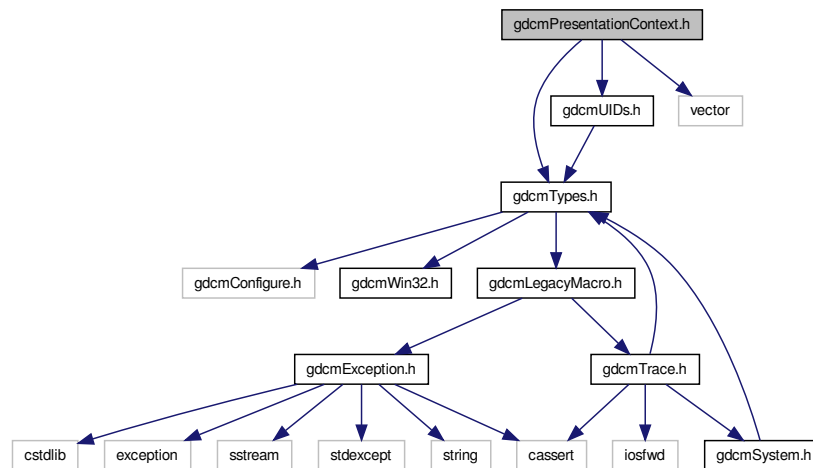
## 28.185 gdcnPresentationContext.h File Reference

```

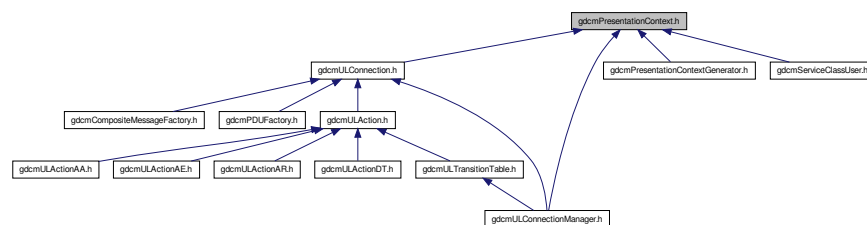
#include "gdcnTypes.h"
#include "gdcnUIDs.h"
#include <vector>

```

Include dependency graph for gdcmPresentationContext.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::PresentationContext](#)  
*PresentationContext.*

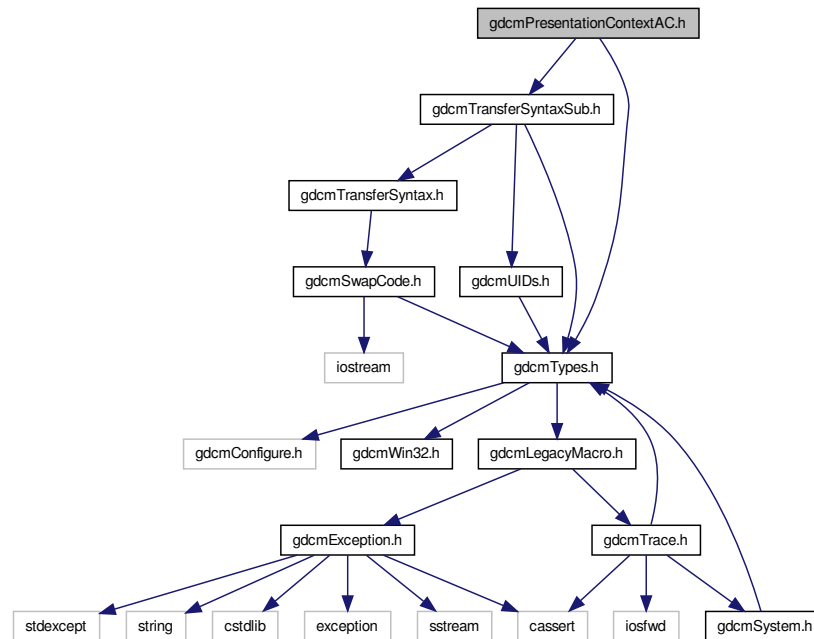
## Namespaces

- [gdcm](#)

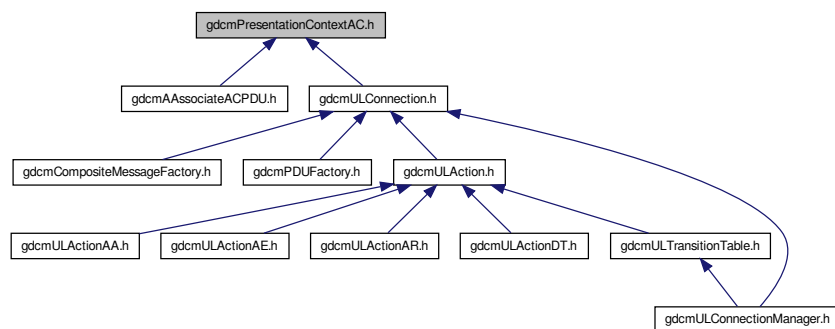
## 28.186 gdcmPresentationContextAC.h File Reference

```
#include "gdcmTypes.h"
#include "gdcmTransferSyntaxSub.h"
```

Include dependency graph for `gdcmPidentationContextAC.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdcmPid::network::PresentationContextAC`

*PresentationContextAC Table 9-18 PRESENTATION CONTEXT ITEM FIELDS.*

## Namespaces

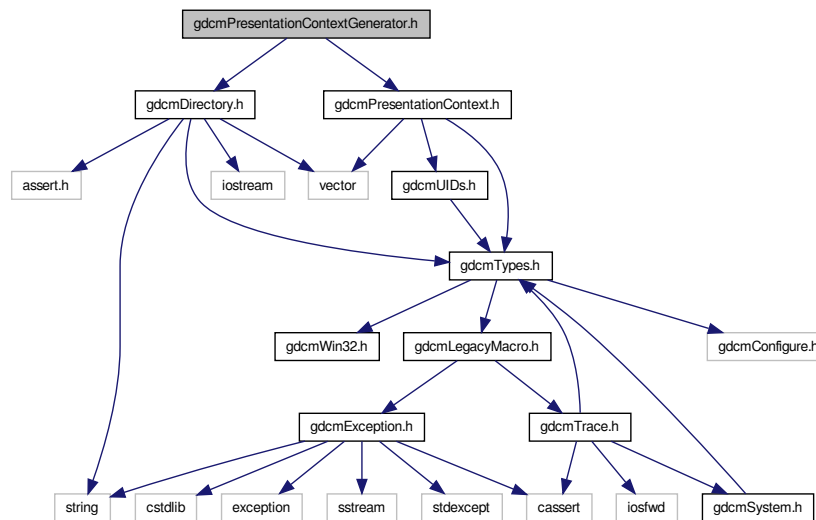
- [gdcm](#)
- [gdcm::network](#)

## 28.187 gdcmPresentationContextGenerator.h File Reference

```
#include "gdcmDirectory.h"
```

```
#include "gdcmPresentationContext.h"
```

Include dependency graph for gdcmPresentationContextGenerator.h:



## Classes

- class [gdcm::PresentationContextGenerator](#)

***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.*

## Namespaces

- [gdcm](#)

## 28.188 gdcmPresentationContextRQ.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmAbstractSyntax.h"
```

```
#include "gdcmTransferSyntaxSub.h"
```

```
#include "gdcmDataSet.h"
```

```

classDiagram
    class gdcmPresentationContextRQ_h["gdcmPresentationContextRQ.h"]
    class gdcmAssociateRQPDU_h["gdcmAssociateRQPDU.h"]
    class gdcmULConnection_h["gdcmULConnection.h"]
    class gdcmCompositeMessageFactory_h["gdcmCompositeMessageFactory.h"]
    class gdcmPDUFactory_h["gdcmPDUFactory.h"]
    class gdcmULAction_h["gdcmULAction.h"]
    class gdcmULActionAA_h["gdcmULActionAA.h"]
    class gdcmULActionAE_h["gdcmULActionAE.h"]
    class gdcmULActionAR_h["gdcmULActionAR.h"]
    class gdcmULActionDT_h["gdcmULActionDT.h"]
    class gdcmULTransitionTable_h["gdcmULTransitionTable.h"]
    class gdcmULConnectionManager_h["gdcmULConnectionManager.h"]

    gdcmPresentationContextRQ_h <|-- gdcmAssociateRQPDU_h
    gdcmPresentationContextRQ_h <|-- gdcmULConnection_h
    gdcmULConnection_h <|-- gdcmCompositeMessageFactory_h
    gdcmULConnection_h <|-- gdcmPDUFactory_h
    gdcmULConnection_h <|-- gdcmULAction_h
    gdcmULAction_h <|-- gdcmULActionAA_h
    gdcmULAction_h <|-- gdcmULActionAE_h
    gdcmULAction_h <|-- gdcmULActionAR_h
    gdcmULAction_h <|-- gdcmULActionDT_h
    gdcmULAction_h <|-- gdcmULTransitionTable_h
    gdcmULConnectionManager_h <|-- gdcmULTransitionTable_h
  
```

- class `gdcm::network::PresentationContextRQ`

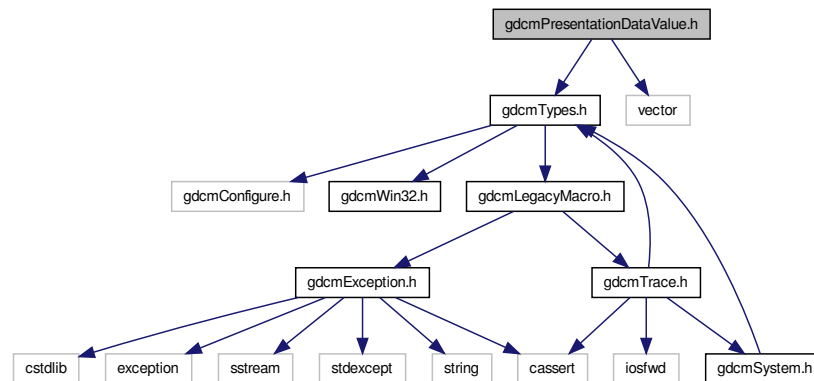
*PresentationContextRQ* Table 9-13 PRESENTATION CONTEXT ITEM FIELDS.

- `gdcm`
- `gdcm::network`

```
#include "gdcmTypes.h"
#include <vector>
```



Include dependency graph for gdcmPresentationDataValue.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::PresentationDataValue](#)

*PresentationDataValue Table 9-23 PRESENTATION-DATA-VALUE ITEM FIELDS.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.190 gdcmPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
```

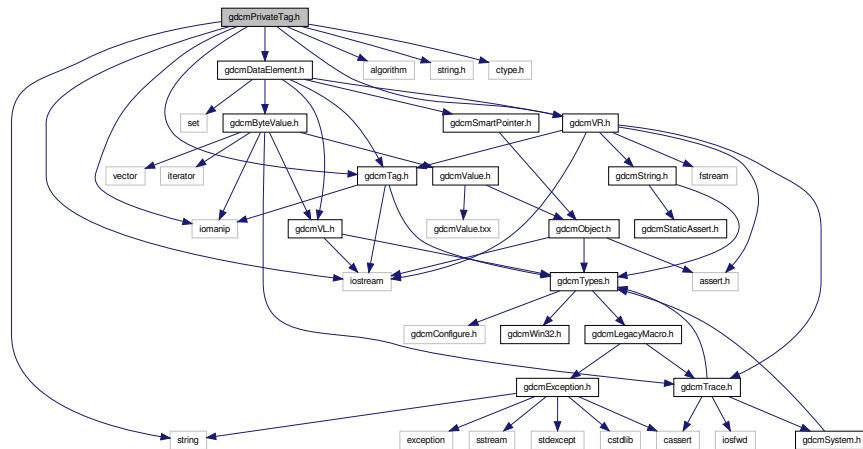


```

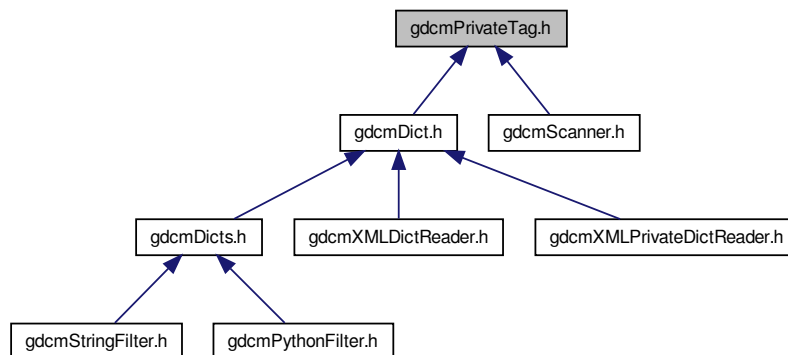
#include "gdcmVR.h"
#include "gdcmDataElement.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <string.h>
#include <ctype.h>

```

Include dependency graph for gdcmPrivateTag.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::PrivateTag](#)

*Class to represent a Private DICOM Data [Element](#) ([Attribute](#)) [Tag](#) (Group, [Element](#), Owner)*

## Namespaces

- [gdcm](#)

## Functions

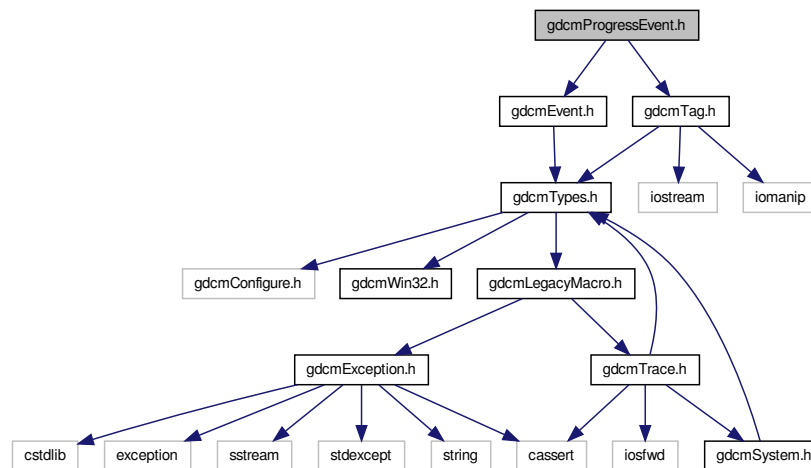
- `std::ostream & gdcm::operator<< (std::ostream &os, const PrivateTag &val)`

## 28.192 gdcmProgressEvent.h File Reference

```
#include "gdcmEvent.h"
```

```
#include "gdcmTag.h"
```

Include dependency graph for `gdcmProgressEvent.h`:



## Classes

- class [gdcm::ProgressEvent](#)  
*ProgressEvent* Special type of event triggered during.

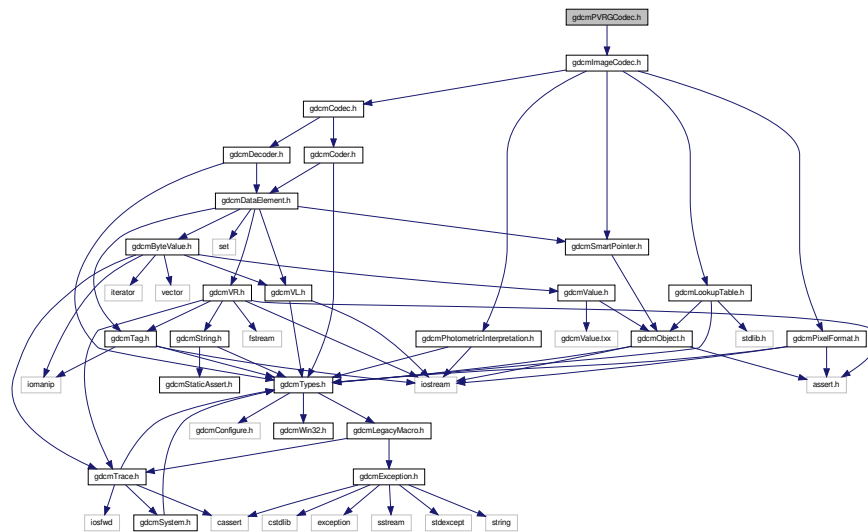
## Namespaces

- [gdcm](#)

## 28.193 gdcmPVRGCodec.h File Reference

```
#include "gdcmImageCodec.h"
```

Include dependency graph for gdcmPVRGCodec.h:



## Classes

- class [gdcm::PVRGCodec](#)

*PVRGCodec.*

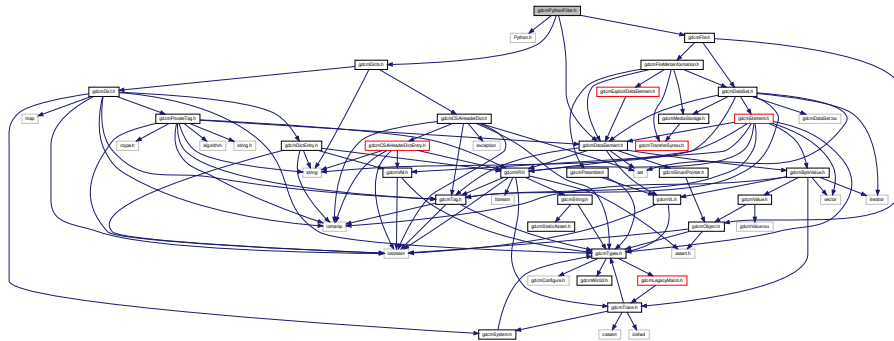
## Namespaces

- [gdcm](#)

## 28.194 gdcmPythonFilter.h File Reference

```
#include <Python.h>
#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"
```

Include dependency graph for `gdcmPythonFilter.h`:



## Classes

- class `gdcm::PythonFilter`

*`PythonFilter` `PythonFilter` is the class that make `gdcm2.x` looks more like `gdcm1` and transform the binary blob contained in a `DataElement` into a string, typically this is a nice feature to have for wrapped language.*

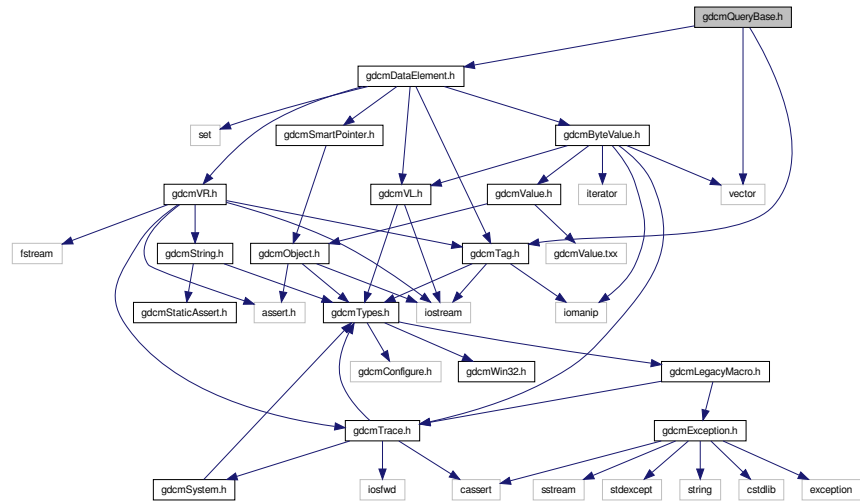
## Namespaces

- `gdcm`

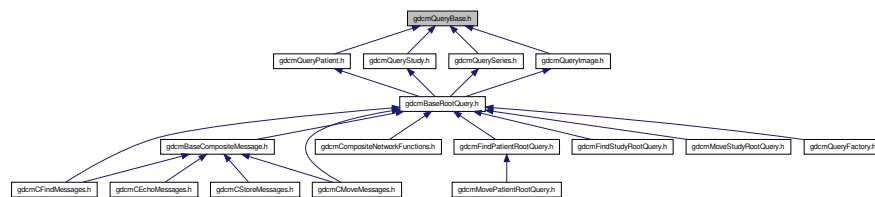
## 28.195 gdcmQueryBase.h File Reference

```
#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* contains: the base class for constructing a query dataset for a C-FIND and a C-MOVE.

## Namespaces

- [gdcm](#)

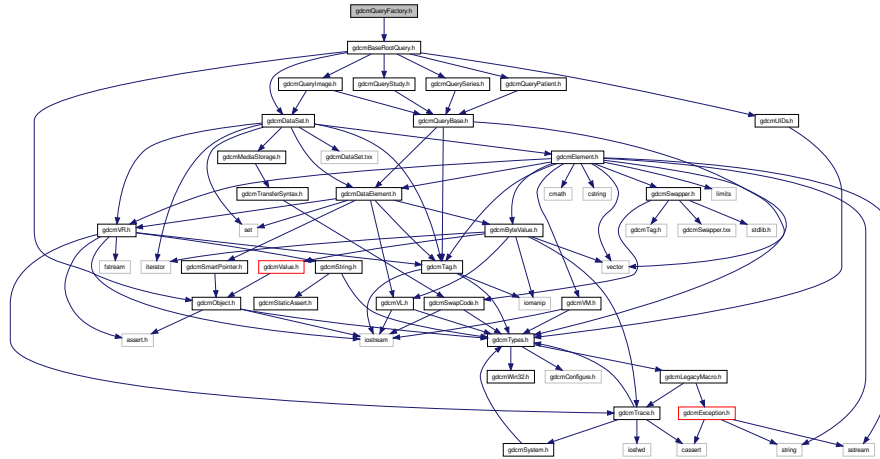
## Enumerations

- enum [gdcm::ERootType](#) {  
[gdcm::ePatientRootType](#),  
[gdcm::eStudyRootType](#) }

## 28.196 gdcmQueryFactory.h File Reference

```
#include "gdcmBaseRootQuery.h"
```

Include dependency graph for gdcMQueryFactory.h:



## Classes

- class `gdcm::QueryFactory`  
*QueryFactory.h.*

## Namespaces

- **gdcm**

## Enumerations

- 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` }







```

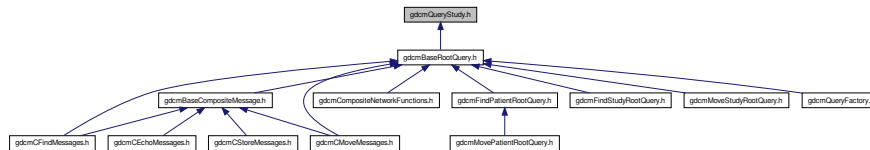
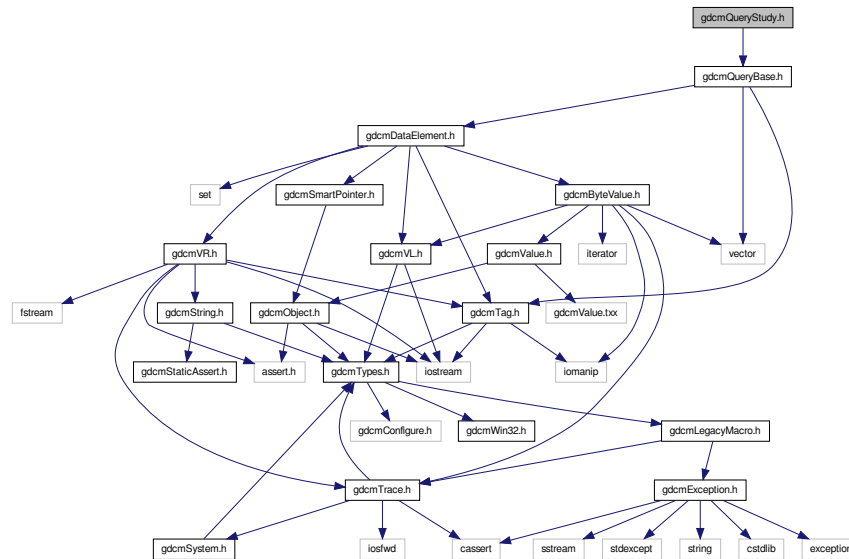
graph TD
    gsdmQueryStatistics[gsdmQueryStatistics.h] --> gsdmBaseFloatQuery[gsdmBaseFloatQuery.h]
    gsdmBaseFloatQuery --> gsdmBaseCompositeMessage[gsdmBaseCompositeMessage.h]
    gsdmBaseFloatQuery --> gsdmCompositeNetworkFunctions[gsdmCompositeNetworkFunctions.h]
    gsdmBaseFloatQuery --> gsdmFindParameterFloatQuery[gsdmFindParameterFloatQuery.h]
    gsdmBaseFloatQuery --> gsdmFindStudyFloatQuery[gsdmFindStudyFloatQuery.h]
    gsdmBaseFloatQuery --> gsdmMoveStudyFloatQuery[gsdmMoveStudyFloatQuery.h]
    gsdmBaseFloatQuery --> gsdmQueryFactory[gsdmQueryFactory]
    gsdmBaseCompositeMessage --> gsdmCFindMessages[gsdmCFindMessages.h]
    gsdmBaseCompositeMessage --> gsdmCFetchMessages[gsdmCFetchMessages.h]
    gsdmBaseCompositeMessage --> gsdmCStoreMessages[gsdmCStoreMessages.h]
    gsdmCompositeNetworkFunctions --> gsdmCMoveMessages[gsdmCMoveMessages.h]
    gsdmFindParameterFloatQuery --> gsdmMoveParameterFloatQuery[gsdmMoveParameterFloatQuery.h]
  
```

- class `gdcm::QuerySeries`

## Namespaces

- gdc

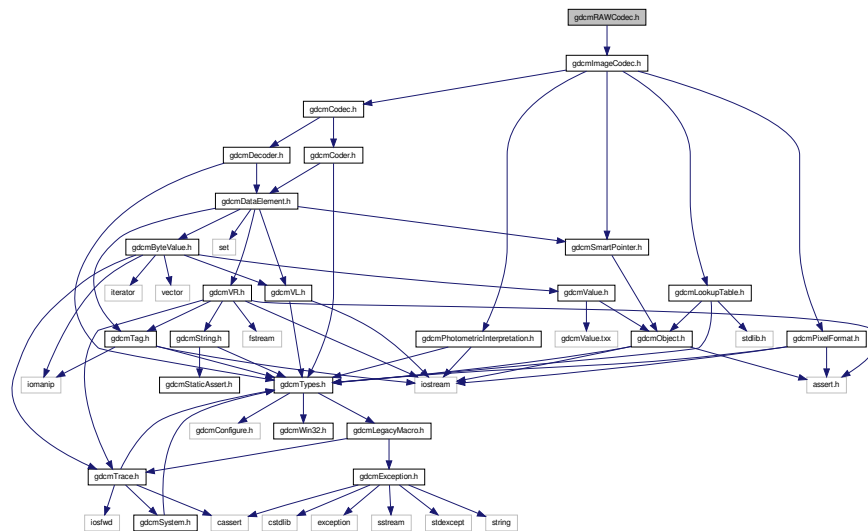
```
#include "gdcmQueryBase.h"
```



- QueryStudy.b contains:

---

Include dependency graph for gdcmRAWCodec.h:



## Classes

- class [gdcm::RAWCodec](#)

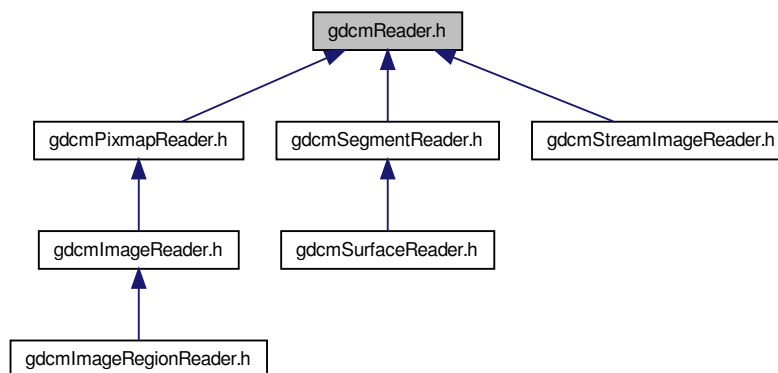
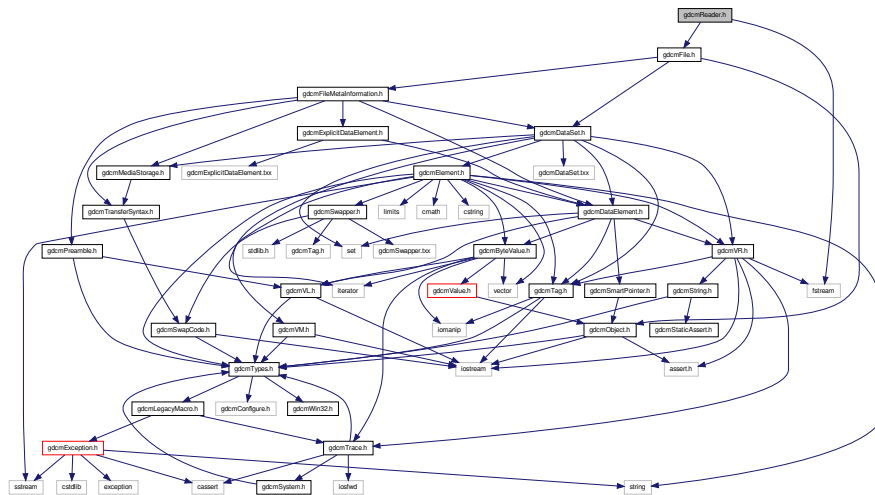
*RAWCodec* class.

## Namespaces

- [gdcm](#)

## 28.203 gdcmReader.h File Reference

```
#include "gdcmFile.h"
#include <fstream>
```

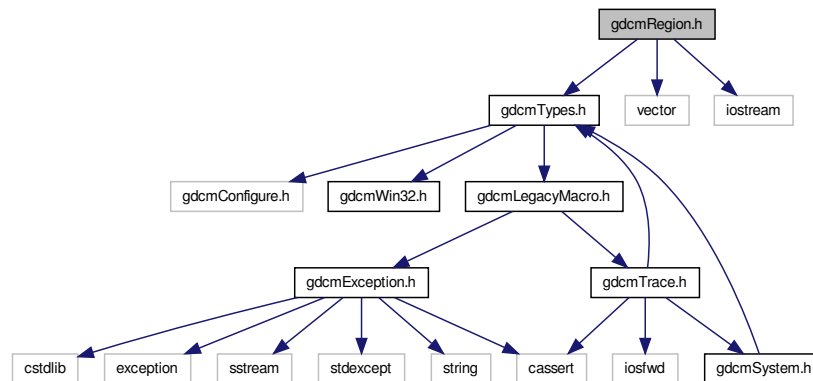


- class `gdcm::Reader`  
*Reader* ala *DOM* (*Document Object Model*)

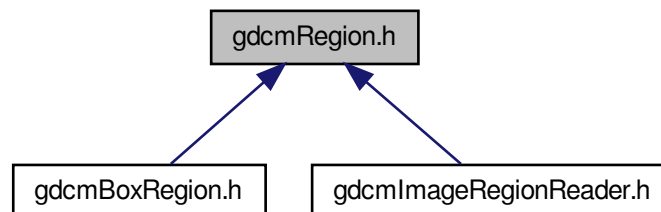
- gdc

## 28.204 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

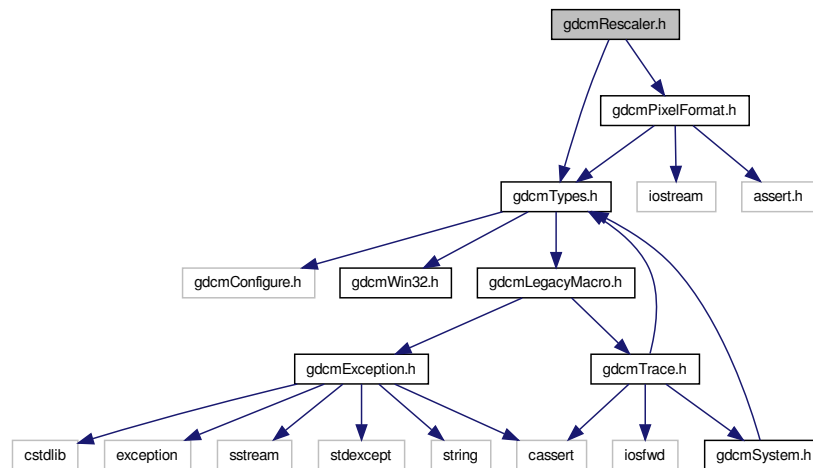
- `std::ostream & gdcm::operator<< (std::ostream &os, const Region &r)`

## 28.205 gdcmRescaler.h File Reference

```
#include "gdcmTypes.h"
```

```
#include "gdcmPixelFormat.h"
```

Include dependency graph for `gdcmRescaler.h`:



## Classes

- class [gdcm::Rescaler](#)

*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.*

## Namespaces

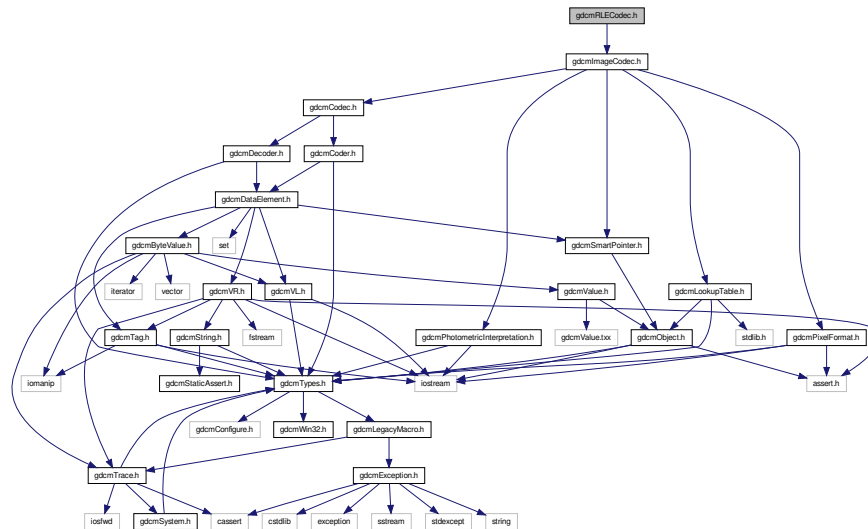
- [gdcm](#)

## 28.206 gdcmRLECodec.h File Reference

```
#include "gdcmImageCodec.h"
```



Include dependency graph for gdcmRLECodec.h:



## Classes

- class [gdcm::RLECodec](#)

*Class to do RLE.*

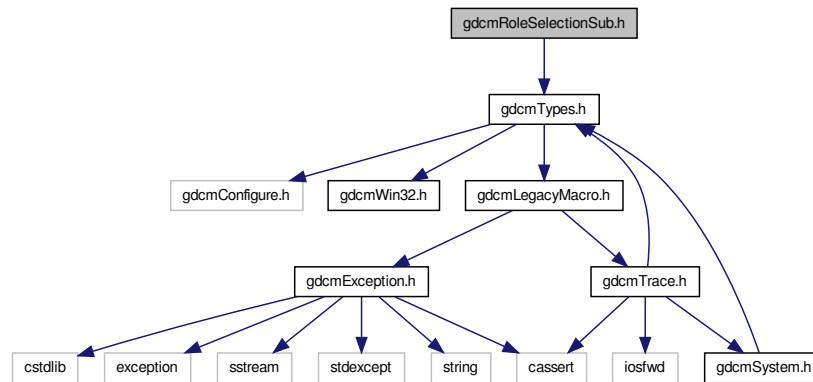
## Namespaces

- [gdcm](#)

## 28.207 gdcmRoleSelectionSub.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcRoleSelectionSub.h`:



## Classes

- class [gdc::network::RoleSelectionSub](#)

*RoleSelectionSub* PS 3.7 [Table D.3-9](#) SCP/SCU ROLE SELECTION SUB-ITEM FIELDS (A-ASSOCIATE-RQ)

## Namespaces

- [gdc](#)
- [gdc::network](#)

## 28.208 gdcScanner.h File Reference

```

#include "gdcDirectory.h"
#include "gdcSubject.h"
#include "gdcTag.h"
#include "gdcPrivateTag.h"
#include "gdcSmartPointer.h"
#include <map>
#include <set>
#include <string>
#include <string.h>

```

[illegible]

- struct `gdcm::Scanner::Itstr`
- class `gdcm::Scanner`

## Namespaces

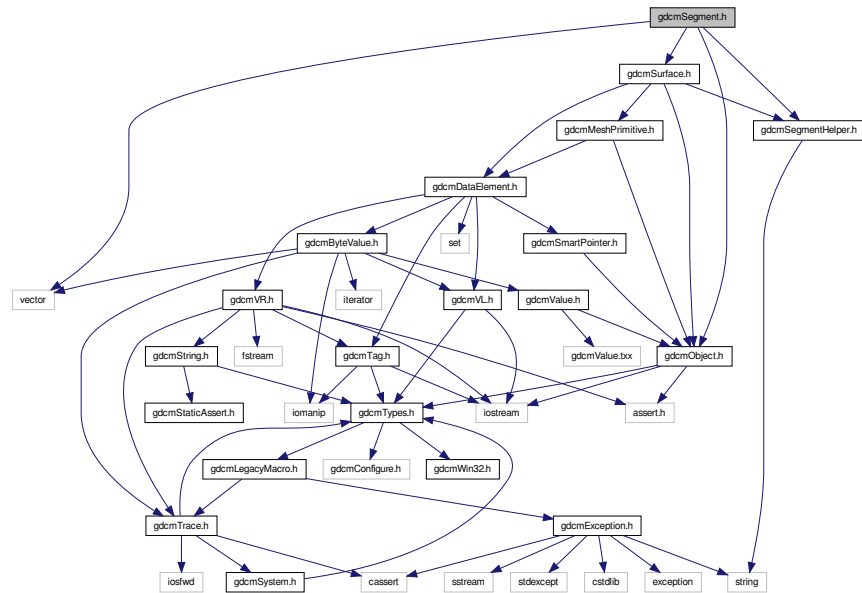
- **gdcm**

- `std::ostream & gdcmm::operator<< (std::ostream &os, const Scanner &s)`

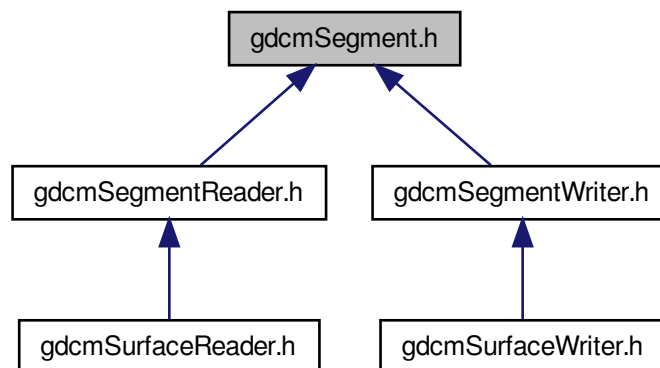
## 28.210 gdcmscu.man File Reference

```
#include <vector>
#include <gdcmObject.h>
#include <gdcmSurface.h>
#include "gdcmSegmentHelper.h"
```

Include dependency graph for `gdcSegment.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `gdc::Segment`

*This class defines a segment. It mainly contains attributes of group 0x0062. In addition, it can be associated with surface.*

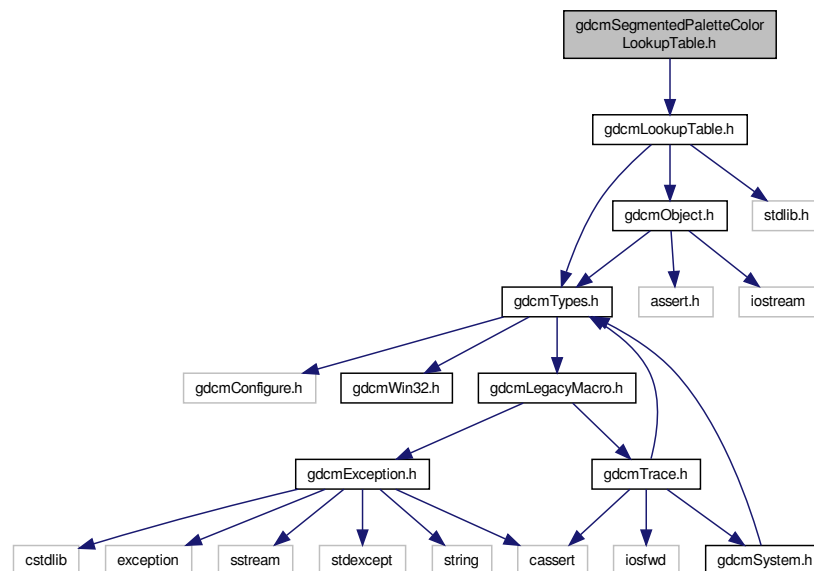
## Namespaces

- [gdcm](#)

## 28.212 gdcmSegmentedPaletteColorLookupTable.h File Reference

```
#include "gdcmLookupTable.h"
```

Include dependency graph for gdcmSegmentedPaletteColorLookupTable.h:



## Classes

- class [gdcm::SegmentedPaletteColorLookupTable](#)  
*SegmentedPaletteColorLookupTable* class.

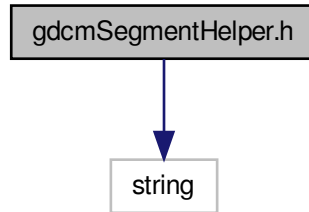
## Namespaces

- [gdcm](#)

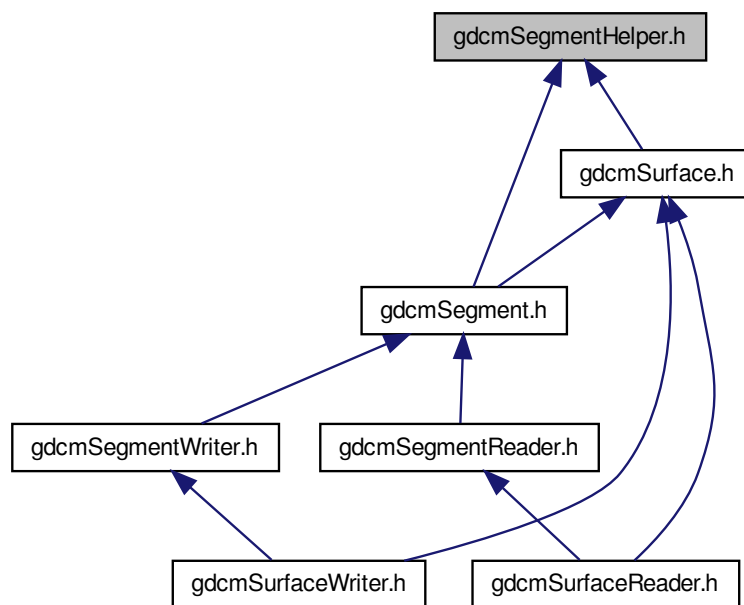
## 28.213 gdcmSegmentHelper.h File Reference

```
#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.*

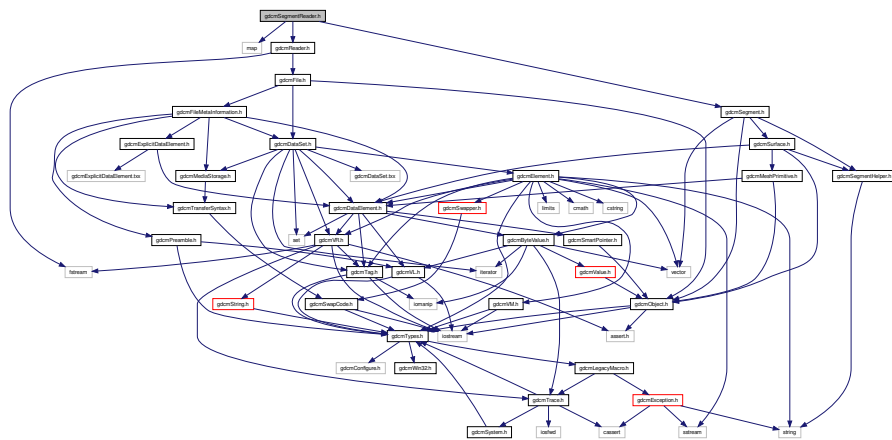
## Namespaces

- [gdcm](#)
- [gdcm::SegmentHelper](#)

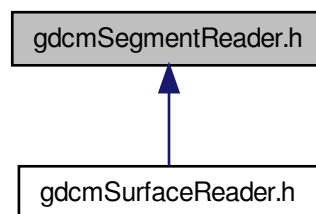
## 28.214 gdcmSegmentReader.h File Reference

```
#include <map>
#include <gdcmReader.h>
#include <gdcmSegment.h>
```

Include dependency graph for gdcmSegmentReader.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::SegmentReader](#)

*This class defines a segment reader. It reads attributes of group 0x0062.*

## Namespaces

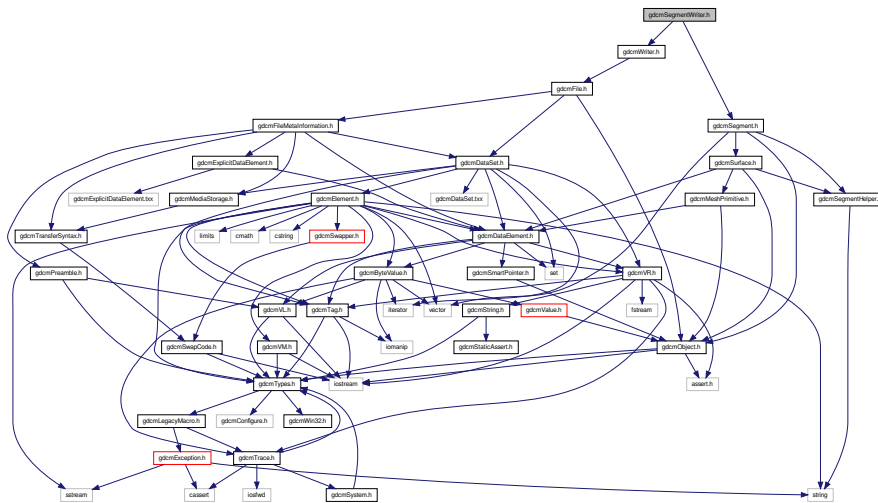
- [gdcm](#)

## 28.215 gdcmSegmentWriter.h File Reference

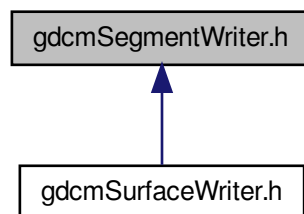
```
#include <gdcmWriter.h>
```

```
#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. It writes attributes of group 0x0062.*

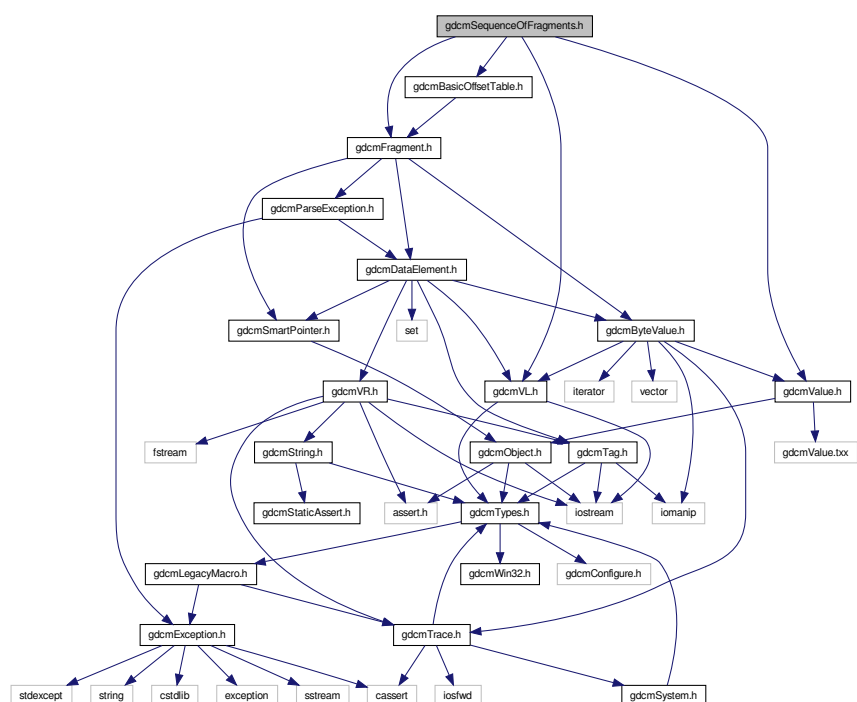


## Namespaces

- **gdcm**

## 28.216 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**



- class `gdcm::FileName`  
*FileName.*
- struct `gdcm::SerieHelper::Rule`
- class `gdcm::SerieHelper`

*SeriesHelper* 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.

- **gdcm**

- typedef bool(\* **gdcm::BOOL\_FUNCTION\_PFILE\_PFILE\_POINTER**)(File \*, File \*)
- typedef std::vector  
 <SmartPointer< FileWithName > > **gdcm::FileList**

- enum `gdcmm::CompOperators` {  
    `gdcmm::GDCM_EQUAL` = 0,  
    `gdcmm::GDCM_DIFFERENT`,  
    `gdcmm::GDCM_GREATER`,  
    `gdcmm::GDCM_GREATEROREQUAL`,  
    `gdcmm::GDCM_LESS`,  
    `gdcmm::GDCM_LESSEOREQUAL` }

```

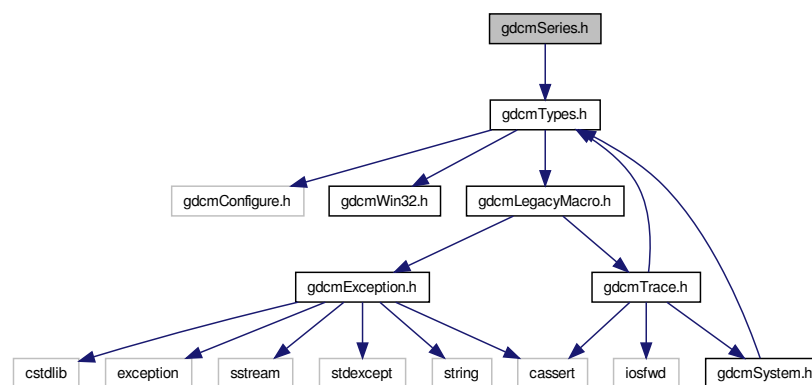
• enum gdcM::LodModeType {
  gdcM::LD_ALL = 0x00000000,
  gdcM::LD_NOSEQ = 0x00000001,
  gdcM::LD_NOSHADOW = 0x00000002,
  gdcM::LD_NOSHADOWSEQ = 0x00000004 }

```

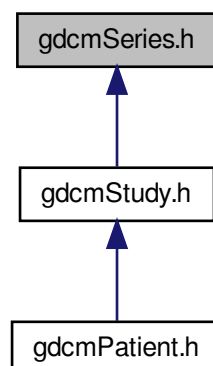
## 28.219 gdcMSeries.h File Reference

```
#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.*

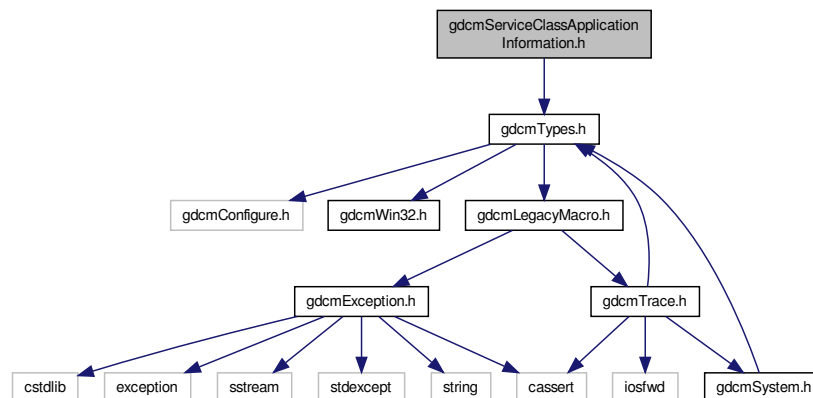
## Namespaces

- [gdcm](#)

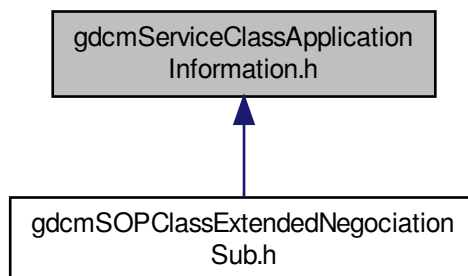
## 28.220 gdcmServiceClassApplicationInformation.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmServiceClassApplicationInformation.h:

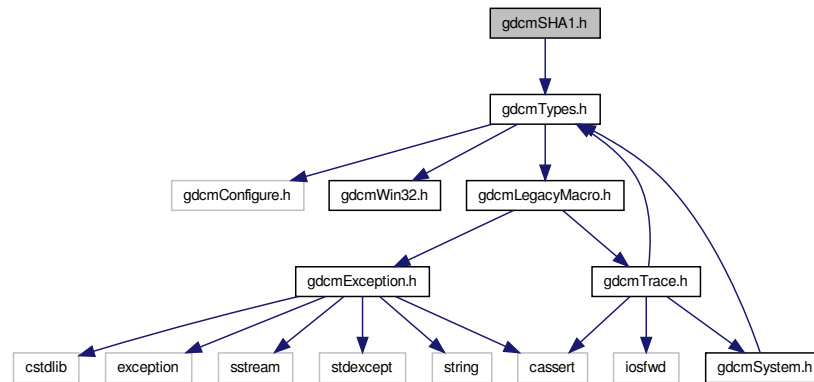


This graph shows which files directly or indirectly include this file:





Include dependency graph for gdcmSHA1.h:



## Classes

- class [gdcm::SHA1](#)

*Class for [SHA1](#).*

## Namespaces

- [gdcm](#)

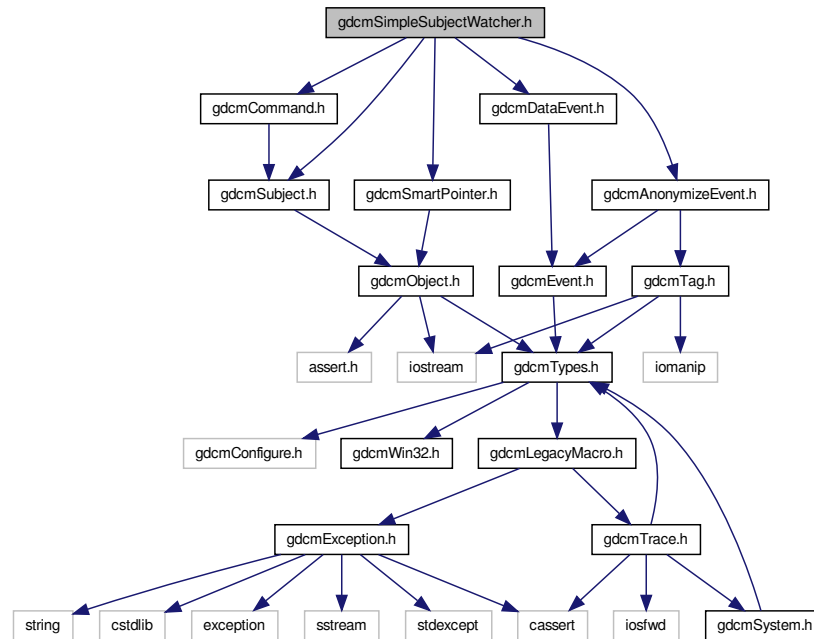
## 28.223 gdcmSimpleSubjectWatcher.h File Reference

```

#include "gdcmSubject.h"
#include "gdcmCommand.h"
#include "gdcmSmartPointer.h"
#include "gdcmAnonymizeEvent.h"
#include "gdcmDataEvent.h"

```

Include dependency graph for `gdcmsimpleSubjectWatcher.h`:



## Classes

- class `gdcmsimpleSubjectWatcher`

*`SimpleSubjectWatcher` This is a typical `Subject` Watcher class. It will observe all events.*

## Namespaces

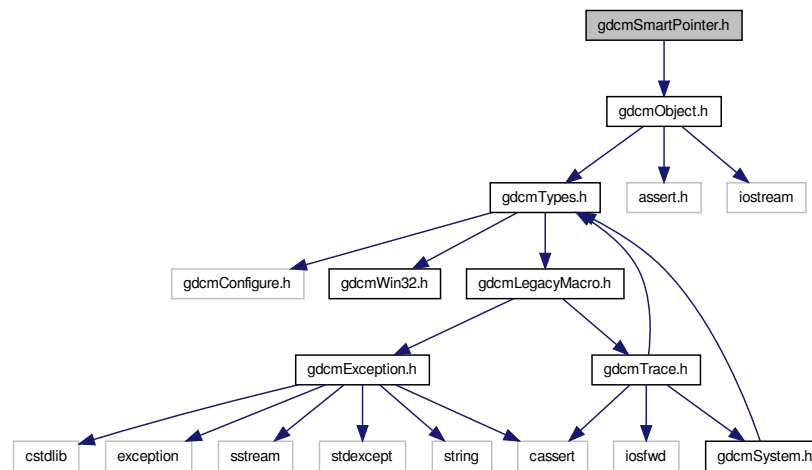
- `gdcmsimple`

## 28.224 `gdcmsmartPointer.h` File Reference

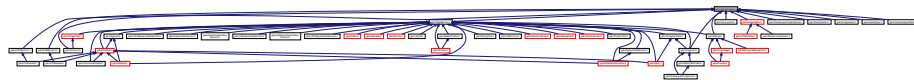
```
#include "gdcmsmartPointer.h"
```



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.*

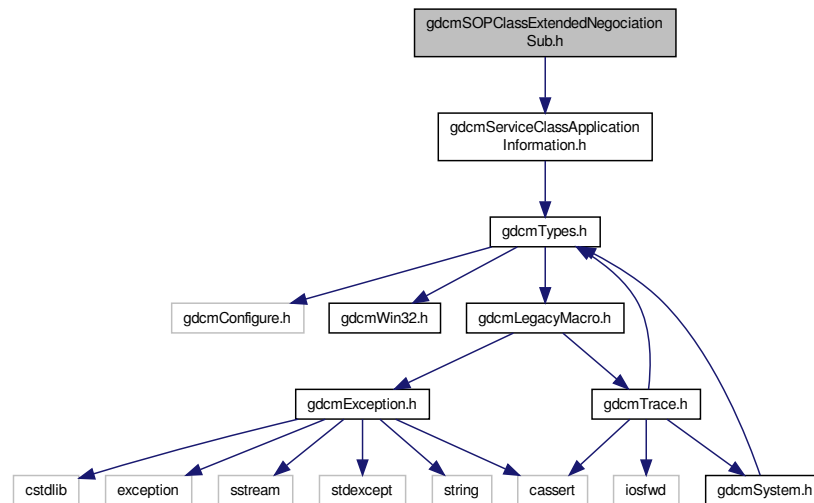
## Namespaces

- `gdcm`

## 28.225 gdcmSOPClassExtendedNegociationSub.h File Reference

```
#include "gdcmServiceClassApplicationInformation.h"
```

Include dependency graph for `gdcmSOPClassExtendedNegociationSub.h`:



## Classes

- class [gdcm::network::SOPClassExtendedNegociationSub](#)

[SOPClassExtendedNegociationSub](#) PS 3.7 [Table D.3-11](#) SOP CLASS EXTENDED NEGOTIATION SUB-ITEM FIELDS (A-ASSOCIATE-RQ and A-ASSOCIATE-AC)

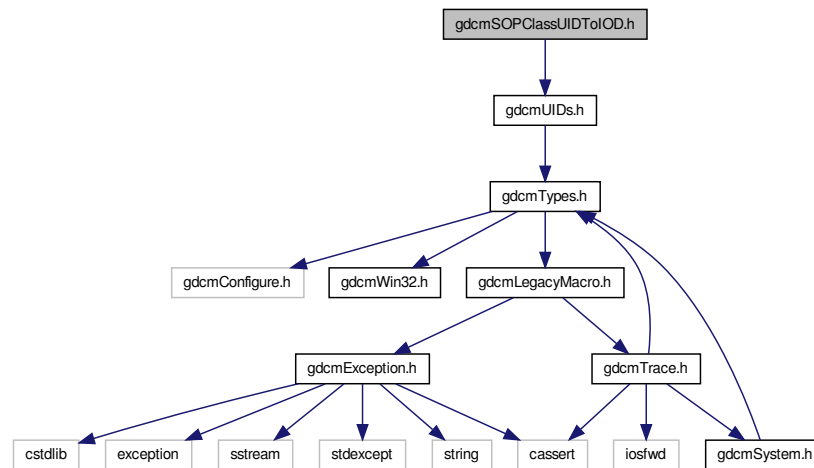
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.226 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](#)

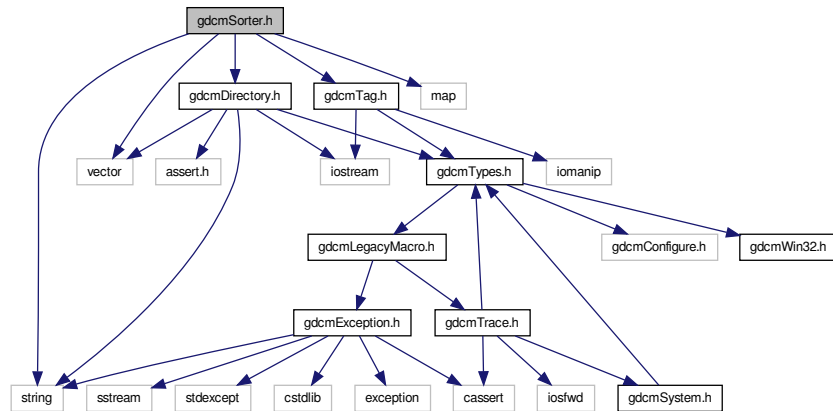
## 28.227 gdcmSorter.h File Reference

```

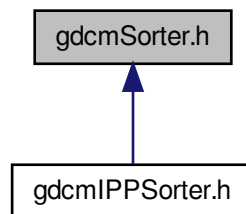
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include <vector>
#include <string>
#include <map>

```

Include dependency graph for `gdcmSorter.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Sorter](#)

*[Sorter](#) General class to do sorting using a custom function You simply need to provide a function of type: [Sorter::SortFunction](#).*

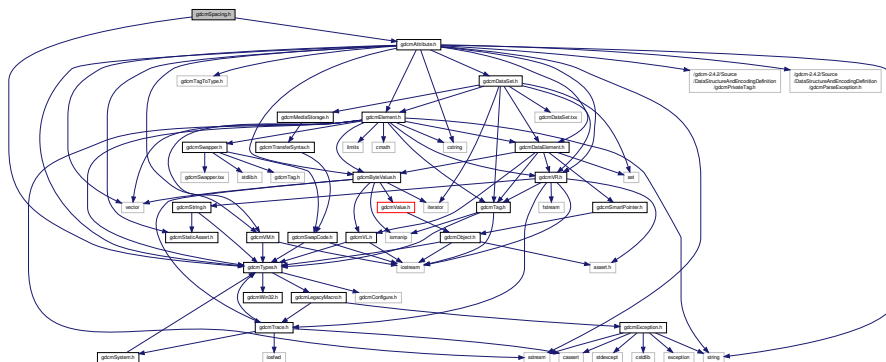
## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const Sorter &s)`

```
#include "gdcmTypes.h"
#include "gdcmAttribute.h"
Include dependency graph for gdcmSpacing.h:
```



- class `gdcm::Spacing`

Class for *Spacing*.

- **gdcm**

```
#include "gdcmFile.h"
```

- class `gdcm::Spectroscopy`  
*Spectroscopy* class.

- **gdcm**

```
#include "gdcmFile.h"
#include "gdcmImage.h"
```

## Classes

- class [gdcm::SplitMosaicFilter](#)

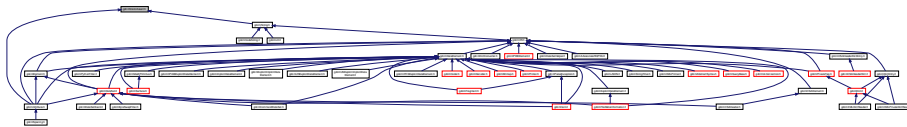
*[SplitMosaicFilter](#) class Class to reshuffle bytes for a SIEMENS Mosaic image Siemens CSA [Image](#) Header CSA:= Common Siemens Architecture, sometimes also known as Common syngo Architecture.*

## Namespaces

- [gdcm](#)

## 28.231 gdcmStaticAssert.h File Reference

This graph shows which files directly or indirectly include this file:



## Classes

- struct [gdcm::static\\_assert\\_test< x >](#)
- struct [gdcm::STATIC\\_ASSERTION\\_FAILURE< x >](#)
- struct [gdcm::STATIC\\_ASSERTION\\_FAILURE< true >](#)

## Namespaces

- [gdcm](#)

## 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.*

### 28.231.1 Macro Definition Documentation

28.231.1.1 #define GDCM\_DO\_JOIN( X, Y ) GDCM\_DO\_JOIN2(X,Y)

28.231.1.2 #define GDCM\_DO\_JOIN2( X, Y ) X##Y

28.231.1.3 #define GDCM\_JOIN( X, Y ) GDCM\_DO\_JOIN( X, Y )





[illegible]

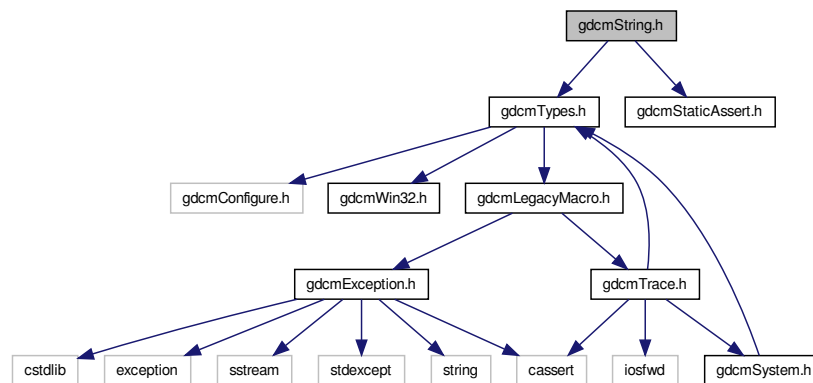
- class `gdcm::StreamImageWriter`

## Namespaces

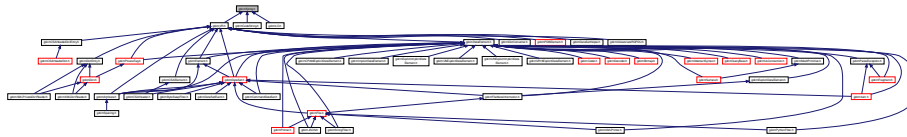
- **gdcm**

```
#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)`

## 28.235 gdcmStringFilter.h File Reference

```

#include "gdcmDataElement.h"
#include "gdcmDicts.h"
#include "gdcmFile.h"

```

- class `gdcm::StringFilter`

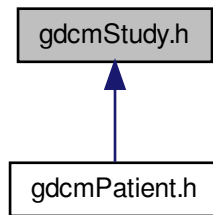
- **gdcm**

```
#include "gdcmTypes.h"
#include "gdcmSeries.h"
#include <vector>
```

```

graph TD
    gdcStudy.h --> gdcSeries.h
    gdcStudy.h --> vector
    gdcSeries.h --> gdcTypes.h
    gdcTypes.h --> gdcConfigure.h
    gdcTypes.h --> gdcWin32.h
    gdcTypes.h --> gdcLegacyMacro.h
    gdcLegacyMacro.h --> gdcException.h
    gdcLegacyMacro.h --> gdcTrace.h
    gdcException.h --> cstdlib
    gdcException.h --> exception
    gdcException.h --> sstream
    gdcException.h --> stdexcept
    gdcException.h --> string
    gdcException.h --> cassert
    gdcTrace.h --> iosfwd
    gdcTrace.h --> gdcSystem.h
    gdcSystem.h --> gdcTypes.h
  
```

This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Study](#)  
[Study](#).

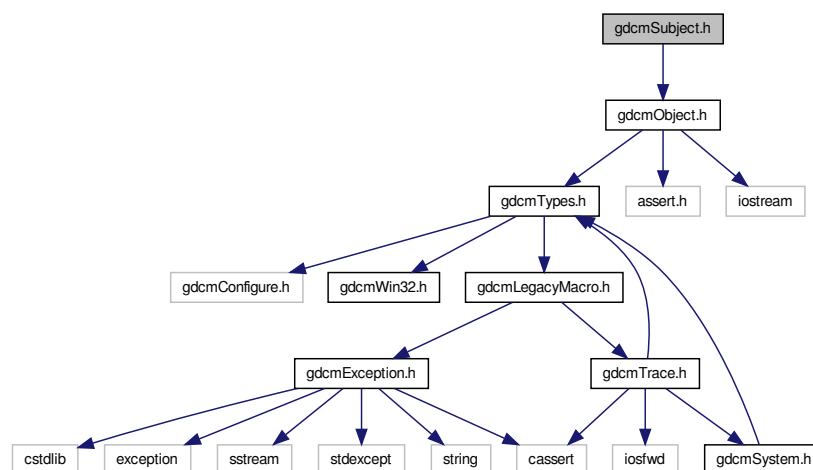
## Namespaces

- [gdcm](#)

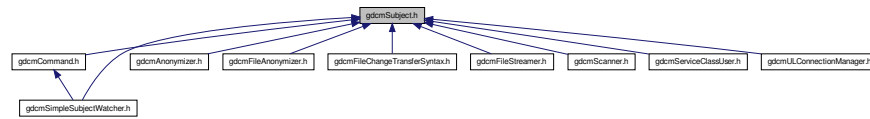
## 28.237 gdcmSubject.h File Reference

```
#include "gdcmObject.h"
```

Include dependency graph for gdcmSubject.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Subject](#)

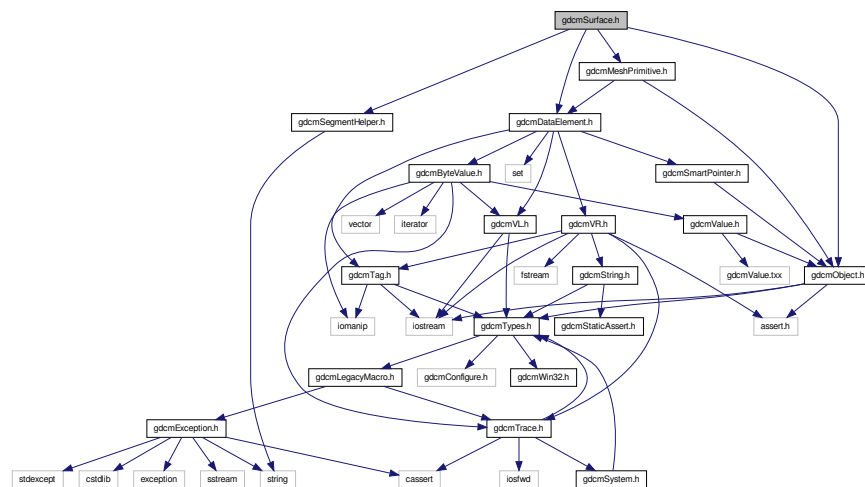
*Subject.*

## Namespaces

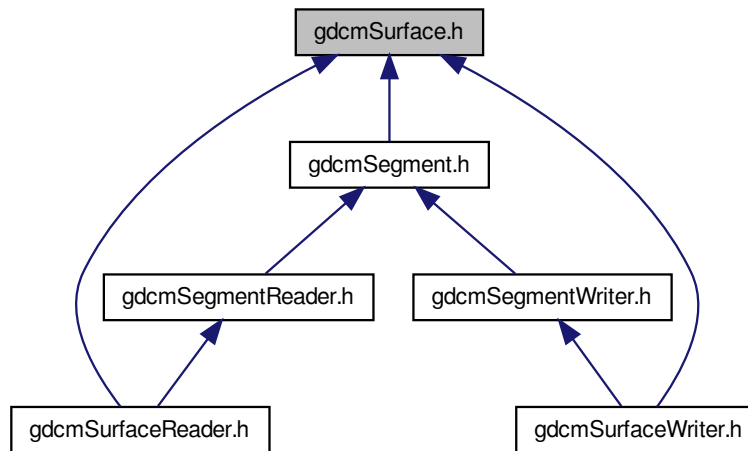
- [gdcm](#)

## 28.238 gdcmSurface.h File Reference

```
#include <gdcmObject.h>
#include <gdcmDataElement.h>
#include <gdcmMeshPrimitive.h>
#include "gdcmSegmentHelper.h"
Include dependency graph for gdcmSurface.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcms::Surface](#)

*This class defines a SURFACE IE. This members are taken from required surface mesh module attributes.*

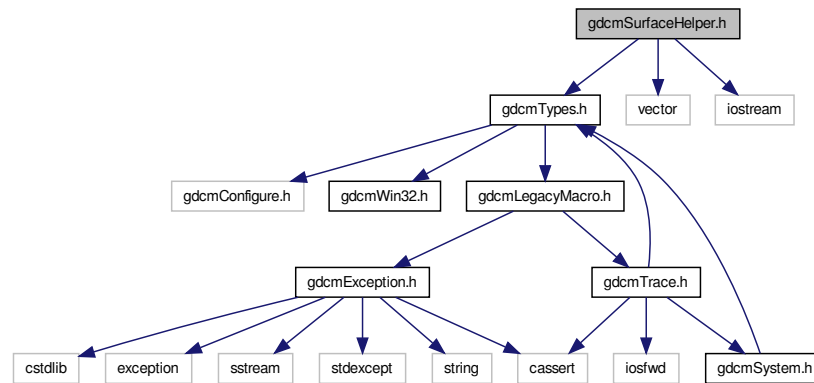
## Namespaces

- [gdcms](#)

## 28.239 gdcmsurfacehelper.h File Reference

```
#include "gdcmtypes.h"
#include <vector>
#include <iostream>
```

Include dependency graph for gdcmSurfaceHelper.h:



## Classes

- class [gdcm::SurfaceHelper](#)

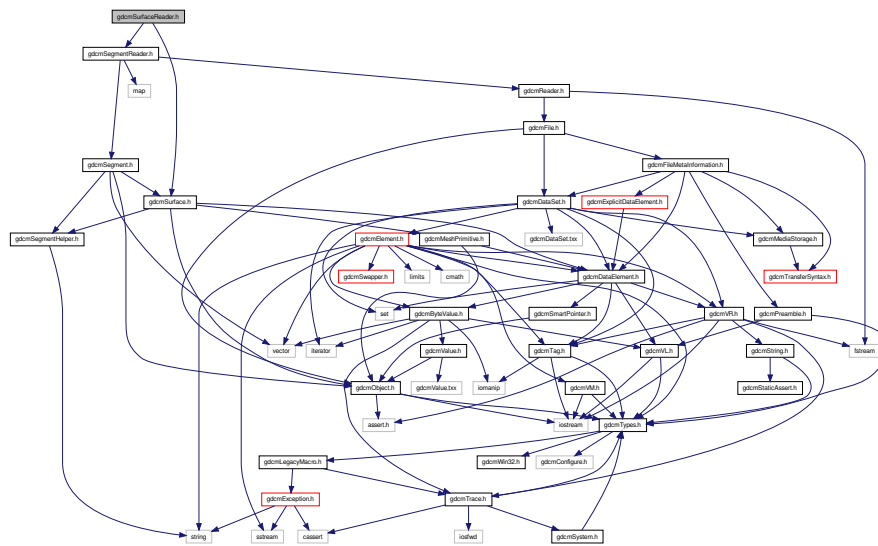
*[SurfaceHelper](#) Helper class for [Surface](#) object.*

## Namespaces

- [gdcm](#)

## 28.240 gdcmSurfaceReader.h File Reference

```
#include <gdcmSegmentReader.h>
#include <gdcmSurface.h>
```



- class `gdcm::SurfaceReader`

*This class defines a SURFACE IE reader. It reads surface mesh module attributes.*

- **gdcm**

## 28.241 gdcmSurfaceWriter.h File Reference

```
#include <gdcmSegmentWriter.h>
#include <gdcmSurface.h>
```



[illegible]

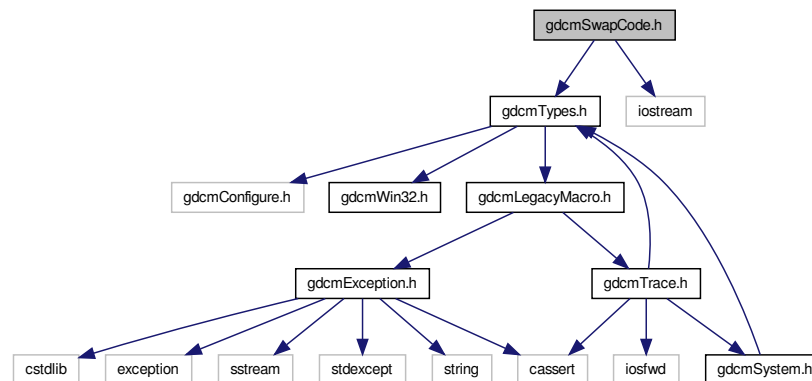
- class `gdcm::SurfaceWriter`

## Namespaces

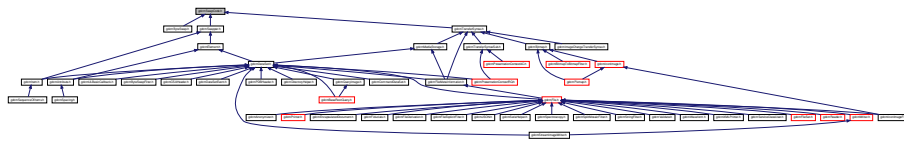
- **gdcm**

```
#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`

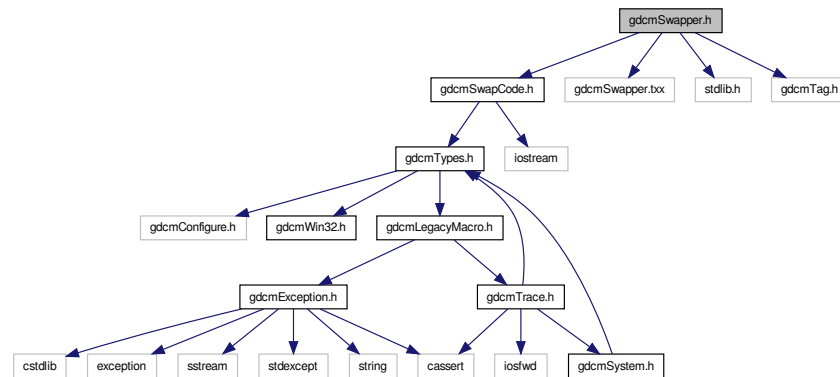
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &os, const SwapCode &sc)`

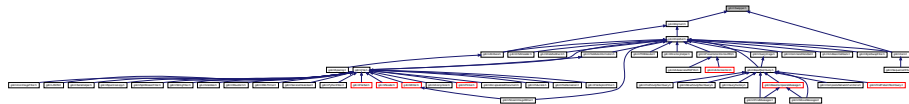
## 28.243 gdcmSwapper.h File Reference

```
#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

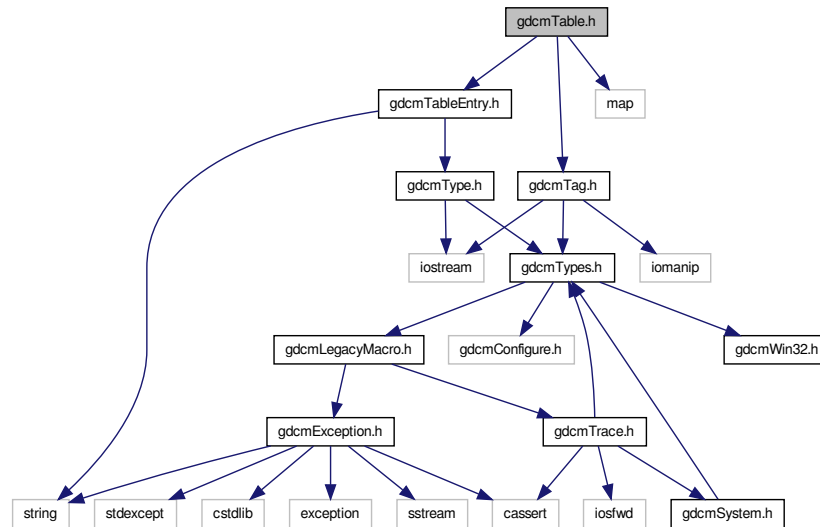
- [gdcm](#)

## 28.244 gdcmSystem.h File Reference

```
#include "gdcmTypes.h"
```



Include dependency graph for gdcmTable.h:



## Classes

- class [gdcm::Table](#)

*Table.*

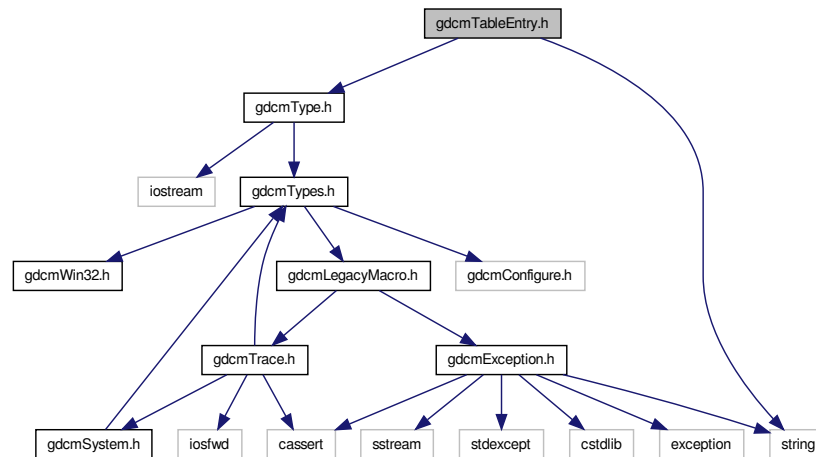
## Namespaces

- [gdcm](#)

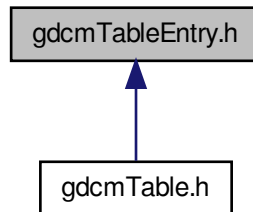
## 28.246 gdcmTableEntry.h File Reference

```
#include "gdcmType.h"
#include <string>
```

Include dependency graph for `gdcTableEntry.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdc::TableEntry](#)  
*TableEntry*.

## Namespaces

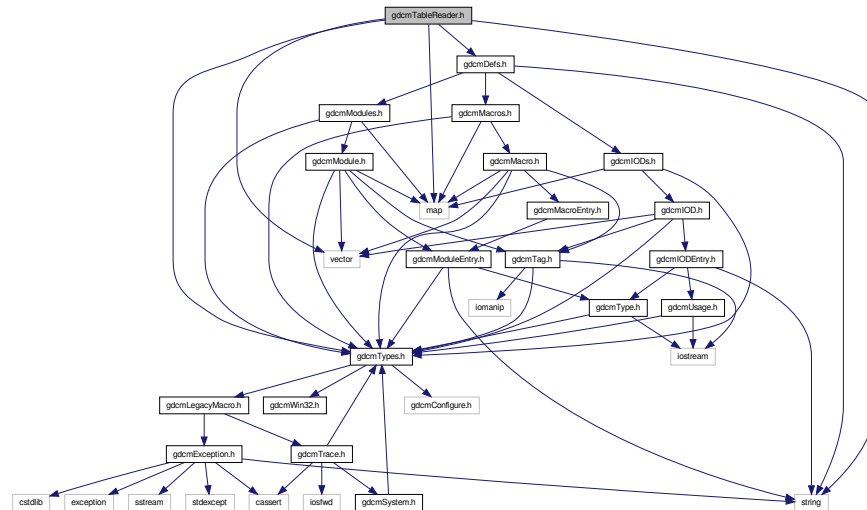
- [gdc](#)

## 28.247 gdcTableReader.h File Reference

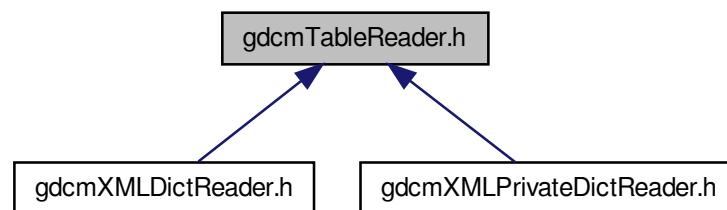
```
#include "gdcTypes.h"
```

```
#include "gdcmDefs.h"
#include <string>
#include <vector>
#include <map>
```

Include dependency graph for gdcmTableReader.h:



This graph shows which files directly or indirectly include this file:



## Classes

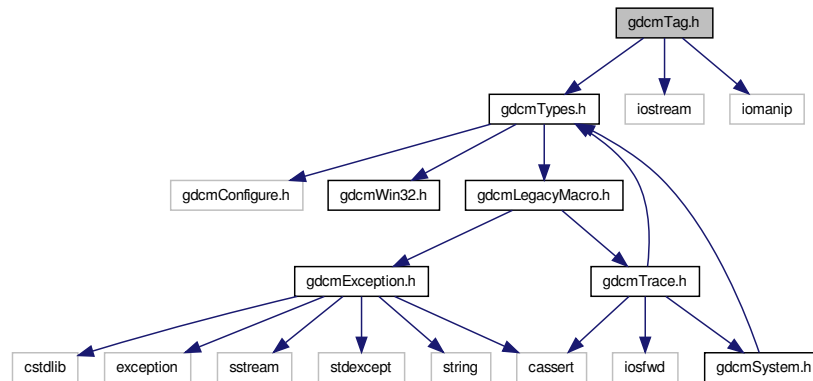
- class [gdcm::TableReader](#)  
Class for representing a *TableReader*.

## Namespaces

- [gdcm](#)

## 28.248 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*). Basically an `uint32_t` which can also be expressed as two `uint16_t` (group and element)

## Namespaces

- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Tag &_val)`
- `std::istream & gdcm::operator>> (std::istream &_is, Tag &_val)`

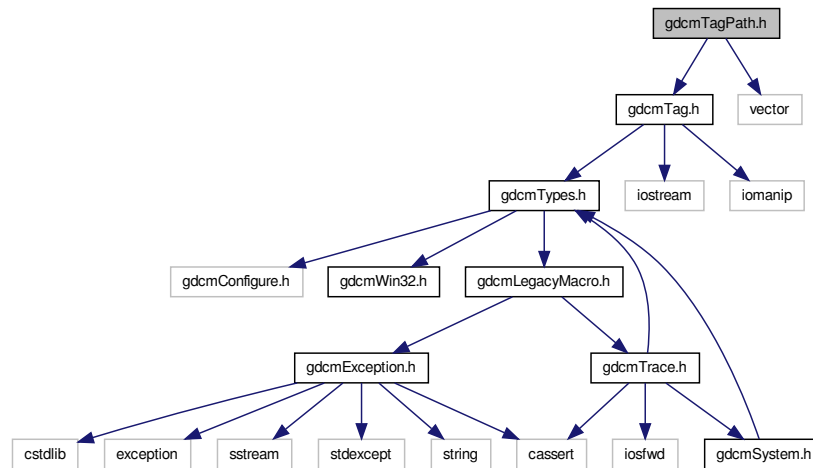


## 28.249 gdcmTagPath.h File Reference

```
#include "gdcmTag.h"
```

```
#include <vector>
```

Include dependency graph for gdcmTagPath.h:



### Classes

- class `gdcm::TagPath`  
class to handle a path of tag.

### Namespaces

- `gdcm`

## 28.250 gdcmTagToVR.h File Reference

### Namespaces

- `gdcm`

### Functions

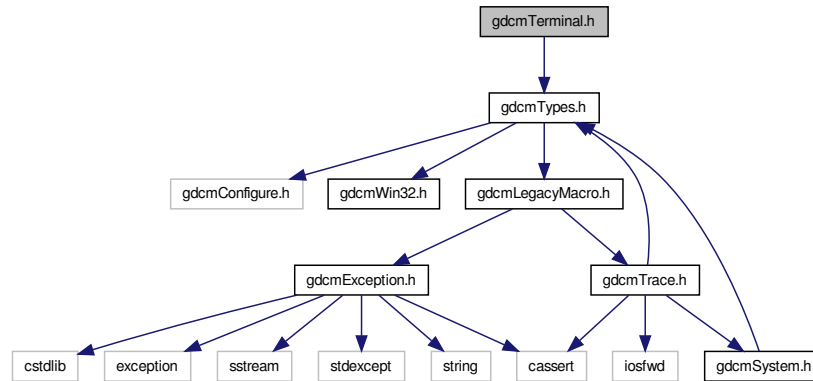
- VR::VRType `gdcm::GetVRFromTag` (Tag const &tag)

## 28.251 gdcmtar.man File Reference

## 28.252 gdcTerminal.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcTerminal.h:



### Namespaces

- [gdcm](#)
- [gdcm::terminal](#)

*Class for Terminal Allow one to print in color in a shell.*

### Enumerations

- enum [gdcm::terminal::Attribute](#) {  
[gdcm::terminal::reset](#) = 0,  
[gdcm::terminal::bright](#) = 1,  
[gdcm::terminal::dim](#) = 2,  
[gdcm::terminal::underline](#) = 3,  
[gdcm::terminal::blink](#) = 5,  
[gdcm::terminal::reverse](#) = 7,  
[gdcm::terminal::hidden](#) = 8 }
- enum [gdcm::terminal::Color](#) {  
[gdcm::terminal::black](#) = 0,  
[gdcm::terminal::red](#),  
[gdcm::terminal::green](#),  
[gdcm::terminal::yellow](#),  
[gdcm::terminal::blue](#),  
[gdcm::terminal::magenta](#),  
[gdcm::terminal::cyan](#),  
[gdcm::terminal::white](#) }
- enum [gdcm::terminal::Mode](#) {  
[gdcm::terminal::CONSOLE](#) = 0,  
[gdcm::terminal::VT100](#) }

## Functions

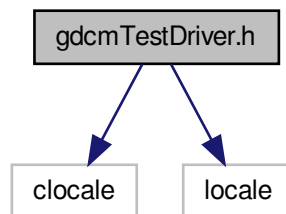
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setAttribute` (Attribute att)
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setbgcolor` (Color c)
- `GDCM_EXPORT` `std::string` `gdcm::terminal::setfgcolor` (Color c)
- `GDCM_EXPORT` `void` `gdcm::terminal::setmode` (Mode m)

## 28.253 gdcmTestDriver.h File Reference

```
#include <clocale>
```

```
#include <locale>
```

Include dependency graph for `gdcmTestDriver.h`:

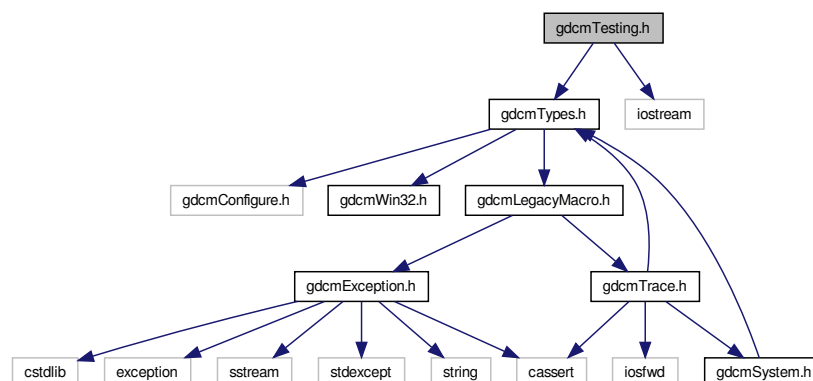


## 28.254 gdcmTesting.h File Reference

```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for `gdcmTesting.h`:



## Classes

- class [gdcm::Testing](#)  
*class for testing*

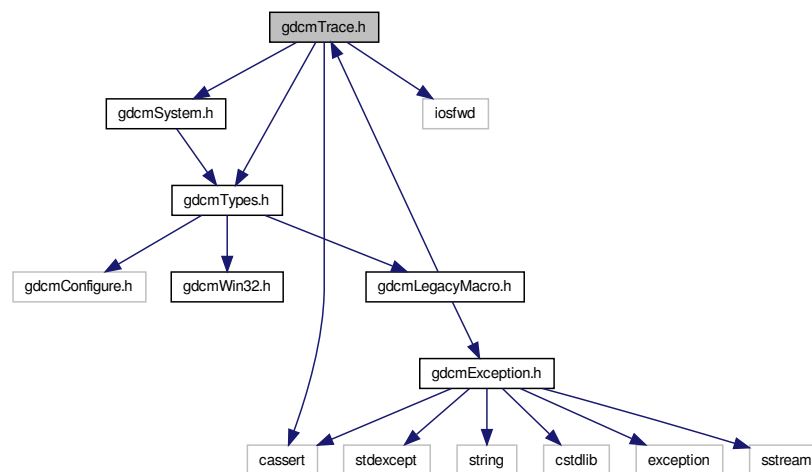
## Namespaces

- [gdcm](#)

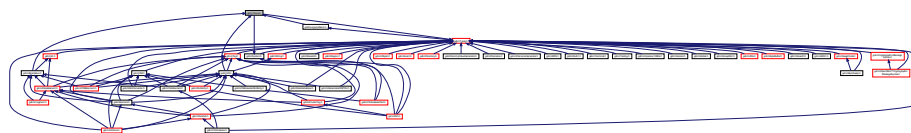
## 28.255 gdcmTrace.h File Reference

```
#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.*

### 28.255.1 Macro Definition Documentation

28.255.1.1 #define [GDCM\\_FUNCTION](#) "<unknown>"

28.255.1.2 #define [gdcmAssertAlwaysMacro](#)( arg ) [gdcmAssertMacro](#)(arg)

[AssertAlways.](#)

Parameters

<i>arg</i>	argument to test An easy solution to pass also a message is to do: <a href="#">gdcmAssertMacro</a> ( "my message" && 2 < 3 )
------------	--

Referenced by [gdcm::SequenceOfFragments::ReadValue\(\)](#), and [gdcm::VR::Write\(\)](#).

28.255.1.3 #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>gdcmaAssertMacro( "my message" &amp;&amp; 2 &lt; 3 )</code>
------------	--

Referenced by `gdcma::PixelFormat::SetSamplesPerPixel()`.

28.255.1.4 `#define gdcmaDebugMacro( msg )`

## Value:

```
{
    if( gdcma::Trace::GetDebugFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Debug: In " __FILE__ ", line " << __LINE__
            << ", function " << GDCM_FUNCTION << '\n'
            << "Last system error was: "
            << gdcma::System::GetLastError() << '\n' << msg;
        std::ostream &_os = gdcma::Trace::GetDebugStream();
        _os << osmacro.str() << "\n\n" << std::endl;
    }
}
```

Debug.

## Parameters

<i>msg</i>	message part
------------	--------------

Referenced by `gdcma::ByteValue::ByteValue()`, `gdcma::OpenSSLCryptoFactory::OpenSSLCryptoFactory()`, `gdcma::OpenSSLP7CryptoFactory::OpenSSLP7CryptoFactory()`, `gdcma::BasicOffsetTable::Read()`, `gdcma::SequenceOfItems::Read()`, `gdcma::Item::Read()`, `gdcma::VR::Read()`, `gdcma::SequenceOfFragments::ReadPreValue()`, `gdcma::SequenceOfFragments::ReadValue()`, and `gdcma::ByteValue::SetLength()`.

28.255.1.5 `#define gdcmaErrorMacro( msg )`

## Value:

```
{
    if( gdcma::Trace::GetErrorFlag() )
    {
        std::ostringstream osmacro;
        osmacro << "Error: In " __FILE__ ", line " << __LINE__
            << ", function " << GDCM_FUNCTION << '\n'
            << msg << "\n\n";
        std::ostream &_os = gdcma::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 `gdcma::CommandDataSet::Insert()`, `gdcma::FileMetaInformation::Insert()`, `gdcma::DataSet::Insert()`, `gdcma::Item::Read()`, and `gdcma::Fragment::ReadBacktrack()`.

28.255.1.6 `#define gdcmaWarningMacro( 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

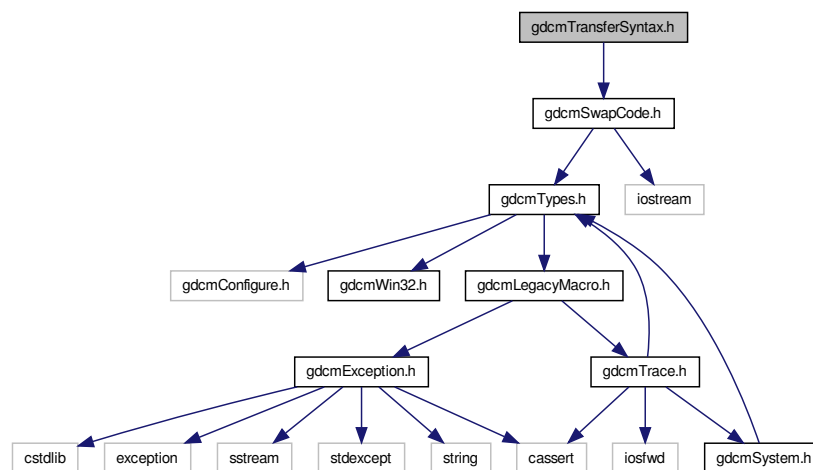
<i>msg</i>	message part
------------	--------------

Referenced by `gdcm::DataSet::InsertDataElement()`, `gdcm::SequenceOfItems::Read()`, `gdcm::Item::Read()`, `gdcm::Fragment::ReadBacktrack()`, `gdcm::Fragment::ReadValue()`, `gdcm::SequenceOfFragments::ReadValue()`, `gdcm::OpenSSL7CryptographicMessageSyntax::SetPassword()`, and `gdcm::Item::Write()`.

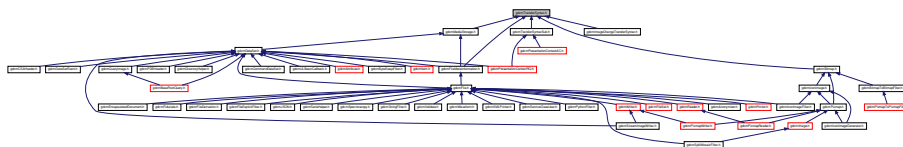
## 28.256 gdcmTransferSyntax.h File Reference

```
#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

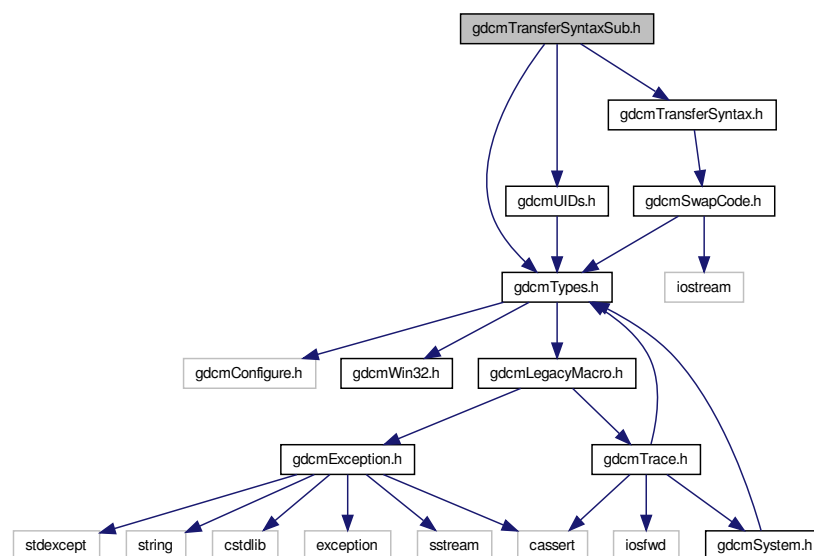
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const TransferSyntax &ts)`

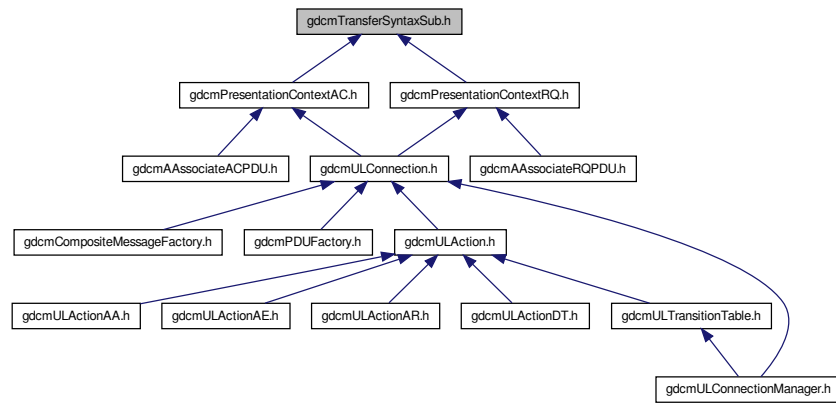
## 28.257 gdcmTransferSyntaxSub.h File Reference

```
#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* Table 9-15 TRANSFER SYNTAX SUB-ITEM FIELDS.

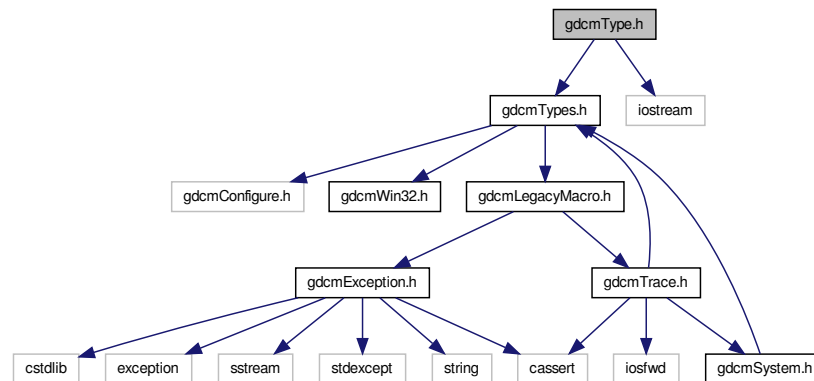
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

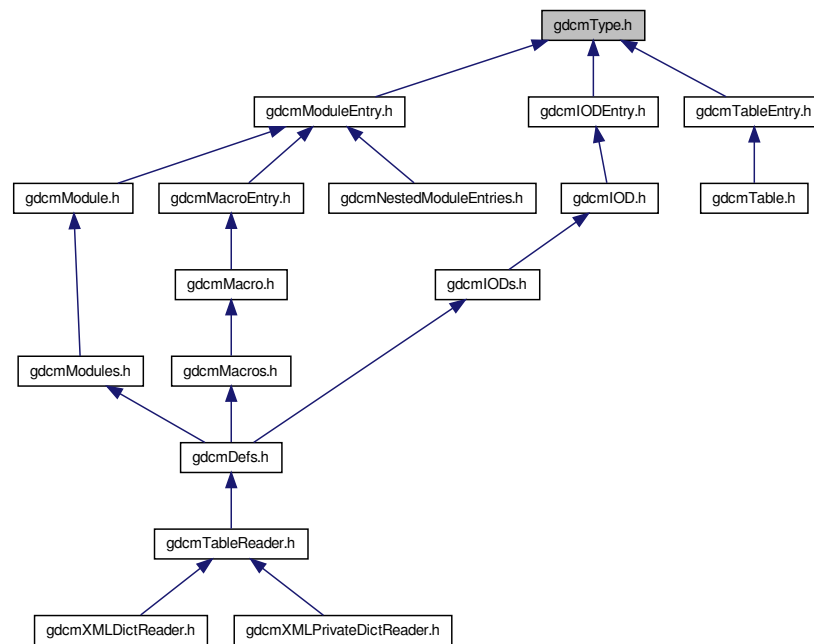
## 28.258 gdcmType.h File Reference

```
#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

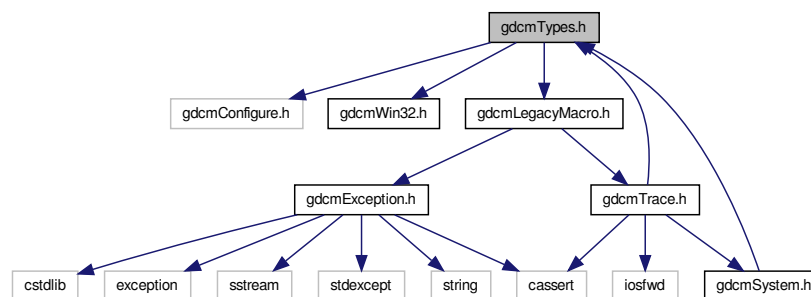
- [gdcm](#)

## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Type &val)`

## 28.259 gdcmTypes.h File Reference

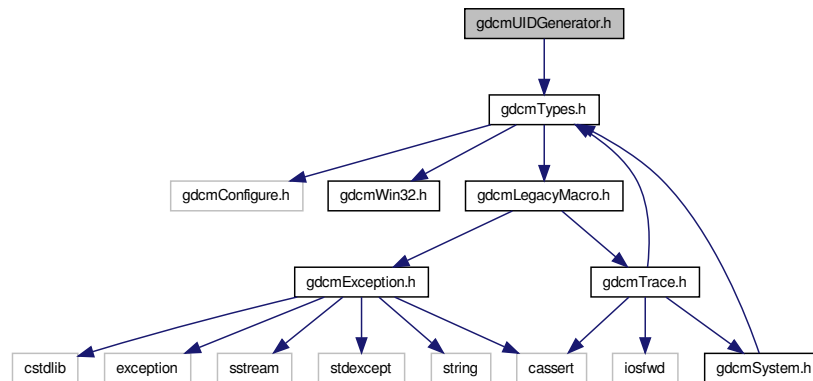
```
#include "gdcmConfigure.h"
#include "gdcmWin32.h"
#include "gdcmLegacyMacro.h"
Include dependency graph for gdcmTypes.h:
```



## 28.260 gdcmUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmUIDGenerator.h`:



## Classes

- class [gdcm::UIDGenerator](#)  
Class for generating unique UID.

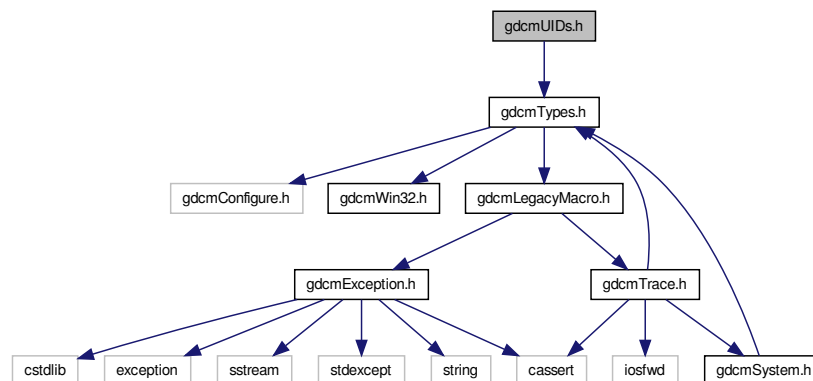
## Namespaces

- [gdcm](#)

## 28.261 gdcmUIDs.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for `gdcmUIDs.h`:



[illegible]

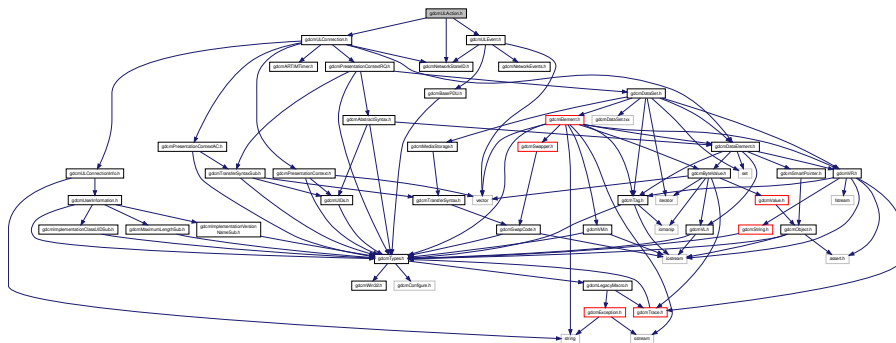
- class `gdcm::UIDs`

## Namespaces

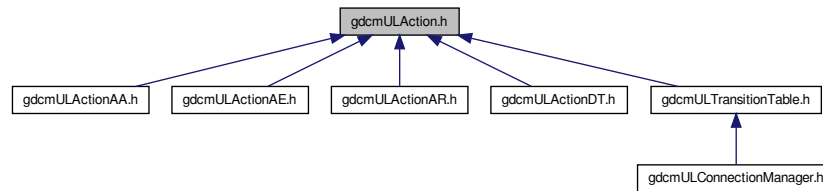
- gdc

- `std::ostream & gdcm::operator<< (std::ostream &_os, const UIDs &uid)`

```
#include "gdcmmNetworkStateID.h"
#include "gdcmmULError.h"
#include "gdcmmULConnection.h"
Include dependency graph for gdcmmULAction.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::network::ULAction](#)

*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*.

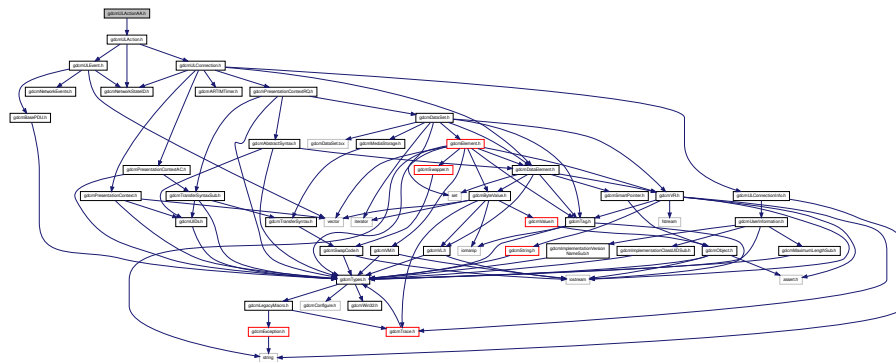
## Namespaces

- [gdcml](#)
- [gdcml::network](#)

## 28.263 gdcmlActionAA.h File Reference

```
#include "gdcmlAction.h"
```

Include dependency graph for `gdcmlActionAA.h`:



## Classes

- class [gdcml::network::ULActionAA1](#)
- class [gdcml::network::ULActionAA2](#)
- class [gdcml::network::ULActionAA3](#)
- class [gdcml::network::ULActionAA4](#)

- class [gdcm::network::ULActionAA5](#)
- class [gdcm::network::ULActionAA6](#)
- class [gdcm::network::ULActionAA7](#)
- class [gdcm::network::ULActionAA8](#)

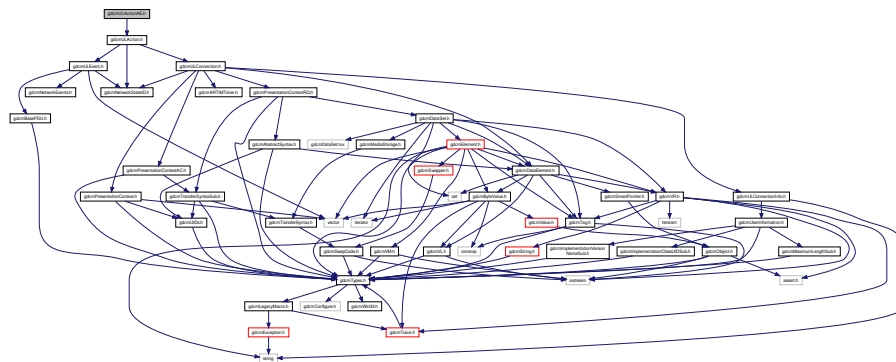
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.264 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](#)

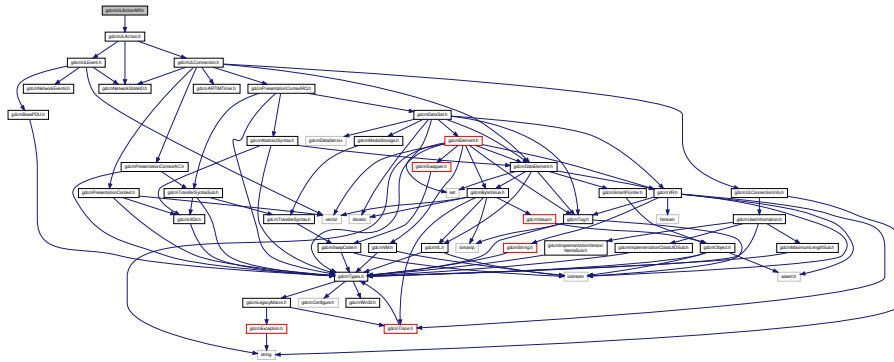
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.265 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

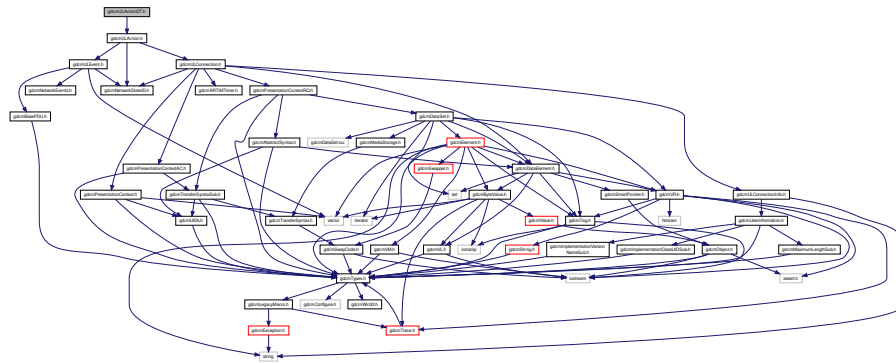
- [gdcm](#)
- [gdcm::network](#)

## 28.266 gdcmULActionDT.h File Reference

```
#include "gdcmULAction.h"
```



Include dependency graph for gdcmULActionDT.h:



## Classes

- class [gdcm::network::ULActionDT1](#)
- class [gdcm::network::ULActionDT2](#)

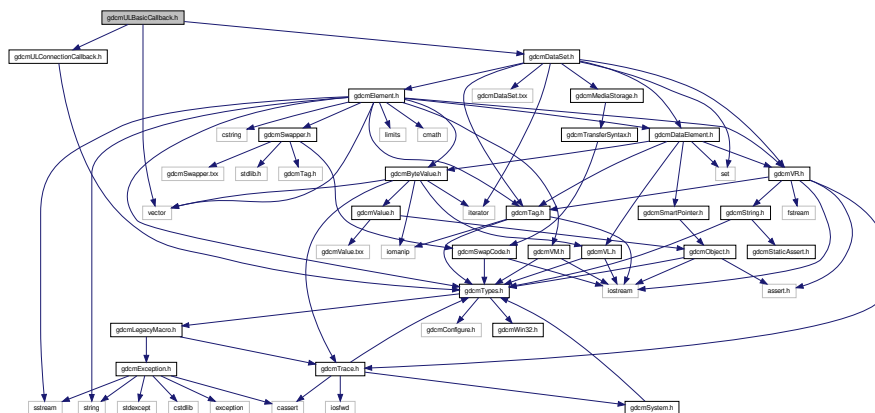
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.267 gdcmULBasicCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
#include "gdcmDataSet.h"
#include <vector>
```

Include dependency graph for gdcmULBasicCallback.h:



## Classes

- class [gdcm::network::ULBasicCallback](#)

*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](#).

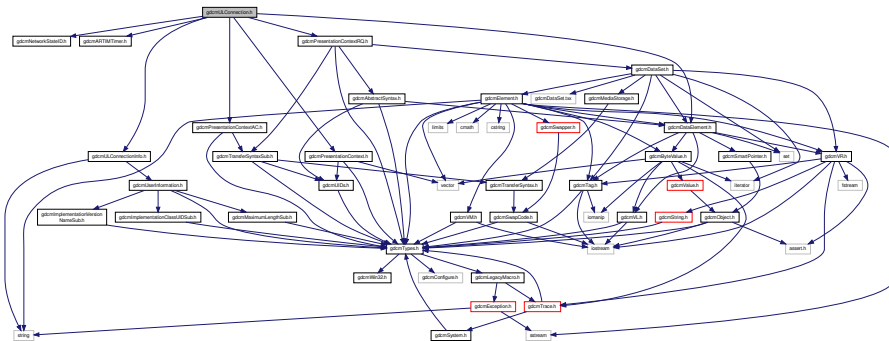
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

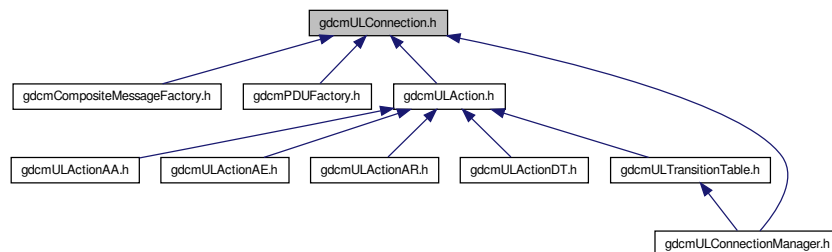
## 28.268 gdcmULConnection.h File Reference

```
#include "gdcmNetworkStateID.h"
#include "gdcmARTIMTimer.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationContextRQ.h"
#include "gdcmDataElement.h"
#include "gdcmPresentationContextAC.h"
#include "gdcmPresentationContext.h"
```

Include dependency graph for `gdcmULConnection.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnection](#)

*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.

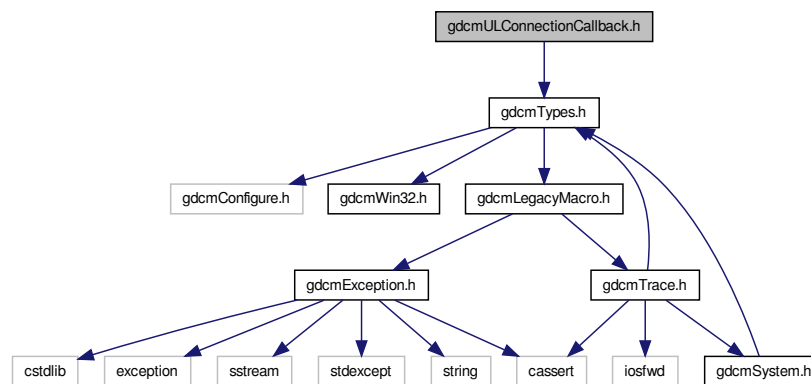
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

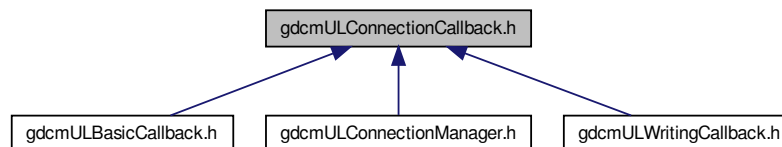
## 28.269 gdcmULConnectionCallback.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmULConnectionCallback.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnectionCallback](#)

## Namespaces

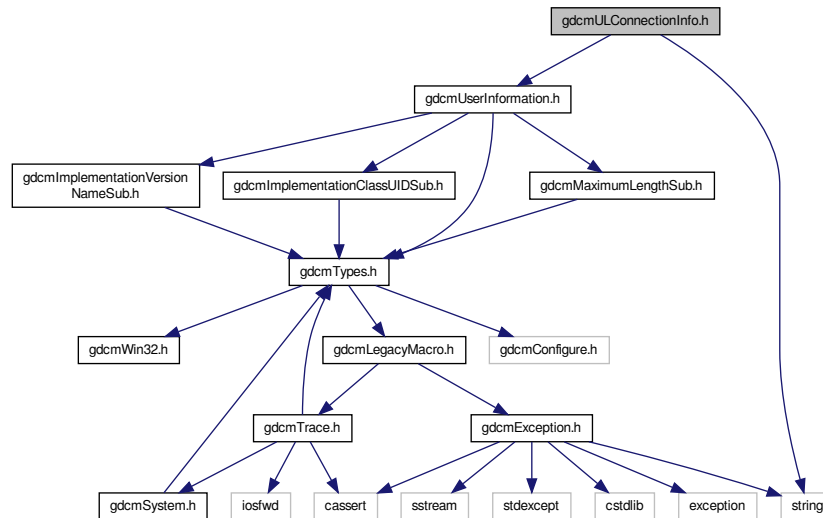
- [gdcm](#)
- [gdcm::network](#)

## 28.270 gdcmULConnectionInfo.h File Reference

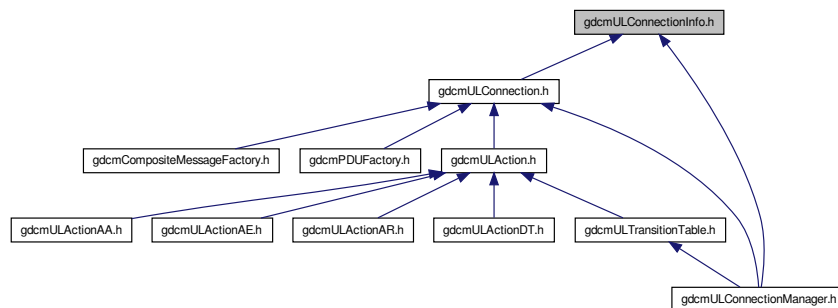
```
#include "gdcmUserInformation.h"
```

```
#include <string>
```

Include dependency graph for gdcmULConnectionInfo.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::network::ULConnectionInfo](#)

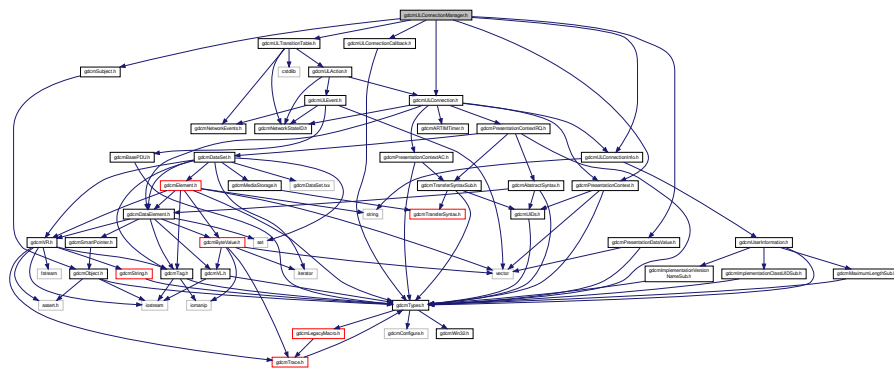
*ULConnectionInfo* this class contains all the information about a particular connection as established by the user. That is, it's: User Information Calling AE Title Called AE Title IP address/computer name IP Port A connection must be established with this information, that's subsequently placed into various primitives for actual communication.

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.271 gdcmULConnectionManager.h File Reference

```
#include "gdcmULTransitionTable.h"
#include "gdcmULConnection.h"
#include "gdcmULConnectionInfo.h"
#include "gdcmPresentationDataValue.h"
#include "gdcmULConnectionCallback.h"
#include "gdcmSubject.h"
#include "gdcmPresentationContext.h"
Include dependency graph for gdcmULConnectionManager.h:
```



## Classes

- class [gdcm::network::ULConnectionManager](#)

*ULConnectionManager* 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).

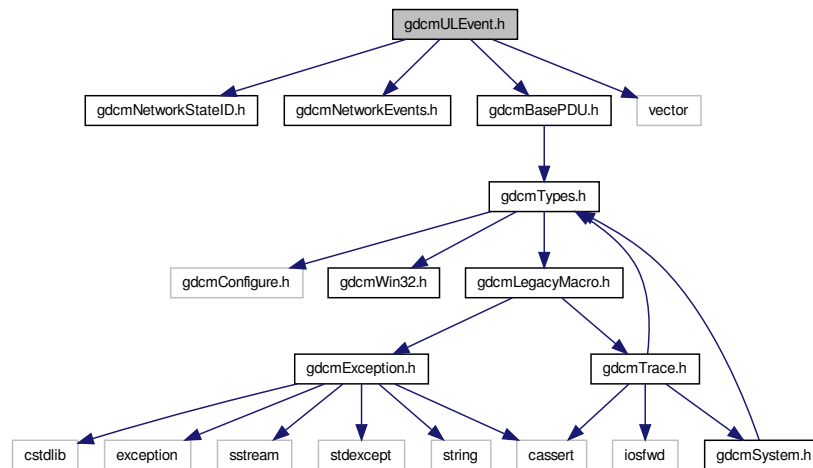
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

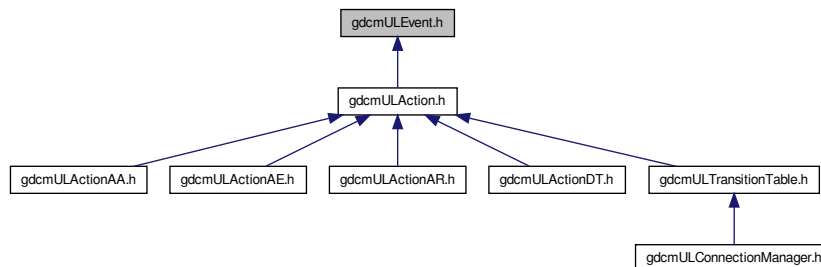
## 28.272 gdcmULEvent.h File Reference

```
#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* base class for network events.

### Namespaces

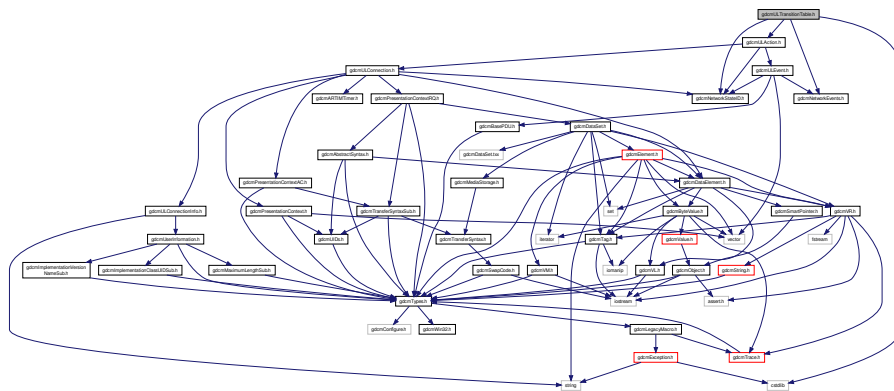
- [gdcm](#)

- [gdcm::network](#)

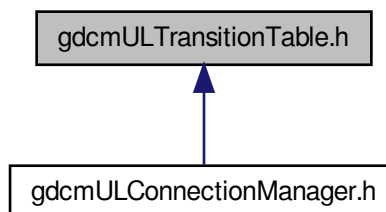
## 28.273 gdcmULTransitionTable.h File Reference

```
#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](#)

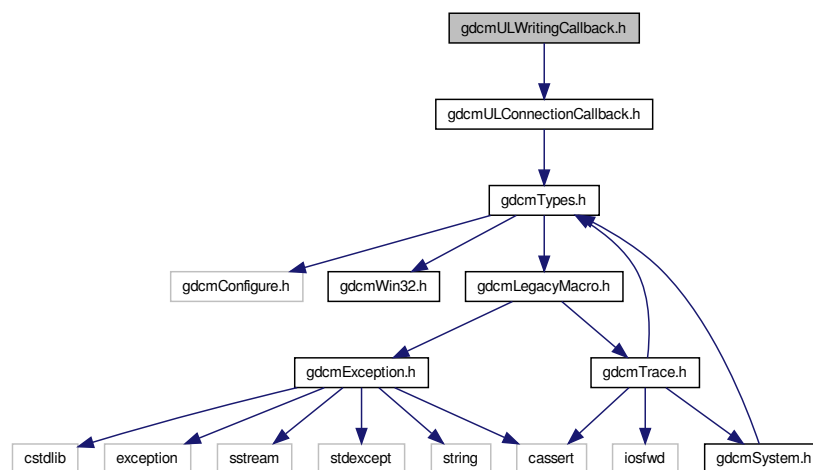
*[ULTransitionTable](#) The transition table of all the ULEvents, new ULActions, and ULStates.*

## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.274 gdcmULWritingCallback.h File Reference

```
#include "gdcmULConnectionCallback.h"
Include dependency graph for gdcmULWritingCallback.h:
```



## Classes

- class [gdcm::network::ULWritingCallback](#)

## Namespaces

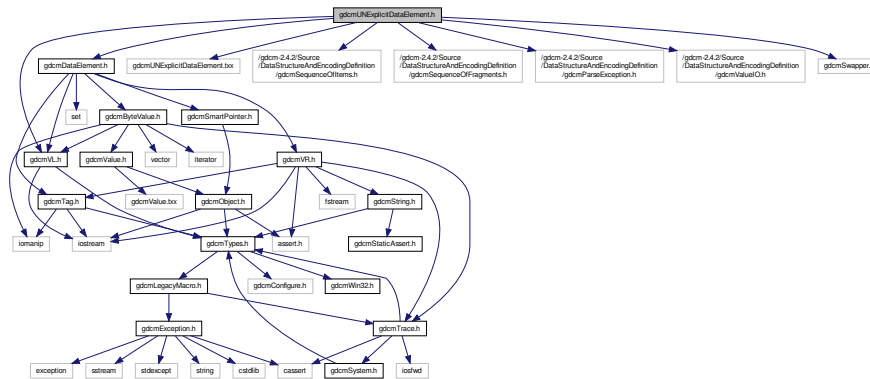
- [gdcm](#)
- [gdcm::network](#)

## 28.275 gdcmUNExplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitDataElement.txx"
```



Include dependency graph for gdcmUNExplicitDataElement.h:



## Classes

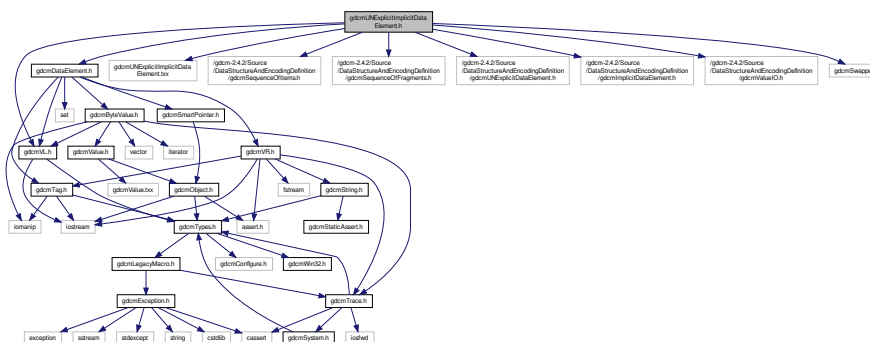
- class [gdcm::UNExplicitDataElement](#)  
Class to read/write a [DataElement](#) as [UNExplicit Data Element](#).

## Namespaces

- [gdcm](#)

## 28.276 gdcmUNExplicitImplicitDataElement.h File Reference

```
#include "gdcmDataElement.h"
#include "gdcmUNExplicitImplicitDataElement.hxx"
Include dependency graph for gdcmUNExplicitImplicitDataElement.h:
```



## Classes

- class [gdcm::UNExplicitImplicitDataElement](#)

Class to read/write a [DataElement](#) as ExplicitImplicit Data [Element](#) This class gather two known bugs:

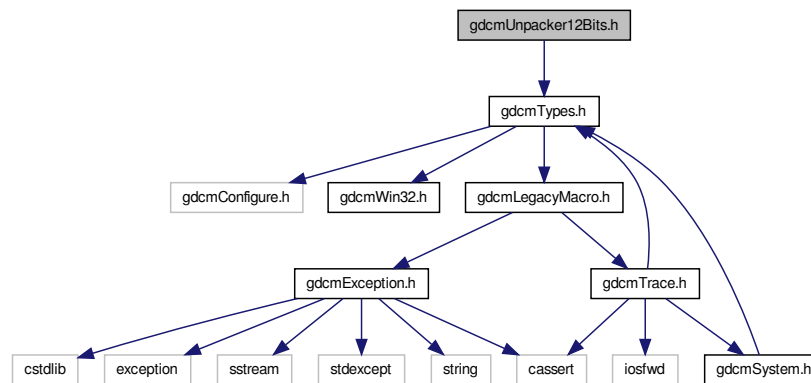
## Namespaces

- [gdcm](#)

## 28.277 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

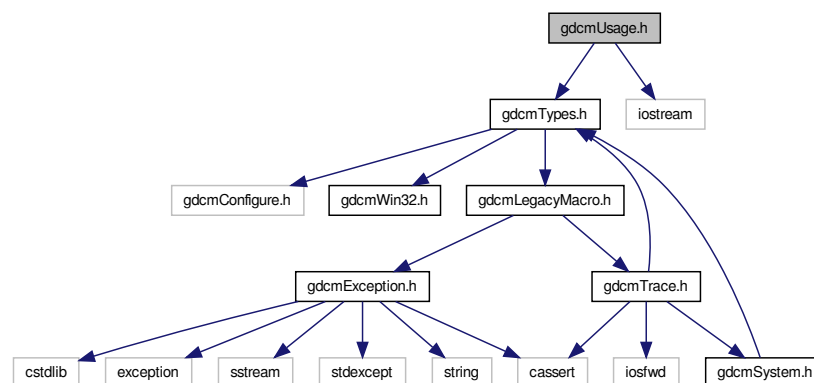
- [gdcm](#)

## 28.278 gdcmUsage.h File Reference

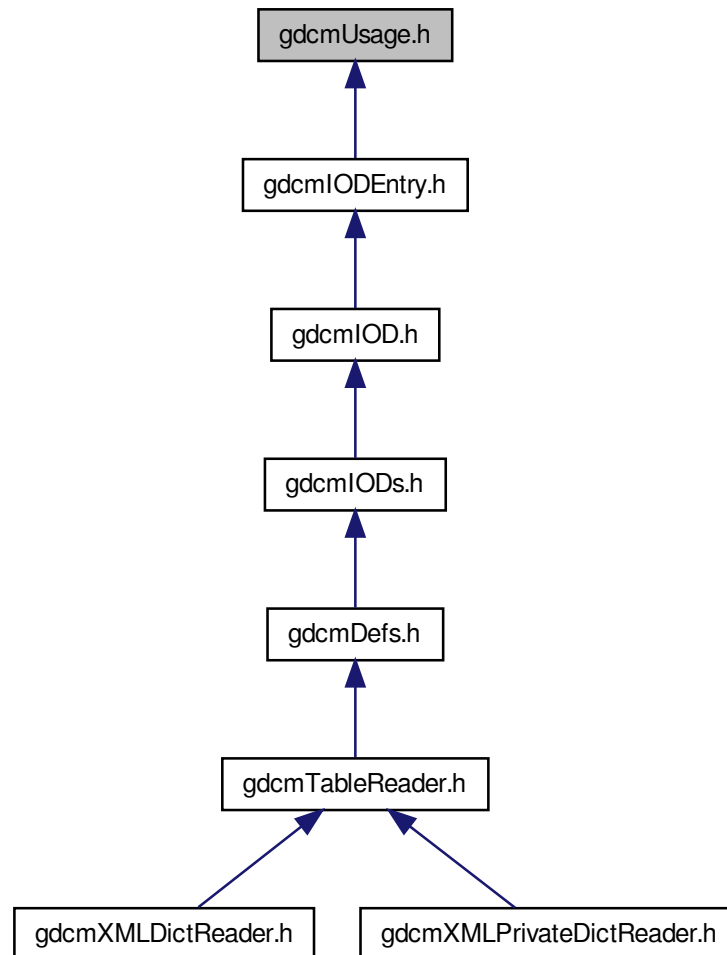
```
#include "gdcmTypes.h"
```

```
#include <iostream>
```

Include dependency graph for gdcmUsage.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcml::Usage](#)  
*Usage.*

## Namespaces

- [gdcml](#)

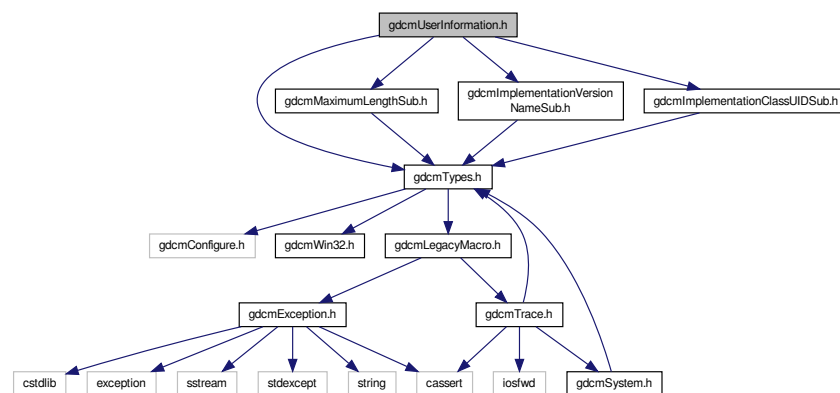
## Functions

- `std::ostream & gdcm::operator<< (std::ostream &_os, const Usage &val)`

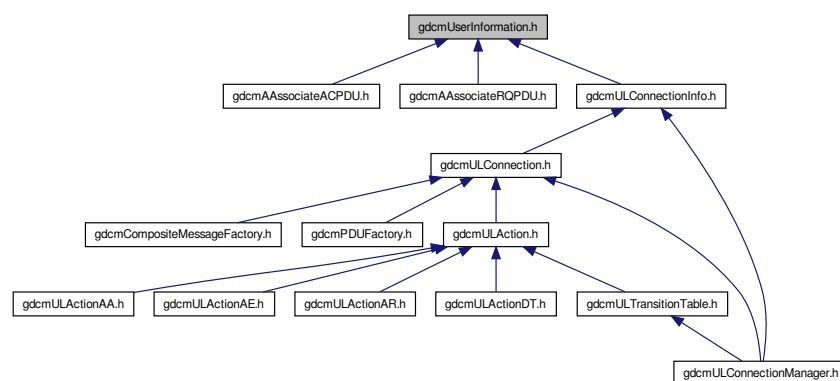
## 28.279 gdcmUserInformation.h File Reference

```
#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::UserInformation`  
*UserInformation Table 9-16 USER INFORMATION ITEM FIELDS.*

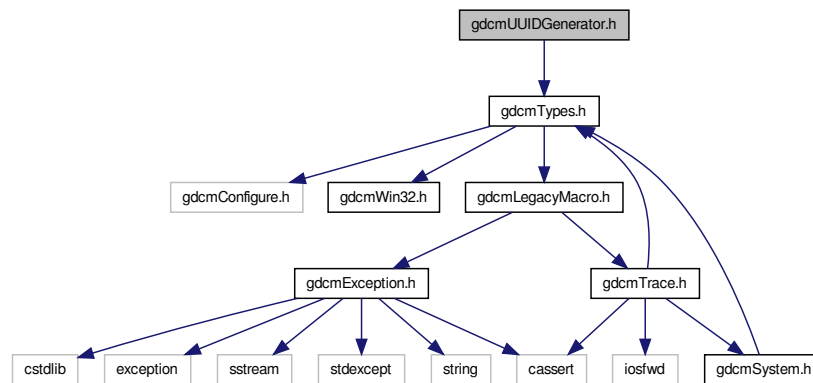
## Namespaces

- [gdcm](#)
- [gdcm::network](#)

## 28.280 gdcmUUIDGenerator.h File Reference

```
#include "gdcmTypes.h"
```

Include dependency graph for gdcmUUIDGenerator.h:



## Classes

- class [gdcm::UUIDGenerator](#)

*Class for generating unique UUID generate DCE 1.1 uid.*

## Namespaces

- [gdcm](#)

## 28.281 gdcmValidate.h File Reference

```
#include "gdcmFile.h"
```

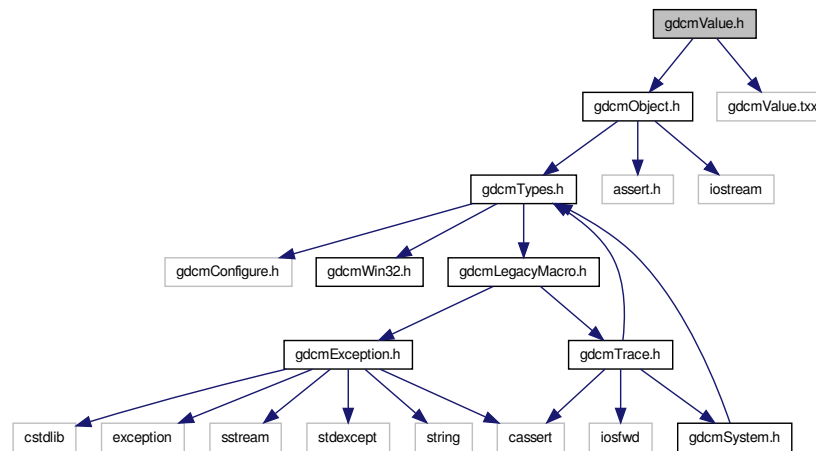
- class `gdcm::Validate`

## Namespaces

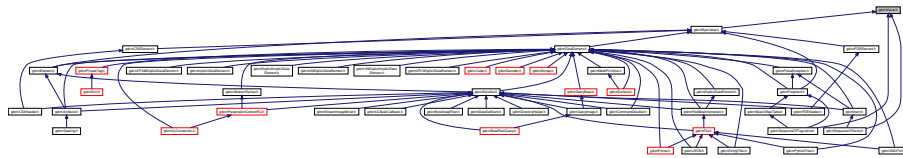
- **gdcm**

```
#include "gdcmObject.h"
#include "gdcmValue.txx"
```

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

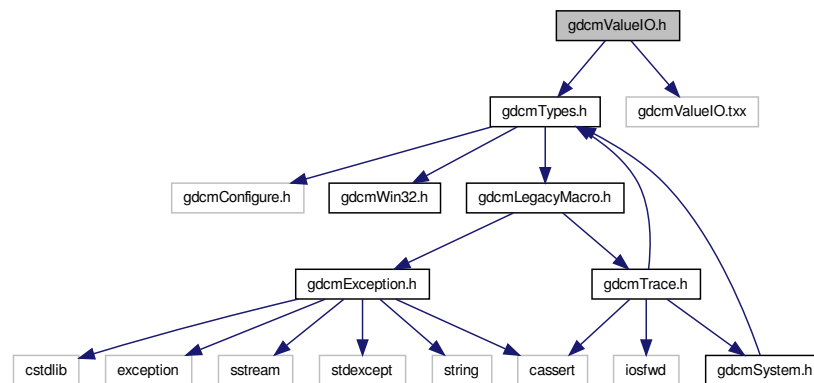
- `gdcm`

## 28.283 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

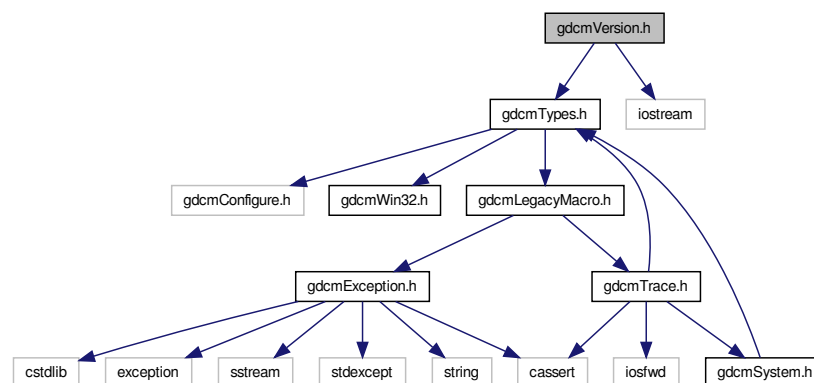
- [gdcm](#)

## 28.284 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

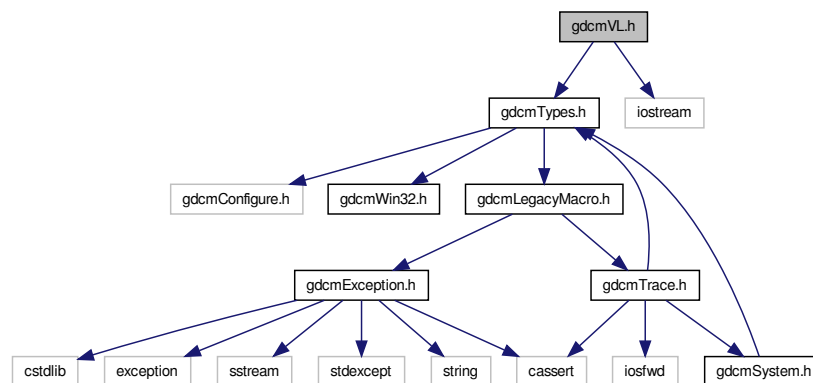
- `std::ostream & gdcm::operator<< (std::ostream &os, const Version &v)`

## 28.285 gdcmviewer.man File Reference

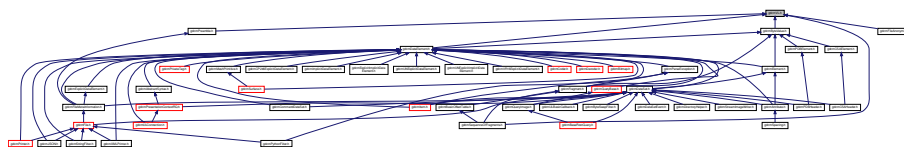
## 28.286 gdcmVL.h File Reference

```
#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

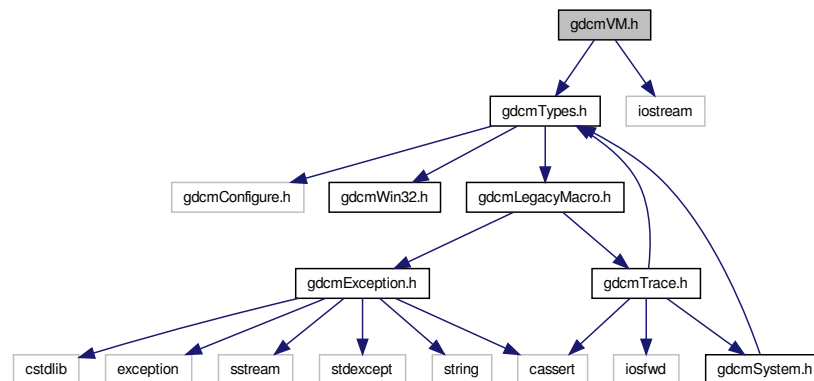
- std::ostream & [gdcm::operator<<](#) (std::ostream &os, const VL &val)

## 28.287 gdcmVM.h File Reference

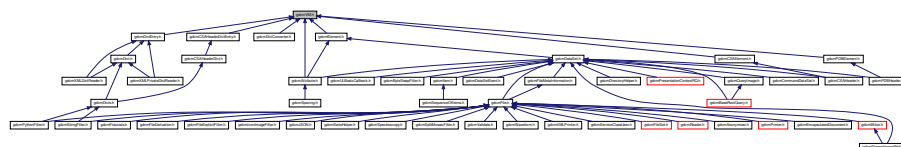
```
#include "gdcmTypes.h"
```

```
#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)`

### 28.287.1 Macro Definition Documentation

#### 28.287.1.1 `#define TYPETOLENGTH( type, length )`

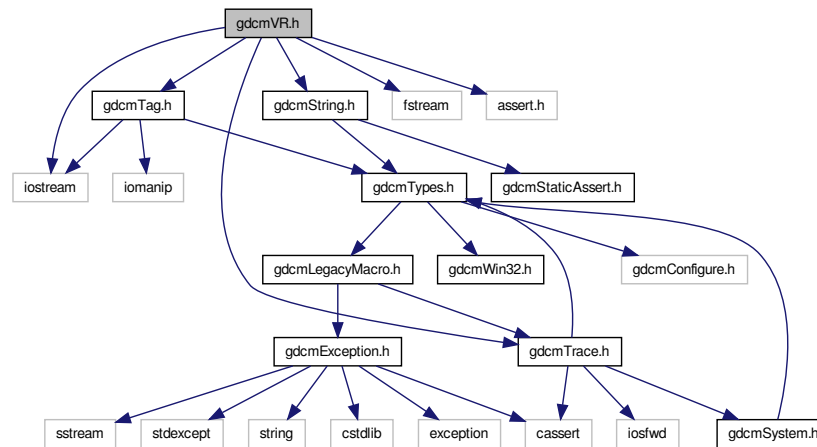
#### Value:

```
template<> struct VMToLength<VM::type> \
{ enum { Length = length }; };
```

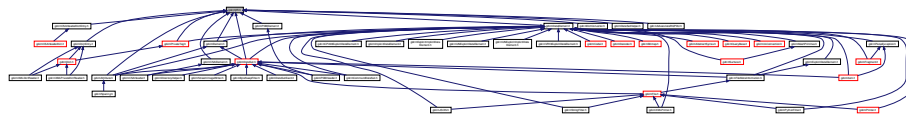
### 28.288 gdcmVR.h File Reference

```
#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 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.*

- struct [gdcm::VRToEncoding< T >](#)
- struct [gdcm::VRToType< T >](#)

## Namespaces

- [gdcm](#)

## Macros

- `#define` [TYPETOENCODING](#)(type, rep, rtype)
- `#define` [VRTypeTemplateCase](#)(type)

## Typedefs

- `typedef` `String<'\\', 16 >` [gdcm::AEComp](#)

- `typedef String<'\', 64 > gdcm::ASComp`
- `typedef String<'\', 16 > gdcm::CSComp`
- `typedef String<'\', 64 > gdcm::DAComp`
- `typedef String<'\', 64 > gdcm::DTComp`
- `typedef String<'\', 64 > gdcm::LOComp`
- `typedef String<'\', 64 > gdcm::LTComp`
- `typedef String<'\', 64 > gdcm::PNComp`
- `typedef String<'\', 64 > gdcm::SHComp`
- `typedef String<'\', 64 > gdcm::STComp`
- `typedef String<'\', 16 > gdcm::TMComp`
- `typedef String<'\', 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`

## Variables

- `gdcm::VRBINARY`

## 28.288.1 Macro Definition Documentation

### 28.288.1.1 `#define TYPETOENCODING( type, rep, rtype )`

#### Value:

```
template<> struct VRToEncoding<VR::type> \
{ enum { Mode = VR::rep }; }; \
template<> struct VRToType<VR::type> \
{ typedef rtype Type; };
```

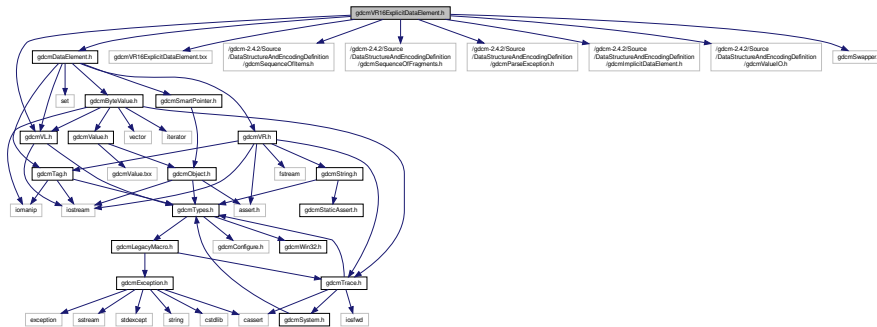
### 28.288.1.2 `#define VRTypeTemplateCase( type )`

#### Value:

```
case VR::type: \
return sizeof ( VRToType<VR::type>::Type );
```

Referenced by `gdcm::VR::GetSize()`.

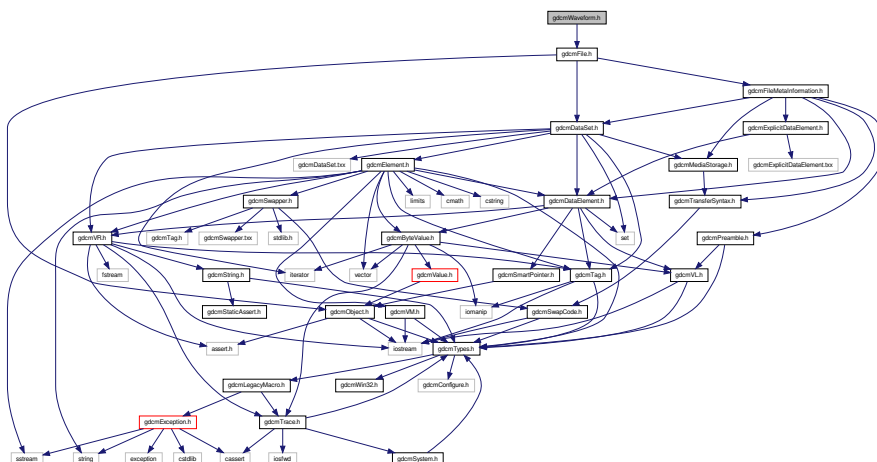
```
#include "gdcmDataElement.h"
#include "gdcmVR16ExplicitDataElement.hxx"
Include dependency graph for gdcmVR16ExplicitDataElement.h:
```



- class `gdcm::VR16ExplicitDataElement`  
*Class to read/write a `DataElement` as Explicit Data `Element`.*

- `gdcm`

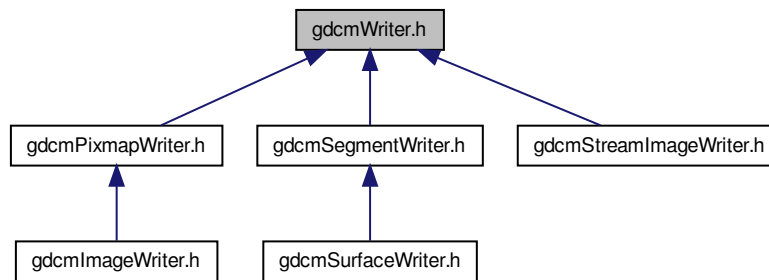
```
#include "gdcmFile.h"
Include dependency graph for gdcmWaveform.h:
```







This graph shows which files directly or indirectly include this file:



## Classes

- class [gdcm::Writer](#)

*Writer* ala DOM (Document *Object* Model) This class is a non-validating writer, it will only performs well- formedness check only.

## Namespaces

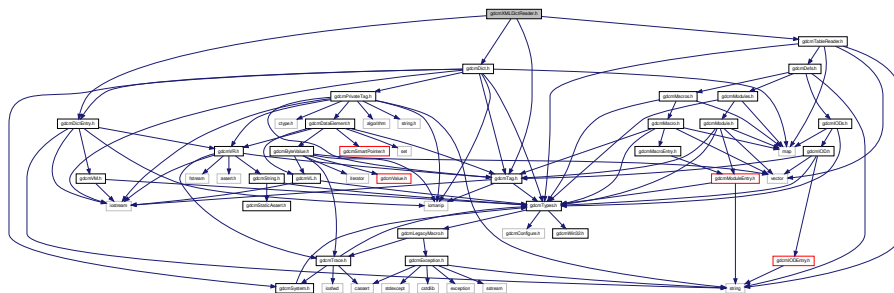
- [gdcm](#)

## 28.293 gdcmxml.man File Reference

## 28.294 gdcmXMLDictReader.h File Reference

```
#include "gdcmTableReader.h"
#include "gdcmDict.h"
#include "gdcmDictEntry.h"
#include "gdcmTag.h"
```

Include dependency graph for gdcmXMLDictReader.h:



## Classes

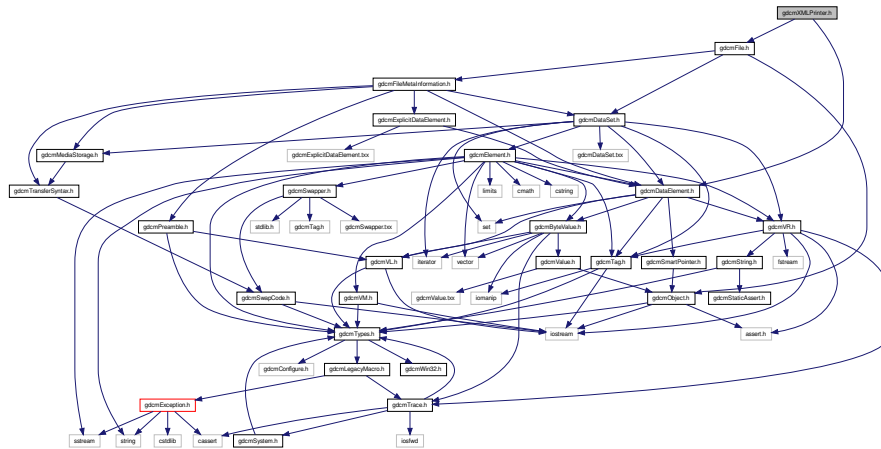
- class [gdcm::XMLDictReader](#)  
Class for representing a *XMLDictReader*.

## Namespaces

- [gdcm](#)

## 28.295 gdcmXMLPrinter.h File Reference

```
#include "gdcmFile.h"
#include "gdcmDataElement.h"
Include dependency graph for gdcmXMLPrinter.h:
```



## Classes

- class [gdcm::XMLPrinter](#)

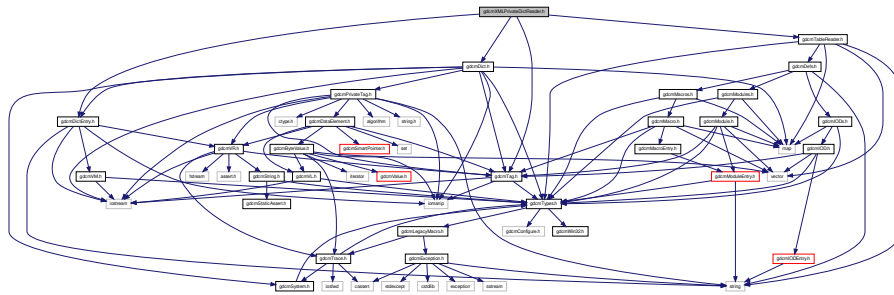
## Namespaces

- [gdcm](#)

## 28.296 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`

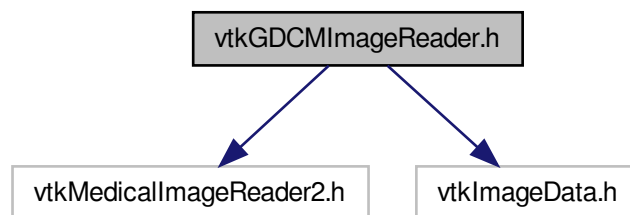
## 28.297 README.txt File Reference

## 28.298 TestsList.txt File Reference

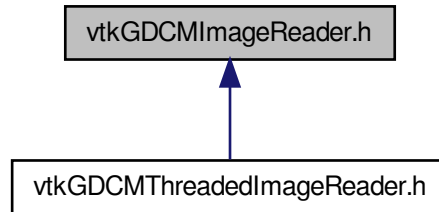
## 28.299 vtkGDCMImageReader.h File Reference

```
#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

- [gdc](#)

## Macros

- `#define VTK_CMYK` 8
- `#define VTK_INVERSE_LUMINANCE` 5
- `#define VTK_LOOKUP_TABLE` 6
- `#define VTK_YBR` 7

### 28.299.1 Macro Definition Documentation

28.299.1.1 `#define VTK_CMYK` 8

28.299.1.2 `#define VTK_INVERSE_LUMINANCE` 5

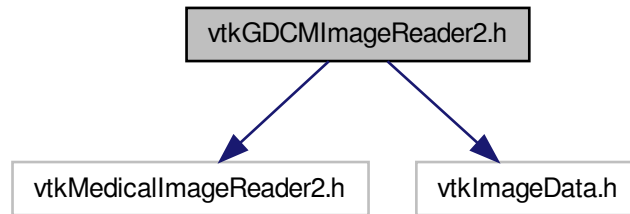
28.299.1.3 `#define VTK_LOOKUP_TABLE` 6

28.299.1.4 `#define VTK_YBR` 7

## 28.300 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`

### 28.300.1 Macro Definition Documentation

28.300.1.1 `#define VTK_CMYK 8`

28.300.1.2 `#define VTK_INVERSE_LUMINANCE 5`

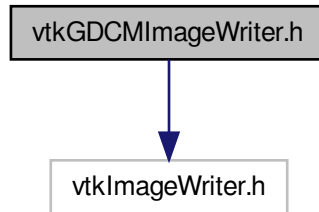
28.300.1.3 `#define VTK_LOOKUP_TABLE 6`

28.300.1.4 `#define VTK_YBR 7`

## 28.301 vtkGDCMImageWriter.h File Reference

```
#include "vtkImageWriter.h"
```

Include dependency graph for vtkGDCMImageWriter.h:



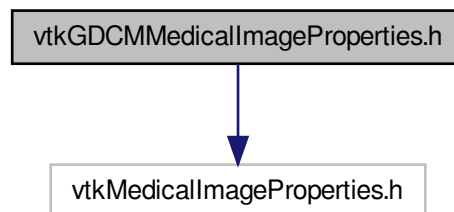
## Classes

- class [vtkGDCMImageWriter](#)

## 28.302 vtkGDCMMedicalImageProperties.h File Reference

```
#include "vtkMedicalImageProperties.h"
```

Include dependency graph for vtkGDCMMedicalImageProperties.h:



## Classes

- class [vtkGDCMMedicalImageProperties](#)

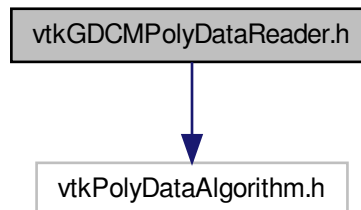
## Namespaces

- [gdc](#)

## 28.303 vtkGDCMPolyDataReader.h File Reference

```
#include "vtkPolyDataAlgorithm.h"
```

Include dependency graph for vtkGDCMPolyDataReader.h:



### Classes

- class [vtkGDCMPolyDataReader](#)

### Namespaces

- [gdc](#)

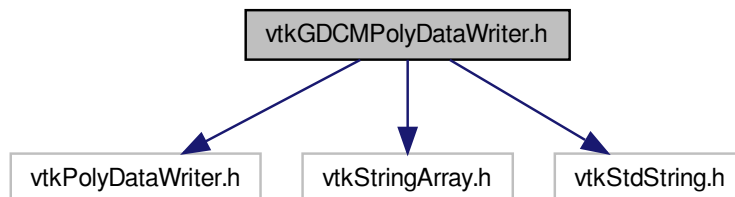
## 28.304 vtkGDCMPolyDataWriter.h File Reference

```
#include "vtkPolyDataWriter.h"
```

```
#include "vtkStringArray.h"
```

```
#include "vtkStdString.h"
```

Include dependency graph for vtkGDCMPolyDataWriter.h:



## Classes

- class [vtkGDCMPolyDataWriter](#)

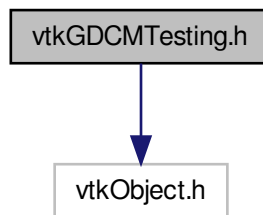
## Namespaces

- [gdc](#)

## 28.305 vtkGDCMTesting.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkGDCMTesting.h:



## Classes

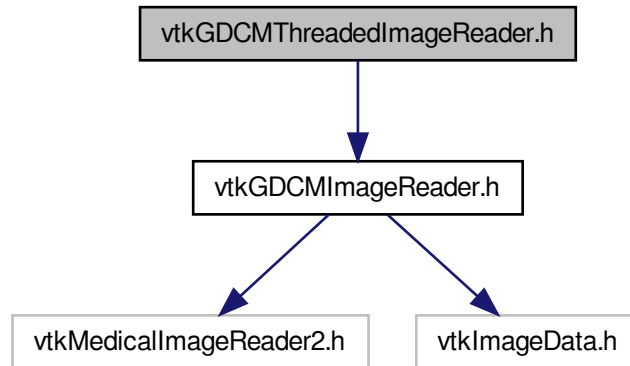
- class [vtkGDCMTesting](#)

## 28.306 vtkGDCMThreadedImageReader.h File Reference

```
#include "vtkGDCMImageReader.h"
```



Include dependency graph for vtkGDCMThreadedImageReader.h:



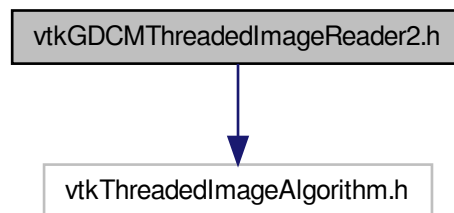
## Classes

- class [vtkGDCMThreadedImageReader](#)

## 28.307 vtkGDCMThreadedImageReader2.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for `vtkGDCMThreadedImageReader2.h`:



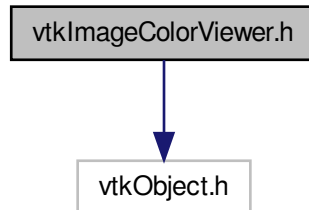
## Classes

- class [vtkGDCMThreadedImageReader2](#)

## 28.308 vtkImageColorViewer.h File Reference

```
#include "vtkObject.h"
```

Include dependency graph for vtkImageColorViewer.h:



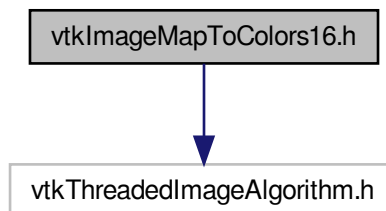
### Classes

- class [vtkImageColorViewer](#)

## 28.309 vtkImageMapToColors16.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageMapToColors16.h:



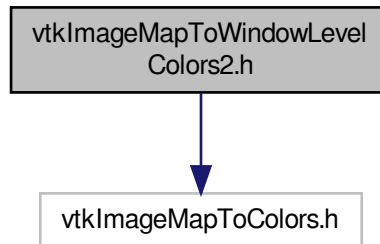
### Classes

- class [vtkImageMapToColors16](#)

## 28.310 vtkImageMapToWindowLevelColors2.h File Reference

```
#include "vtkImageMapToColors.h"
```

Include dependency graph for vtkImageMapToWindowLevelColors2.h:



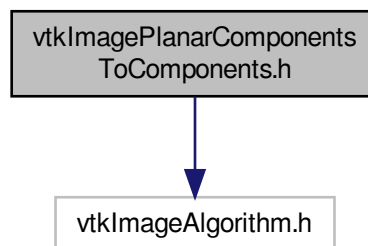
### Classes

- class [vtkImageMapToWindowLevelColors2](#)

## 28.311 vtkImagePlanarComponentsToComponents.h File Reference

```
#include "vtkImageAlgorithm.h"
```

Include dependency graph for vtkImagePlanarComponentsToComponents.h:



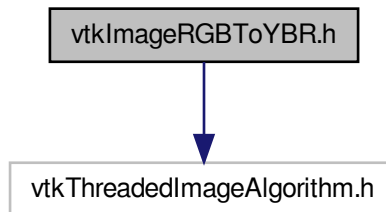
### Classes

- class [vtkImagePlanarComponentsToComponents](#)

### 28.312 vtkImageRGBToYBR.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageRGBToYBR.h:



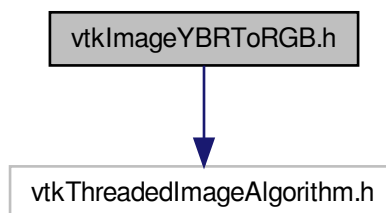
#### Classes

- class [vtkImageRGBToYBR](#)

### 28.313 vtkImageYBRTToRGB.h File Reference

```
#include "vtkThreadedImageAlgorithm.h"
```

Include dependency graph for vtkImageYBRTToRGB.h:

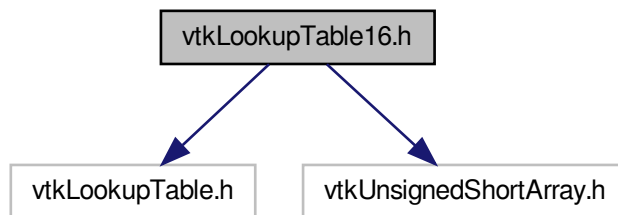


#### Classes

- class [vtkImageYBRTToRGB](#)

## 28.314 vtkLookupTable16.h File Reference

```
#include "vtkLookupTable.h"  
#include "vtkUnsignedShortArray.h"  
Include dependency graph for vtkLookupTable16.h:
```

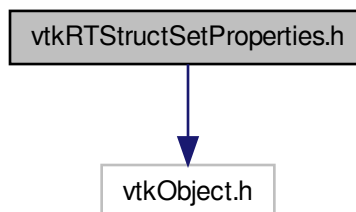


### Classes

- class [vtkLookupTable16](#)

## 28.315 vtkRTStructSetProperties.h File Reference

```
#include "vtkObject.h"  
Include dependency graph for vtkRTStructSetProperties.h:
```



### Classes

- class [vtkRTStructSetProperties](#)



## Chapter 29

# Example Documentation

### 29.1 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);
    }
}

```

## 29.2 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;
    }
}

```

## 29.3 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;
    }
}

```

## 29.4 CastConvertPhilips.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 vtkgdcm
31 import vtk
32 import sys
33 import gdcm
34
35 def ProcessOneFilePublic(filename, outfilename, tmpfile):
36     gdcm.ImageHelper.SetForceRescaleInterceptSlope(True)
37     vtkreader = vtkgdcm.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 = vtkgdcm.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 = gdcm.ImageReader()
60     tmpreader.SetFileName( tmpfile )
61     if not tmpreader.Read():
62         sys.exit(1)
63
64     reader = gdcm.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         gdcm.Tag(0x0028,0x1052),
73         gdcm.Tag(0x0028,0x1053),
74         gdcm.Tag(0x0028,0x1053),
75     ]
76     for tag in tags:
77         ds.Remove( tag )
78

```

```

79 writer = gdcmm.ImageWriter()
80 writer.SetFileName( outfile )
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, outfile, 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     el1 = ds.GetDataElement( tag1 )
112     el2 = ds.GetDataElement( tag2 )
113
114
115     #pf = gdcmm.PythonFilter()
116     #pf.SetFile( reader.GetFile() )
117     #print el1.GetTag()
118
119     print el1.GetByteValue()
120     v1 = eval(el1.GetByteValue().GetBuffer())
121     print el2.GetByteValue()
122     v2 = eval(el2.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( outfile )
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     gdcmm.Trace.DebugOff()
169     gdcmm.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 = gdcmm.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"

```

## 29.5 ChangeSequenceUltrasound.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 "gdcmmSmartPointer.h"
#include "gdcmmDataSetHelper.h"

/*
./ChangeSequenceUltrasound gdcmmData/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 < 0 )
    {
        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() )
{
    gdc::Item &item1 = sqsis->GetItem(1);
    gdc::DataSet &nestedds = item1.GetNestedDataSet();
    gdc::Tag tprcs(0x0040,0xa170); // PurposeOfReferenceCodeSequence
    if( nestedds.FindDataElement( tprcs ) )
    {
        const gdc::DataElement &prcs = nestedds.GetDataElement( tprcs );
        gdc::SmartPointer<gdc::SequenceOfItems> sqprcs = prcs.
        GetValueAssQ();
        if ( sqprcs && sqprcs->GetNumberOfItems() )
        {
            gdc::Item &item2 = sqprcs->GetItem(1);
            gdc::DataSet &nestedds2 = item2.GetNestedDataSet();
            // (0008,0104) LO [Uncompressed predecessor] # 24, 1 CodeMeaning
            gdc::Tag tcm(0x0008,0x0104);
            if( nestedds2.FindDataElement( tcm ) )
            {
                gdc::DataElement cm = nestedds2.GetDataElement( tcm );
                std::string mystr = "GDCM was here";
                cm.SetByteValue( mystr.c_str(), (uint32_t)mystr.size() );
                nestedds2.Replace( cm );
            }
        }
    }
}

gdc::Writer writer;
writer.SetFile( file );
writer.SetFileName( outfilename );
if ( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 29.6 CheckBigEndianBug.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.

=====*/
/*
 * WARNING: This is a dev tool, do not use !
 *
 * Usage: after a gdcconv, 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 "gdcmImageReader.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;
}

```

## 29.7 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( gdcm::Tag(0x12,0x31), "My Clinical Trial Site Name" );
    // <entry group="0012" element="0040" vr="LO" vm="1" name="Clinical Trial Subject ID"/>
    ano.Replace( gdcm::Tag(0x12,0x40), "My Clinical Trial Subject ID" );
    // <entry group="0012" element="0042" vr="LO" vm="1" name="Clinical Trial Subject Reading ID"/>
    ano.Replace( gdcm::Tag(0x12,0x42), "My Clinical Trial Subject Reading ID" );

    gdcm::Writer writer;
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
}

```

```

    return 0;
}

```

## 29.8 ClinicalTrialIdentificationWorkflow.cs

This is a C# example on how to use [gdcm::Anonymizer](#)

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * Typical usage on UNIX:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ClinicalTrialIdentificationWorkflow.exe input_dir output_dir
 */
using System;
using gdcm;

public class MyWatcher : SimpleSubjectWatcher
{
    public MyWatcher(Subject s):base(s,"Override String"){
    protected override void StartFilter() {
        System.Console.WriteLine( "This is my start" );
    }
    protected override void EndFilter(){
        System.Console.WriteLine( "This is my end" );
    }
    protected override void ShowProgress(Subject caller, Event evt){
        ProgressEvent pe = ProgressEvent.Cast(evt);
        System.Console.WriteLine( "This is my progress: " + pe.GetProgress() );
    }
    protected override void ShowIteration(){
        System.Console.WriteLine( "This is my iteration" );
    }
    protected override void ShowAnonymization(Subject caller, Event evt){
/*
 * A couple of explanation are necessary here to understand how SWIG work
 * http://www.swig.org/Doc1.3/Java.html#adding_downcasts
 *
 * System.Console.WriteLine( "This is my Anonymization. Type: " + evt.GetEventName() );
 * System.Type type = evt.GetType();
 * System.Console.WriteLine( "This is my Anonymization. System.Type: " + type.ToString() );
 * System.Console.WriteLine( "This is my Anonymization. CheckEvent: " + ae.CheckEvent( evt ) );
 * System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + ae.GetTag().toString() );
 */
        AnonymizeEvent ae = AnonymizeEvent.Cast(evt);
        if( ae != null )
        {
            Tag t = ae.GetTag();
            System.Console.WriteLine( "This is my Anonymization. Processing Tag #" + t.toString() );
        }
        else
        {
            System.Console.WriteLine( "This is my Anonymization. Unhandled Event type: " + evt.GetEventName() );
        }
    }
    protected override void ShowAbort(){
        System.Console.WriteLine( "This is my abort" );
    }
}

public class ClinicalTrialIdentificationWorkflow
{
    public static bool ProcessOneFile( gdcm.Anonymizer ano , string filename, string outfilename )
    {

```

```

Reader reader = new Reader();
reader.SetFileName( filename );
bool ret = reader.Read();
if( !ret )
{
    return false;
}
// Pass in the file:
ano.SetFile( reader.GetFile() );

// First step, let's protect all Patient information as per
// PS 3.15 / E.1 / Basic Application Level Confidentiality Profile
if( !ano.BasicApplicationLevelConfidentialityProfile() )
{
    return false;
}

// Now let's pass in all Clinical Trial fields
// PS 3.3 - 2008 / C.7.1.3 Clinical Trial Subject Module
/*
Clinical Trial Sponsor Name (0012,0010) 1 The name of the clinical trial sponsor. See C.7.1.3.1.1.
Clinical Trial Protocol ID (0012,0020) 1 Identifier for the noted protocol. See C.7.1.3.1.2.
Clinical Trial Protocol Name (0012,0021) 2 The name of the clinical trial protocol. See C.7.1.3.1.3.
Clinical Trial Site ID (0012,0030) 2 The identifier of the site responsible for submitting clinical
    trial data. See C.7.1.3.1.4.
Clinical Trial Site Name (0012,0031) 2 Name of the site responsible for submitting clinical trial data.
    See C.7.1.3.1.5
Clinical Trial Subject ID (0012,0040) 1C The assigned identifier for the clinical trial subject. See
    C.7.1.3.1.6. Shall be present if Clinical Trial Subject Reading ID (0012,0042) is absent. May be present
    otherwise.
Clinical Trial Subject Reading ID (0012,0042) 1C Identifies the subject for blinded evaluations. Shall
    be present if Clinical Trial Subject ID (0012,0040) is absent. May be present otherwise. See C.7.1.3.1.7.
*/
ano.Replace( new gdcm.Tag(0x0012,0x0010), "MySponsorName");
ano.Replace( new gdcm.Tag(0x0012,0x0020), "MyProtocolID");
ano.Replace( new gdcm.Tag(0x0012,0x0021), "MyProtocolName");
ano.Replace( new gdcm.Tag(0x0012,0x0030), "MySiteId");
ano.Replace( new gdcm.Tag(0x0012,0x0031), "MySiteName");
ano.Replace( new gdcm.Tag(0x0012,0x0040), "MySponsorId");
ano.Replace( new gdcm.Tag(0x0012,0x0050), "MyTPId");
ano.Replace( new gdcm.Tag(0x0012,0x0051), "MyTPDescription");

// The following two are not required as they are guaranteed to be filled in by the
// Basic Application Level Confidentiality Profile. Only override if you understand what
// you are doing
//ano.Replace( new gdcm.Tag(0x0012,0x0062), "YES");
//ano.Replace( new gdcm.Tag(0x0012,0x0063), "My Super Duper Anonymization Overload");

// We might be generating a subdirectory. Let's make sure the subdir exist:
gdcm.FileMetaInformation fmi = ano.GetFile().GetHeader();
// The following three lines make sure to regenerate any value:
fmi.Remove( new gdcm.Tag(0x0002,0x0012) );
fmi.Remove( new gdcm.Tag(0x0002,0x0013) );
fmi.Remove( new gdcm.Tag(0x0002,0x0016) );

Writer writer = new Writer();
writer.SetFileName( outfilename );
writer.SetFile( ano.GetFile() );
ret = writer.Write();
if( !ret )
{
    return false;
}

return true;
}

public static int Main(string[] args)
{
    gdcm.FileMetaInformation.SetSourceApplicationEntityTitle( "My ClinicalTrial App" );

    // http://www.oid-info.com/get/1.3.6.1.4.17434
    string THERALYS_ORG_ROOT = "1.3.6.1.4.17434";
    gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT );
}

```

```

System.Console.WriteLine( "Root dir is now: " + gdcmm.UIDGenerator.GetRoot() );

gdcmm.Global global = gdcmm.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( !gdcmm.PosixEmulation.FileIsDirectory(dir1) )
{
    System.Console.WriteLine( "Input directory: " + dir1 + " does not exist. Sorry" );
    return 1;
}
if( !gdcmm.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 = gdcmm.Filename.Join(gdcmm.Testing.GetSourceDirectory(), "
/Testing/Source/Data/certificate.pem" );
gdcmm.CryptoFactory fact = gdcmm.CryptoFactory.GetFactoryInstance();
gdcmm.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);

// Explicitely specify the Cryptographic Message Syntax to use:
ano.SetCryptographicMessageSyntax( cms );

// Process all filenames:
FilenamesType filenames = d.GetFilesNames();
for( uint i = 0; i < nfiles; ++i )
{
    string filename = filenames[ (int)i ];
    string outfilename = filename.Replace( dir1, dir2 );
    System.Console.WriteLine( "Filename: " + filename );
    System.Console.WriteLine( "Out Filename: " + outfilename );
    if( !ProcessOneFile( ano , filename, outfilename ) )
    {
        System.Console.WriteLine( "Could not process filename: " + filename );
        return 1;
    }
}

return 0;
}
}

```

## 29.9 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(
        gdcm::TransferSyntax::JPEG2000Lossless );
    change.SetTransferSyntax(
        gdcm::TransferSyntax::JPEGLosslessProcess14_1 );
    //change.SetTransferSyntax( gdcm::TransferSyntax::JPEGBaselineProcess1 );
    //change.SetTransferSyntax( image.GetTransferSyntax() );
    change.SetInput( image );
    bool b = change.Change();
    if( !b )
    {
        std::cerr << "Could not change the Transfer Syntax" << std::endl;
        return 1;
    }

    //std::ofstream out( outfile, std::ios::binary );
    //image.GetBuffer2(out);
    //out.close();
    gdcm::ImageWriter writer;
    writer.SetImage( change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfile );
    if( !writer.Write() )
    {

```

```

    return 1;
}

return 0;
}

```

## 29.10 CompressLossyJPEG.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcm/debug-gcc/bin
 * $ mono bin/CompressLossyJPEG.exe input.dcm output.dcm
 */

using System;
using gdcm;

public class CompressLossyJPEG
{
    public static int Main(string[] args)
    {
        if( args.Length < 2 )
        {
            System.Console.WriteLine( " input.dcm output.dcm" );
            return 1;
        }
        string filename = args[0];
        string outfilename = args[1];

        ImageReader reader = new ImageReader();
        reader.SetFileName( filename );
        if( !reader.Read() )
        {
            System.Console.WriteLine( "Could not read: " + filename );
            return 1;
        }

        // The output of gdcm::Reader is a gdcm::File
        File file = reader.GetFile();

        // the dataset is the the set of element we are interested in:
        DataSet ds = file.GetDataSet();

        Image image = reader.GetImage();
        //image.Print( cout );

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        TransferSyntax targetts = new TransferSyntax( TransferSyntax.TSType.JPEGBaselineProcess1 );
        change.SetTransferSyntax( targetts );

        // Setup our JPEGCodec, warning it should be compatible with JPEGBaselineProcess1
        JPEGCodec jpegcodec = new JPEGCodec();
        if( !jpegcodec.CanCode( targetts ) )
        {
            System.Console.WriteLine( "Something went really wrong, JPEGCodec cannot handle JPEGBaselineProcess1" );
            return 1;
        }
        jpegcodec.SetLossless( false );
        jpegcodec.SetQuality( 50 ); // poor quality !
        change.SetUserCodec( jpegcodec ); // specify the codec to use to the ImageChangeTransferSyntax

        change.SetInput( image );
        bool b = change.Change();
    }
}

```



```

    if( !b )
    {
        System.Console.WriteLine( "Could not change the Transfer Syntax" );
        return 1;
    }

    ImageWriter writer = new ImageWriter();
    writer.SetImage( (gdcm.Image)change.GetOutput() );
    writer.SetFile( reader.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
    {
        System.Console.WriteLine( "Could not write: " + outfilename );
        return 1;
    }

    return 0;
}
}

```

## 29.11 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:
// gdcmData/012345.002.050.dcm

int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/012345.002.050.dcm";
    std::cout << file << std::endl;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    #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;
}

```

## 29.12 ConvertMPL.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 gdcm
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     ## use float for accurate scaling
63     result = numpy.frombuffer(gdcm_array, dtype=dtype).astype(float)
64     ## optional gamma scaling

```

```

65     #maxV = float(result[result.argmax()])
66     #result = result + .5*(maxV-result)
67     #result = numpy.log(result+50) ## apprxx background level
68     result.shape = d
69     return result
70
71 if __name__ == "__main__":
72     import sys
73     r = gdcm.ImageReader()
74     filename = sys.argv[1]
75     r.SetFileName( filename )
76     if not r.Read(): sys.exit(1)
77     numpy_array = gdcm_to_numpy( r.GetImage() )
78
79     subplot(111)# one plot, on left
80     title(filename)
81     ## many colormaps are available
82     imshow(numpy_array, interpolation='bilinear', cmap=cm.jet)
83     ## set the plot sizes and placement
84     subplots_adjust(bottom=0.1, right=0.8, top=0.9)
85     cax = axes([0.85, 0.1, 0.075, 0.8])
86     colorbar(cax=cax)
87     title('values')
88     get_current_fig_manager().window.title('plot')
89     show()

```

## 29.13 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::ostringstream 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;
}

```

## 29.14 ConvertNumpy.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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_tymap():
29     """Returns the GDCM Pixel Format to numpy array type mapping."""
30     _gdcm_np = {gdcm.PixelFormat.UINT8 :numpy.int8,
31                 gdcm.PixelFormat.INT8  :numpy.uint8,
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         gdcmm.PixelFormat.UINT32 :numpy.uint32,
37         gdcmm.PixelFormat.INT32  :numpy.int32,
38         #gdcmm.PixelFormat.FLOAT16:numpy.float16,
39         gdcmm.PixelFormat.FLOAT32:numpy.float32,
40         gdcmm.PixelFormat.FLOAT64:numpy.float64 }
41     return _gdcmm_np
42
43 def get_numpy_array_type(gdcmm_pixel_format):
44     """Returns a numpy array typecode given a GDCM Pixel Format."""
45     return get_gdcmm_to_numpy_typemap()[gdcmm_pixel_format]
46
47 def gdcmm_to_numpy(image):
48     """Converts a GDCM image to a numpy array.
49     """
50     pf = image.GetPixelFormat()
51
52     assert pf.GetScalarType() in get_gdcmm_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     gdcmm_array = image.GetBuffer()
61     result = numpy.frombuffer(gdcmm_array, dtype=dtype)
62     result.shape = shape
63     return result
64
65 if __name__ == "__main__":
66     import sys
67     r = gdcmm.ImageReader()
68     filename = sys.argv[1]
69     r.SetFileName( filename )
70     if not r.Read():
71         sys.exit(1)
72
73     numpy_array = gdcmm_to_numpy( r.GetImage() )
74     print numpy_array

```

## 29.15 ConvertPIL.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 gdcmm
30 import numpy
31 from PIL import Image, ImageOps
32
33
34 def get_gdcmm_to_numpy_typemap():
35     """Returns the GDCM Pixel Format to numpy array type mapping."""

```

```

36     _gdcn_np = {gdcn.PixelFormat.UINT8 :numpy.int8,
37                 gdcn.PixelFormat.INT8 :numpy.uint8,
38                 gdcn.PixelFormat.UINT16 :numpy.uint16,
39                 gdcn.PixelFormat.INT16 :numpy.int16,
40                 gdcn.PixelFormat.UINT32 :numpy.uint32,
41                 gdcn.PixelFormat.INT32 :numpy.int32,
42                 gdcn.PixelFormat.FLOAT32 :numpy.float32,
43                 gdcn.PixelFormat.FLOAT64 :numpy.float64 }
44     return _gdcn_np
45
46 def get_numpy_array_type(gdcn_pixel_format):
47     """Returns a numpy array typecode given a GDCM Pixel Format."""
48     return get_gdcn_to_numpy_typemap()[gdcn_pixel_format]
49
50 def gdcn_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_gdcn_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     gdcn_array = image.GetBuffer()
62     result = numpy.frombuffer(gdcn_array, dtype=dtype)
63     maxV = float(result[result.argmax()])
64     ## linear gamma adjust
65     #result = result + .5*(maxV-result)
66     ## log gamma
67     result = numpy.log(result+50) ## 50 is apprxx background level
68     maxV = float(result[result.argmax()])
69     result = result*(2.**8/maxV) ## histogram stretch
70     result.shape = d
71     return result
72
73 if __name__ == "__main__":
74     import sys
75     r = gdcn.ImageReader()
76     filename = sys.argv[1]
77     r.SetFileName( filename )
78     if not r.Read(): sys.exit(1)
79     numpy_array = gdcn_to_numpy( r.GetImage() )
80     ## L is 8 bit grey
81     ## http://www.pythonware.com/library/pil/handbook/concepts.htm
82     pilImage = Image.frombuffer('L',
83                                numpy_array.shape,
84                                numpy_array.astype(numpy.uint8),
85                                'raw','L',0,1)
86     ## cutoff removes background noise and spikes
87     pilImage = ImageOps.autocontrast(pilImage, cutoff=.1)
88     pilImage.save(sys.argv[1]+' .jpg')

```

## 29.16 ConvertRGBToLuminance.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcn.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageLuminance.h"

#include "gdcnTesting.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;
}
```

## 29.17 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( outfilename );
//writer->SetInput( cast->GetOutput() );
#if (VTK_MAJOR_VERSION >= 6)
writer->SetInputData( copy );
#else
writer->SetInput( copy );
#endif
writer->SetImageFormat( reader->GetImageFormat() );
writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
writer->SetDirectionCosines( reader->GetDirectionCosines() );
writer->SetShift( reader->GetShift() );
writer->SetScale( reader->GetScale() );
writer->SetFileDimensionality( reader->GetFileDimensionality() );
writer->Write();

reader->Delete();
copy->Delete();
writer->Delete();

return 0;
}

```

## 29.18 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:
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                *pubuffer++ = (unsigned char)std::min(255, (32768 + *buffer16) / 255);
                buffer16++;
            }

            imageQt = new QImage(ubuffer, dimX, dimY, QImage::Format_RGB888);
        }
        else
        {
            std::cerr << "Pixel Format is: " << gimage.GetPixelFormat() << std::endl;
            return false;
        }
    }
    else
    {
        std::cerr << "Unhandled PhotometricInterpretation: " << gimage.
            GetPhotometricInterpretation() << std::endl;
    }
}

```

```

        return false;
    }

    return true;
}

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader ir;
    ir.SetFileName( filename );
    if(!ir.Read())
    {
        //Read failed
        return 1;
    }

    std::cout<<"Getting image from ImageReader..."<<std::endl;

    const gdcm::Image &gimage = ir.GetImage();
    std::vector<char> vbuffer;
    vbuffer.resize( gimage.GetBufferLength() );
    char *buffer = &vbuffer[0];

    QImage *imageQt = NULL;
    if( !ConvertToFormat_RGB888( gimage, buffer, imageQt ) )
    {
        return 1;
    }

    QImageWriter writer;
    writer.setFormat("png");
    writer.setFileName( outfile );
    if( !writer.write( *imageQt ) )
    {
        return 1;
    }

    return 0;
}

```

## 29.19 CreateARGBImage.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * http://www.w3.org/Graphics/PNG/inline-alpha.html
 * alphatest.png: PNG image data, 380 x 287, 8-bit/color RGBA, non-interlaced
 *
 * $ convert alphatest.png alphatest.rgba
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmSystem.h"
#include "gdcmImageWriter.h"

#include <iostream>
#include <fstream>

```

```

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.rgb output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    size_t len = gdcm::System::FileSize(filename);
    std::ifstream is(filename, std::ios::binary);

    char * buf = new char[len];
    is.read(buf, len);

    gdcm::ImageWriter writer;
    gdcm::Image &image = writer.GetImage();
    image.SetNumberOfDimensions( 2 );
    unsigned int dims[3] = {};
    dims[0] = 380;
    dims[1] = 287;
    image.SetDimensions( dims );
    gdcm::PixelFormat pf = gdcm::PixelFormat::UINT8;
    pf.SetSamplesPerPixel( 4 );
    image.SetPixelFormat( pf );
    gdcm::PhotometricInterpretation pi =
        gdcm::PhotometricInterpretation::ARGB;
    image.SetPhotometricInterpretation( pi );
    image.SetTransferSyntax(
        gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::DataElement pixeldata( gdcm::Tag(0x7fe0,0x0010) );
    pixeldata.SetByteValue( buf, (uint32_t)len );
    image.SetDataElement( pixeldata );

    writer.SetFileName( outfile );
    if( !writer.Write() )
    {
        return 1;
    }
    delete[] buf;

    return 0;
}

```

## 29.20 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;
}

```

## 29.21 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" );

gdcm::Attribute<0x0028,0x7FE0> at;
at.SetValue( "http://dicom.example.com/jpipserver.cgi?target=img.jp2" );
ds.Insert( at.GetAsDataElement() );

// Need to retrieve the PixelFormat information from the given file

if (!w.Write() )
{
    std::cerr << "Could not write: " << outfile << std::endl;
    return 1;
}

return 0;
}

```

## 29.22 CreateRAWStorage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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         gdcmm.Tag(0x0018,0x1110),
70         gdcmm.Tag(0x0018,0x1111),
71         gdcmm.Tag(0x0018,0x1120),
72         gdcmm.Tag(0x0018,0x1130),
73         gdcmm.Tag(0x0018,0x1150),
74         gdcmm.Tag(0x0018,0x1151),
75         gdcmm.Tag(0x0018,0x1152),
76         gdcmm.Tag(0x0018,0x1160),
77         gdcmm.Tag(0x0018,0x1190),
78         gdcmm.Tag(0x0018,0x1210),
79         gdcmm.Tag(0x0020,0x0012),
80         gdcmm.Tag(0x0020,0x0032),
81         gdcmm.Tag(0x0020,0x0037),
82         gdcmm.Tag(0x0020,0x1041),
83         gdcmm.Tag(0x0020,0x4000),
84         gdcmm.Tag(0x0028,0x0002),
85         gdcmm.Tag(0x0028,0x0004),
86         gdcmm.Tag(0x0028,0x0010),
87         gdcmm.Tag(0x0028,0x0011),
88         gdcmm.Tag(0x0028,0x0030),
89         gdcmm.Tag(0x0028,0x0100),
90         gdcmm.Tag(0x0028,0x0101),
91         gdcmm.Tag(0x0028,0x0102),
92         gdcmm.Tag(0x0028,0x0103),
93         gdcmm.Tag(0x0028,0x1052),
94         gdcmm.Tag(0x0028,0x1053),
95         gdcmm.Tag(0x0028,0x2110),
96         gdcmm.Tag(0x0028,0x2112),
97         gdcmm.Tag(0x7fe0,0x0010),
98         gdcmm.Tag(0x0018,0x0020),
99         gdcmm.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 # uid = gen.Generate()
139 # de.SetTag( gdcM.Tag(0x0008,0x0018) )
140 # de.SetByteValue( uid, gdcM.VL(len(uid)) )
141 # ds.Insert( de )
142 #
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()

```

## 29.23 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; //
    bytewidth = 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(0x7fe1,0x10,"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;
}

```

## 29.24 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 gdcData/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::ostream 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::FilenameType 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();

```

```
}
```

## 29.25 DecompressImage.cs

This is a C# example on how to use [gdcm::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;
        }

        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;
    }
}

```

## 29.26 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.TSType.ImplicitVRLittleEndian) );
        change.SetInput( reader.GetImage() );
        if( !change.Change() )
        {
            throw new Exception("Could not change: " + file1 );
        }

        Image out = change.GetOutput();
        System.out.println( out.toString() );

        // Set the Source Application Entity Title
        FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

        ImageWriter writer = new ImageWriter();
        writer.SetFileName( file2 );
        writer.SetFile( reader.GetFile() );
    }
}

```

```

writer.SetImage( out );
ret = writer.Write();
if( !ret )
{
    throw new Exception("Could not write: " + file2 );
}
}
}

```

## 29.27 DecompressImage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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     image = gdcm.Image()
35     ir = r.GetImage()
36
37     image.SetNumberOfDimensions( ir.GetNumberOfDimensions() );
38     dims = ir.GetDimensions();
39     print ir.GetDimension(0);
40     print ir.GetDimension(1);
41     print "Dims:",dims
42
43     # Just for fun:
44     dircos = ir.GetDirectionCosines()
45     t = gdcm.Orientation.GetType(dircos)
46     l = gdcm.Orientation.GetLabel(t)
47     print "Orientation label:",l
48
49     image.SetDimension(0, ir.GetDimension(0) );
50     image.SetDimension(1, ir.GetDimension(1) );
51
52     pixeltype = ir.GetPixelFormat();
53     image.SetPixelFormat( pixeltype );
54
55     pi = ir.GetPhotometricInterpretation();
56     image.SetPhotometricInterpretation( pi );
57
58     pixeldata = gdcm.DataElement( gdcm.Tag(0x7fe0,0x0010) )
59     str1 = ir.GetBuffer()
60     #print ir.GetBufferLength()
61     pixeldata.SetByteValue( str1, gdcm.VL( len(str1) ) )
62     image.SetDataElement( pixeldata )
63
64     w = gdcm.ImageWriter()
65     w.SetFileName( file2 )

```

```

66 w.SetFile( r.GetFile() )
67 w.SetImage( image )
68 if not w.Write():
69     sys.exit(1)

```

## 29.28 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):
*
* $ gdcmrw -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
* gdcmrw 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;
}

```

## 29.29 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;
    }
}

```

## 29.30 DecompressPixmap.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This example will take in a DICOM file, and tries to decompress it (actually write it
 * as ImplicitVRLittleEndian Transfer Syntax).
 *
 * Compilation:
 * $ CLASSPATH=gdcm.jar javac ../../gdcm/Examples/Java/DecompressPixmap.java -d .
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcm.jar:. java DecompressPixmap gdcmData/012345.002.050.dcm out.dcm
 */
import gdcm.*;

public class DecompressPixmap
{
    public static void main(String[] args) throws Exception
    {
        String file1 = args[0];
        String file2 = args[1];
        PixmapReader reader = new PixmapReader();
        reader.SetFileName( file1 );
        boolean ret = reader.Read();
        if( !ret )
        {
            throw new Exception("Could not read: " + file1 );
        }

        ImageChangeTransferSyntax change = new ImageChangeTransferSyntax();
        change.SetTransferSyntax( new TransferSyntax(TransferSyntax.TSType.ImplicitVRLittleEndian) );
        PixmapToPixmapFilter filter = (PixmapToPixmapFilter)change;
        filter.SetInput( reader.GetPixmap() );
        if( !change.Change() )
        {
            throw new Exception("Could not change: " + file1 );
        }

        // The following does not work in Java/swig 2.0.7
        //Pixmap p = ((PixmapToPixmapFilter)change).GetOutput();
        Pixmap p = change.GetOutputAsPixmap(); // be explicit
        //System.out.println( p.toString() );

        // Set the Source Application Entity Title
        FileMetaInformation.SetSourceApplicationEntityTitle( "Just For Fun" );

        PixmapWriter writer = new PixmapWriter();
        writer.SetFileName( file2 );
        writer.SetFile( reader.GetFile() );
        writer.SetImage( p );
        ret = writer.Write();
        if( !ret )
        {
            throw new Exception("Could not write: " + file2 );
        }
    }
}

```

## 29.31 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;
}

```

## 29.32 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 intertinted. 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);

    gdcm::Scanner::ValueType vt3 = s.GetValues(t3);
    for(
        gdcm::Scanner::ValueType::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;
    gdcm::Scanner::ValueType vt2 = s.GetValues(t2);
    for(
        gdcm::Scanner::ValueType::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 )
{
    gdcm::Scanner::ValueType vt1 = s.GetValues( gdcm::t1 );
    for(
        gdcm::Scanner::ValueType::const_iterator it = vt1.begin()
        ; it != vt1.end(); ++it )
    {
        ProcessAStudy( s, it->c_str() );
    }
}

};

} // namespace gdcm

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;
        }
        dirl = extradataroot;
        dirl += "/gdcmlSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dirl = argv[1];
    }

    gdcml::Directory d;
    d.Load( dirl.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.GetFilesNames() );
    if( !b )
    {
        std::cerr << "Scanner failed" << std::endl;
        return 1;
    }

    gdcml::DiscriminateVolume dv;
    dv.ProcessIntoVolume( s );
    dv.Print( std::cout );

    return 0;
}

```

## 29.33 DumbAnonymizer.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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","GetTPId"),
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
55     # Other:
56     (0x0012,0x0051):("Patient Field","Type Examen"),
57     (0x0018,0x1250):("Sequence Field","Receive Coil"),
58     (0x0018,0x0088):("Sequence Field","Spacing Between Slice"),
59     (0x0018,0x0095):("Sequence Field","Pixel Bandwidth"),
60     (0x0018,0x0082):("Sequence Field","Inversion Time"),
61 }
62
63 class MyAnon:
64     def __init__(self):
65         self.studyuid = None
66         self.seriesuid = None
67         generator = gdcm.UIDGenerator()
68         if not self.studyuid:
69             self.studyuid = generator.Generate()
70         if not self.seriesuid:
71             self.seriesuid = generator.Generate()
72
73     def GetSponsorInitials(self):
74         return "dummy^foobar"
75
76     def GenerateStudyId(self):
77         return self.studyuid
78
79     def GenerateSeriesId(self):
80         return self.seriesuid
81
82     #def GenerateMSOPIId(self):
83     def GenerateMSOPIId(self):
84         generator = gdcm.UIDGenerator()
85         return generator.Generate()
86
87     def GetSiteId(self):
88         return "MySiteId"
89
90     def GetSiteName(self):
91         return "MySiteName"
92
93     def GetSponsorId(self):
94         return "MySponsorId"
95
96     def GetTPId(self):
97         return "MyTP"
98
99 if __name__ == "__main__":
100     import sys
101     gdcm.FileMetaInformation.SetSourceApplicationEntityTitle
102         ( "DumbAnonymizer" )
103     gdcm.UIDGenerator.SetRoot( THERALYS_ORG_ROOT )
104
105     r = gdcm.Reader()
106     filename = sys.argv[1]
107     r.SetFileName( filename )
108     if not r.Read(): sys.exit(1)
109
110     obj = MyAnon()
111
112     w = gdcm.Writer()
113     ano = gdcm.Anonymizer()
114     ano.SetFile( r.GetFile() )
115     ano.RemoveGroupLength()
116     for tag,rule in tag_rules.items():
117         if rule[0] == 'Value':
118             print tag,rule
119             ano.Replace( gdcm.Tag( tag[0], tag[1] ), rule[1] )
120         elif rule[0] == 'Method':
121             print tag,rule
122             # result = locals()[rule[1]]()
123             methodname = rule[1]

```



```

114         if hasattr(obj, methodname):
115             _member = getattr(obj, methodname)
116             result = _member()
117             ano.Replace( gdcml.Tag( tag[0], tag[1] ), result )
118         else:
119             print "Problem with: ", methodname
120
121     outfilename = sys.argv[2]
122     w.SetFileName( outfilename )
123     w.SetFile( ano.GetFile() )
124     if not w.Write(): sys.exit(1)

```

## 29.34 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 "gdcmlReader.h"
#include "gdcmlPrivateTag.h"
#include "gdcmlAttribute.h"
#include "gdcmlImageWriter.h"

#include <iostream>
#include <fstream>
#include <vector>

#include <string.h>
#include <assert.h>
#include <stdint.h>

struct dict
{
    uint16_t key;
    const char *name;
};

dict Array[] = {
    { 0x01, "Patient name" },
    { 0x02, "Patient ID" },
    { 0x03, "Patient sex" },
    { 0x04, "Patient age" },
    { 0x05, "Patient height" },
    { 0x06, "Patient weight" },
    { 0x07, "Exam date" },
    { 0x08, "Dose admin. time" },
    { 0x09, "Unique exam key" },
    { 0x0a, "Exam procedure" },
    { 0x0b, "Referring physician" },
    { 0x0c, "Attending physician" },
    { 0x0d, "Imaging modality" },
    { 0x0e, "Hospital ID" },
    { 0x0f, "Histogram crv file" },
    { 0x10, "Acq. start time" },
    { 0x11, "Object data type" },
    { 0x12, "Image viewid" },
    { 0x13, "Imaging device name" },
    { 0x14, "Device serial number" },
    { 0x15, "Collimator" },
    { 0x16, "Software version" },
    { 0x17, "Radiopharmaceutical #1" },
    { 0x18, "Energy window #1 center" },
    { 0x19, "Radiopharmaceutical #2" },
    { 0x1a, "Energy window #1 width" },

```

```

{ 0x1b, "Isotope imaging mode" },
{ 0x1c, "Energy window #2 center" },
{ 0x1d, "Energy window #2 width" },
{ 0x1e, "Energy window #3 center" },
{ 0x1f, "Energy window #3 width" },
{ 0x20, "Energy window #4 center" },
{ 0x21, "Energy window #4 width" },
{ 0x22, "??Energy window #5 center" },
{ 0x23, "??Energy window #5 width" },
{ 0x24, "Patient orientation" },
{ 0x25, "Spatial resolution" },
{ 0x26, "Slice thickness" },
{ 0x27, "Image X dimension" },
{ 0x28, "Image Y dimension" },
{ 0x29, "Image Z dimension" },
{ 0x2a, "Image pixel width" },
{ 0x2b, "Uniformity corr. file" },
{ 0x2c, "Acquisition zoom factor" },
{ 0x2d, "Total counts in set" },
{ 0x2e, "Time / frame" },
{ 0x2f, "Total acq. time" },
{ 0x30, "Maximum pixel value" },
{ 0x31, "Minimum pixel value" },
{ 0x32, "R-R interval time" },
{ 0x33, "Percent of cycle imaged" },
{ 0x34, "# of cycles accepted" },
{ 0x35, "# of cycles rejected" },
{ 0x36, "Approximate ED frame" },
{ 0x37, "Approximate ES frame" },
{ 0x38, "Approximate EF" },
{ 0x39, "Starting angle" },
{ 0x3a, "Degrees of rotation" },
{ 0x3b, "Direction of rotation" },
{ 0x3c, "Cont. or step/shoot" },
{ 0x3d, "Lim recon start frame" },
{ 0x3e, "Upper window grey shade" },
{ 0x3f, "Lower lvl grey shade" },
{ 0x40, "Associated color map" },
{ 0x41, "Custom color map file" },
{ 0x42, "Manipulated image" },
{ 0x43, "Axis of rotation corr." },
{ 0x44, "Reorientation azimuth" },
{ 0x45, "Reorientation elevation" },
{ 0x46, "Filter type" },
{ 0x47, "Filter order" },
{ 0x48, "Filter cutoff frequency" },
{ 0x49, "Reconstruction type" },
{ 0x4a, "Attenuation coefficient" },
{ 0x4b, "Associated parent file" },
{ 0x4c, "Unique patient key" },
{ 0x52, "Normalization crv file" },
{ 0x53, "Unique object key" },
{ 0x54, "This phase of VFR is" },
{ 0x55, "True color value" },
{ 0x56, "# of sets of x,y,z grps" },
{ 0x57, "Scale factor of set" },
{ 0x6d, "Date of birth" },
{ 0x6e, "Directional orientation" },
{ 0x6f, "Number of VFR studies" },
{ 0x70, "R-R low tolerance" },
{ 0x71, "R-R high tolerance" },
{ 0x72, "Prog specific results:" },

{ 0x99, 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::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpADAC( is );
    if( !b ) return 1;

    return 0;
}

```

## 29.35 DumpExamCard.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

PURPOSE. See the above copyright notice for more information.

```

=====*/
/*

Try to extract contents of Philips RAW storage class:

(0002,0002) UI [1.2.840.10008.5.1.4.1.1.66]          # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.46.670589.11.17240.5.23.4.1.3012.2010032409482568018] # 56,1 Media Storage SOP
      Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                # 20,1 Transfer Syntax UID
(0002,0012) UI [1.3.46.670589.11.0.0.51.4.4.1]       # 30,1 Implementation Class UID
(0002,0013) SH [MR DICOM 4.1]                       # 12,1 Implementation Version Name

* Everything done in this code is for the sole purpose of writing interoperable
* software under Sect. 1201 (f) Reverse Engineering exception of the DMCA.
* If you believe anything in this code violates any law or any of your rights,
* please contact us (gdcm-developers@lists.sourceforge.net) so that we can
* find a solution.
*
* Everything you do with this code is at your own risk, since decompression
* algorithm was not written from specification documents.
*
* Special thanks to:
* Triplett, William T for bringing to your attention on this ExamCard stuff
*/
#include "gdcmReader.h"
#include "gdcmDataSet.h"
#include "gdcmPrivateTag.h"
#include "gdcmBase64.h"

#include <iomanip>

static bool compfn(const char *s1, const char *s2)
{
    return strcmp(s1,s2) < 0 ? true : false;
}

static const char *PDFStrings[] = { // Keep me ordered please
    "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_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,
    param_string,
    param_3, // ??
    param_enum,
} 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));
        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];
    int8_t boolean;
    int32_t type;
    uint32_t dim;
    uint32_t v4;
    /*int32_t*/ std::streamoff offset;
    param_type gettype() const { return (param_type)type; }
    uint32_t getdim() const { return dim; }
    void read( std::istream & is )
    {
        is.read( name, 32 + 1);
        //assert( name[32] == 0 ); // fails sometimes...
        // 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 ); // sigh
        is.read( (char*)&boolean, 1);
        assert( boolean == 0 || boolean == 1 ); // some kind of bool...
        is.read( (char*)&type, sizeof( type ) );
        assert( gettypenamefromtype( type ) );
        is.read( (char*)&dim, sizeof( dim ) );
    }
};

```

```

is.read( (char*)&v4, sizeof( v4 ) );
//assert( v4 == 0 ); // always 0 ? sometimes not...
const std::streamoff cur = is.tellg();
is.read( (char*)&offset, sizeof( offset ) );
offset += cur;
}

void print( std::ostream & os ) const
{
    os << name << ", ";
    os << (int)boolean << ", ";
    os << type << ", ";
    os << dim << ", ";
    os << v4 << ", ";
    os << offset << std::endl;
}

void printvalue( std::ostream & os, std::istream & is ) const
{
    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:
        {
            for( uint32_t idx = 0; idx < dim; ++idx )
            {
                if( idx ) os << ", ";
                int32_t v;
                is.read( (char*)&v, sizeof(v) );
                os << v;
            }
        }
        break;
        case param_string:
        {
            std::string v;
            v.resize( dim );
            is.read( &v[0], dim );
            os << v;
        }
        break;
        case param_enum:
        {
            for( uint32_t idx = 0; idx < dim; ++idx )
            {
                if( idx ) os << ", ";
                int32_t v;
                is.read( (char*)&v, sizeof(v) );
                os << v;
            }
        }
        break;
    }
}

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 ?
    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 ?
    const gdcm::PrivateTag pt1(0x2005,0x39,"Philips MR Imaging DD 002");
    if( !ds.FindDataElement( pt1 ) ) return false;
    const gdcm::DataElement &de1 = ds.GetDataElement( pt1 );

    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) );

        // ugly 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 = gdcm::Base64::GetDecodeLength(b64.c_str(), b64.size() );
    }
}

```

```

std::string decoded;
decoded.resize( dlen );
gdcmm::Base64::Decode( &decoded[0], decoded.size(), b64.c_str(), b64.size() );

std::ofstream f64( "soap.xml" );
f64.write( decoded.c_str(), decoded.size() );
f64.close();

ret = true;
}
else
{
    if( del.IsEmpty() ) return false;
    const gdcmm::ByteValue * bv1 = del.GetByteValue();
    std::string s1( bv1->GetPointer() , bv1->GetLength() );

    if( s1 == "IEEE_PDF" )
    {
        // std::cout << "Len= " << bv->GetLength() << std::endl;
    }
    #if 0
        std::string fn = gdcmm::LOComp::Trim( s.c_str() ); // remove trailing space
        std::ofstream out( fn.c_str() );
        out.write( bv->GetPointer(), bv->GetLength() );
        out.close();
    #endif

    std::istringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );

    header h;
    h.read( is );
    #if 0
        std::cout << s0.c_str() << std::endl;
        h.print( std::cout );
    #endif

    assert( is.tellg() == std::streampos(0x20) );
    is.seekg( 0x20 );

    std::vector< param > params;
    param p;
    for( uint32_t i = 0; i < h.getnparams(); ++i )
    {
        p.read( is );
        //p.print( std::cout );
        params.push_back( p );
    }

    std::string fn = gdcmm::LOComp::Trim( s0.c_str() ); // remove trailing space
    assert( isvalidpdfstring( fn.c_str() ) );
    fn += ".csv";
    //fn += ".xml";
    std::ofstream csv( fn.c_str() );

    // let's do some bookkeeping:
    uint32_t nfloats = 0;
    uint32_t nints = 0;
    uint32_t nstrings = 0;
    for( std::vector<param>::const_iterator it = params.begin();
        it != params.end(); ++it )
    {
        param_type type = it->gettype();
        switch( type )
        {
            case param_float:
                nfloats += it->getdim();
                break;
            case param_integer:
                nints += it->getdim();
                break;
            case param_string:
                nstrings += it->getdim();
                break;
            default:
                ;
        }
    }
    #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;
#ifdef
    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 = gdcm::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 = gdcm::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( s1 == "BINARY" )
{
    std::cerr << "BINARY is not handled" << std::endl;
    std::string fn = gdcm::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();

    #if 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];
    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();
    /*
(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\5f\64\
    # 3104,1 ?
(2005,1147) CS [Y ]                                # 2,1 ?
(fffe,e00d)
*/
const gdcm::PrivateTag pt(0x2005,0x32,"Philips MR Imaging DD 002");
if( !ds.FindElement( 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 ) ) return 1;
}

return 0;
}

```

## 29.36 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 VRToType<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;
        }
        gdcm::Element<VR::UL, VM::VM1> el1;
        el1.SetFromDataElement( index );

        gdcm::Element<VR::LO, VM::VM1> el2;
        el2.SetFromDataElement( name );
        // std::cout << el1.GetValue() << " " << el2.GetValue() << std::endl;
        names.insert( std::make_pair( el1.GetValue(), el2.GetValue() ) );
    }

    SequenceOfItems::SizeType s2 = sqi_values->
        GetNumberOfItems();
    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;
        }
        gdcm::Element<VR::FD, VM::VM1_2> el1;
        el1.SetFromDataElement( index2 );

        UL copy = (UL)el1.GetValue();
        #if 1
        std::cout << indent;
        std::cout << "( " << names[ copy ];
        #endif
        // (7fe1,1052) FD 1560 # 8,1 ?
        // (7fe1,1057) LT [MscSkelSup] # 10,1 ?
        //PrivateTag tvalue(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001");
        PrivateTag tvalueint(0x7fe1,0x49,"GEMS_Ultrasound_MovieGroup_001"); // UL
        PrivateTag tvaluefloat1(0x7fe1,0x51,"GEMS_Ultrasound_MovieGroup_001"); // FL
        PrivateTag tvaluefloat(0x7fe1,0x52,"GEMS_Ultrasound_MovieGroup_001"); // FD
        PrivateTag tvalueul(0x7fe1,0x53,"GEMS_Ultrasound_MovieGroup_001"); // UL
        PrivateTag tvaluesl(0x7fe1,0x54,"GEMS_Ultrasound_MovieGroup_001"); // SL
        PrivateTag tvalueob(0x7fe1,0x55,"GEMS_Ultrasound_MovieGroup_001"); // OB
        PrivateTag tvaluetext(0x7fe1,0x57,"GEMS_Ultrasound_MovieGroup_001"); // LT
        PrivateTag tvaluefd(0x7fe1,0x77,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
        PrivateTag tvaluesl3(0x7fe1,0x79,"GEMS_Ultrasound_MovieGroup_001"); // SL / 1-N
        PrivateTag tvaluesl2(0x7fe1,0x86,"GEMS_Ultrasound_MovieGroup_001"); // SL ??
    }
}

```

```

PrivateTag tvaluefd1(0x7fe1,0x87,"GEMS_Ultrasound_MovieGroup_001"); // FD / 1-N
PrivateTag tvaluefloat2(0x7fe1,0x88,"GEMS_Ultrasound_MovieGroup_001"); // FD ??
#endif
std::cout << " ) = ";
#endif
if( ds.FindDataElement( tvalueint ) )
{
    const DataElement & value = ds.GetDataElement( tvalueint );
    gdcm::Element<VR::UL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluefloat1 ) )
{
    const DataElement & value = ds.GetDataElement( tvaluefloat1 );
    gdcm::Element<VR::FL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluefloat ) )
{
    const DataElement & value = ds.GetDataElement( tvaluefloat );
    gdcm::Element<VR::FD,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvaluesl ) )
{
    const DataElement & value = ds.GetDataElement( tvaluesl );
    gdcm::Element<VR::SL,VM::VM1> el2;
    el2.SetFromDataElement( value );
    std::cout << el2.GetValue() << std::endl;
}
else if( ds.FindDataElement( tvalueul ) )
{
    const DataElement & value = ds.GetDataElement( tvalueul );
    gdcm::Element<VR::UL,VM::VM1_n> el2;
    el2.SetFromDataElement( value );
    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( gdcm::PrivateTag const & privtag, const
    gdcm::DataSet & ds,
    gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag ) ) return 1;
    const gdcm::DataElement& seq_values = ds.GetDataElement( privtag );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi = seq_values.
        GetValueAsSQ();

    return PrintNameValueMapping( sqi, sqi_names, indent);
}

bool PrintNameValueMapping3( gdcm::PrivateTag const & privtag1,
    gdcm::PrivateTag const & privtag2, const gdcm::DataSet & ds,
    gdcm::SequenceOfItems *sqi_names, std::string const & indent )
{
    if( !ds.FindDataElement( privtag1 ) )
    {
        assert( 0 );
        return false;
    }
    const gdcm::DataElement& values10name = ds.GetDataElement( privtag1 );
    gdcm::Element<gdcm::VR::LO,gdcm::VM::VM1> el;
    el.SetFromDataElement( values10name );
    std::cout << std::endl;
    std::cout << " <" << el.GetValue().c_str() << ">" << std::endl;

    return PrintNameValueMapping2( privtag2, ds, sqi_names, indent);
}

bool print73( gdcm::DataSet const & ds10, gdcm::SequenceOfItems *sqi_dict
    , std::string const & indent )
{
    const gdcm::PrivateTag tseq_values73(0x7fe1,0x73,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values73 ) )
    {
        std::cout << indent << "No group 73" << std::endl;
        return false;
    }
    const gdcm::DataElement& seq_values73 = ds10.GetDataElement( tseq_values73
    );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values73 =
        seq_values73.GetValueAsSQ();

    size_t ni3 = sqi_values73->GetNumberOfItems();
    for( size_t i3 = 1; i3 <= ni3; ++i3 )
    {
        gdcm::Item &item_73 = sqi_values73->GetItem(i3);
        gdcm::DataSet &ds73 = item_73.GetNestedDataSet();
        assert( ds73.Size() == 3 );
    }
}

```

```

    const gdcmm::PrivateTag tseq_values74name(0x7fe1,0x74,"GEMS_Ultrasound_MovieGroup_001");
    const gdcmm::PrivateTag tseq_values75(0x7fe1,0x75,"GEMS_Ultrasound_MovieGroup_001");
    PrintNameValueMapping3( tseq_values74name, tseq_values75, ds73, sqi_dict, indent);
    std::cout << std::endl;
}
return true;
}

bool print36( gdcmm::DataSet const & ds10, gdcmm::SequenceOfItems *sqi_dict
, std::string const & indent )
{
    const gdcmm::PrivateTag tseq_values36(0x7fe1,0x36,"GEMS_Ultrasound_MovieGroup_001");
    if( !ds10.FindDataElement( tseq_values36 ) )
    {
        std::cout << indent << "No group 36" << std::endl;
        return false;
    }
    const gdcmm::DataElement& seq_values36 = ds10.GetDataElement( tseq_values36
    );
    gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi_values36 =
        seq_values36.GetValueAsSQ();

    size_t ni3 = sqi_values36->GetNumberOfItems();
    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( gdcm::PrivateTag const & privtag0, const
    gdcm::DataSet & subds, gdcm::PrivateTag const & privtag1,
    gdcm::PrivateTag const & privtag2,
    gdcm::SequenceOfItems *sqi_dict, std::string const & indent )
{
    (void)indent;
    if( !subds.FindDataElement( privtag0 ) )
    {
        assert( 0 );
        return 1;
    }
    const gdcm::DataElement& seq_values10 = subds.GetDataElement( privtag0 );
    gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values10 =
        seq_values10.GetValueAssSQ();

    size_t nil = sqi_values10->GetNumberOfItems();
    // assert( nil == 1 );
    for( size_t i1 = 1; i1 <= nil; ++i1 )
    {
        gdcm::Item &item_10 = sqi_values10->GetItem(i1);
        gdcm::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 1;
        }
        const gdcm::DataElement& seq_values20 = ds10.GetDataElement(
            tseq_values20 );
        gdcm::SmartPointer<gdcm::SequenceOfItems> sqi_values20 =
            seq_values20.GetValueAssSQ();

        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;
}

```

## 29.37 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.
=====

```

[illegible]

```

    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)
    // TUSCLIPPAREMTE (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::stringstream is;
    std::string dup( bv->GetPointer(), bv->GetLength() );
    is.str( dup );
    bool b = DumpImageHeaderInfo( is, bv->GetLength() );
    if( !b ) return 1;

#if 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;
}

```

## 29.38 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 > crchheaders;
    crchheaders.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
    crchheaders.push_back( header );
}

std::istream is;
is.str( std::string( buf, 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 = totalsize - offsets[r] - 16;
    else
        sourceLen = offsets[r+1] - 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 > crcheaders;
    crcheaders.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 ) );
#ifdef 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::istringstream is;
        is.str( std::string( buf, 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 << outfile;
        ss << ' ';
        ss << crcheaders[0].imgsize; // FIXME: Assume all header are identical !
        ss << ' ';
        ss << nframes;
        ss << ".raw";
        std::ofstream os( ss.str().c_str(), std::ios::binary );
        outbuf.resize( buf_size ); // overallocated + 16
        char *buffer = &outbuf[0];

        hframe header;
        for( unsigned int r = 0; r < nframes; ++r )
        {
            is.read( (char*)&header, sizeof( header ) );
#ifdef 0
            std::cout << header.val0
                << " " << header.val1[0]
                << " " << header.val1[1]
                << " " << header.val2[0]
                << " " << header.val2[1]
                << " " << header.imgsize << std::endl;
#endif

```

```

    assert( header == crcheaders[r] );

    is.read( buffer, buf_size - 16 );
    os.write( buffer, header.imgsize );
}
assert( is.tellg() == totalsize );
os.close();

return true;
}

#ifndef 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 = sqi1->GetItem(item);
    DataSet &ds2 = item1.GetNestedDataSet();

    // (200d,300d) LO 28 UDM_USD_DATATYPE_DIN_2D_ECHO
    const PrivateTag tdatatype(0x200d,0x300d,"Philips US Imaging DD 033");
    if( !ds2.FindDataElement( tdatatype ) ) return 1;
    const DataElement& datatype = ds2.GetDataElement( tdatatype );
    const ByteValue *bvdatatype = datatype.GetByteValue();
    if( !bvdatatype ) return 1;

    const PrivateTag tseq2(0x200d,0x3cf1,"Philips US Imaging DD 045");
    if( !ds2.FindDataElement( tseq2 ) ) return 1;
    const DataElement& seq2 = ds2.GetDataElement( tseq2 );

    SmartPointer<SequenceOfItems> sqi2 = seq2.
        GetValueAsSQ();
    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( outfilename, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->
            GetLength() ) )
        {
            return 1;
        }
    }
    else if( strcmp(bv->GetPointer(), "None", 4) == 0 )
    {
        if( !ProcessNone( outfilename, nslicesref, zallocref, bv2->GetPointer(),
            std::streampos(bv2->GetLength()), bv3->GetPointer(), bv3->
            GetLength() ) )
        {
            return 1;
        }
    }
    else
    {
        std::string str( bv->GetPointer(), bv->GetLength() );
        std::cerr << "Unhandled: " << str << std::endl;
        return 1;
    }
}

return 0;
}

```

## 29.39 DumpToSQLITE3.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * Ref:
 * http://massmail.spl.harvard.edu/public-archives/slicer-devel/2010/004408.html
 *
 * Implementation details:
 * http://www.sqlite.org/c3ref/bind_blob.html
 * http://www.adp-gmbh.ch/sqlite/bind_insert.html
 */
#include "gdcmScanner.h"
#include "gdcmDirectory.h"
#include "gdcmTag.h"
#include "gdcmTrace.h"

#include "sqlite3.h"

#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        return 1;
    }
    time_t time_start = time(0);

    gdcm::Trace::SetDebug( false );
    gdcm::Trace::SetWarning( false );
    const char *inputdirectory = argv[1];

    gdcm::Directory d;
    unsigned int nfiles = d.Load( inputdirectory, true );

```

```

gdcmm::Scanner s;
using gdcmm::Tag;
s.AddTag( Tag(0x20,0xd) ); // Study Instance UID
s.AddTag( Tag(0x20,0xe) ); // Series Instance UID

bool b0 = s.Scan( d.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 gdcmm::Directory;
using gdcmm::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;
}

```

## 29.40 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 "gdcmlReader.h"
#include "gdcmlWriter.h"
#include "gdcmlItem.h"
#include "gdcmlImageReader.h"
#include "gdcmlSequenceOfItems.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"
/*
Usage:
DuplicatePCDE gdcmlData/D_CLUNIE_CT1_J2KI.dcm out.dcm

aka:
medical.nema.org/medical/dicom/DataSets/WG04/IMAGES/J2KI/CT1_J2KI

See:
gdcmlConformanceTests/CT1_J2KI_DuplicatePCDE.dcm

Original thread can be found at:
http://groups.google.com/group/comp.protocols.dicom/browse_thread/thread/82f28c4db28963af

Question:
1.
There is no restriction for a specific Private Creator Data Element
(PCDE) to be unique within the same group, right ?
Decoders of Private Data would have to handle the case where a PCDE
would be repeated and should NOT stop on the first instance of a
particular PCDE, right ?

Eg. when searching for the tag associated with
(0x0029,0x0010,"SIEMENS CSA HEADER") in the following (pseudo)
dataset:

(0029,0010) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator
(0029,0011) LO [SIEMENS MEDCOM HEADER] # 22, 1
PrivateCreator
(0029,0012) LO [SIEMENS MEDCOM HEADER2] # 22, 1
PrivateCreator
(0029,0013) LO [SIEMENS CSA HEADER] # 18, 1
PrivateCreator

```

```

(0029,1008) CS [IMAGE NUM 4] # 12, 1
CSAImageHeaderType
(0029,1009) LO [20050723] # 8, 1
CSAImageHeaderVersion
(0029,1010) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo
(0029,1018) CS [MR] # 2, 1
CSASeriesHeaderType
(0029,1019) LO [20050723] # 8, 1
CSASeriesHeaderVersion
(0029,1020) OB 53\56\31\30\04\03\02\01\2c\00\00\00\4d
\00\00\00\55\73\65\64\50\61... # 51520, 1 CSASeriesHeaderInfo
(0029,1131) LO [4.0.163088300] # 14, 1
PMTFInformation1
(0029,1132) UL 32768 # 4, 1
PMTFInformation2
(0029,1133) UL 0 # 4, 1
PMTFInformation3
(0029,1134) CS [DB TO DICOM] # 12, 1
PMTFInformation4
(0029,1260) ?? 63\6f\6d\20 # 4, 1
Unknown Tag & Data
(0029,1310) OB 53\56\31\30\04\03\02\01\38\00\00\00\4d
\00\00\00\45\63\68\6f\4c\69... # 6788, 1 CSAImageHeaderInfo

```

one should return two instances, correct ?

Answer:

I would say that this is covered in principle by the PS 3.5 7.1  
 "The Data Elements ... shall occur at most once in a Data Set"  
 rule, since the data element is defined by the tuple  
 (private creator,gggg,ee) where xxee is the element  
 number and xx is arbitrary and has no inherent meaning and  
 does not serve to disambiguate the data element.

E.g.:

```

(0019,0030) Private Creator ID = "Smith"
...
(0019,0032) Private Creator ID = "Smith"
...
(0019,3015) Fractal Index = "32"
...
(0019,3215) Fractal Index = "32"

```

would be illegal because even though they are assigned different  
 (completely arbitrary) blocks, with the same group, element  
 number and private creator, (0019,3015) and (0019,3215) are the  
 "same" data element.

\*/

```

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcm::File &file = reader.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    // Let's get all private element from group 0x9:
    /*
(0009,0010) LO [GEMS_IDEN_01] # 12,1 Private Creator
(0009,1001) LO [GE_GENESIS_FF ] # 14,1 Full fidelity
(0009,1002) SH [CT01] # 4,1 Suite id
(0009,1004) SH [HiSpeed CT/i] # 12,1 Product id
(0009,1027) SL 862399669 # 4,1 Image actual date
(0009,1030) SH (no value) # 0,1 Service id
(0009,1031) SH (no value) # 0,1 Mobile location number
(0009,10e6) SH [05] # 2,1 Genesis Version - now

```

```

(0009,10e7) UL 973283917                                     # 4,1 Exam Record checksum
(0009,10e9) SL 862399669                                     # 4,1 Actual series data time stamp
*/
gdcmm::Tag start(0x0009,0x0);
// Create a temporary duplicate dataset, since we cannot insert data element as we go over them (std::set
// would reorganize itself as we go over it ...)
gdcmm::DataSet dup;
gdcmm::Tag new_private(0x0009,0x0);
while (start.GetGroup() == 0x9 )
{
    const gdcmm::DataElement& de = ds.FindNextDataElement(start);
    const gdcmm::Tag &t = de.GetTag();
    if( t.IsPrivateCreator() )
    {
        std::cout << t << std::endl;
        // Ok let's duplicate into the next available attribute:
        gdcmm::DataElement duplicate = de;
        duplicate.GetTag().SetElement( (uint16_t)(t.GetElement() + 1) );
        dup.Insert( duplicate );
        new_private = duplicate.GetTag();
    }
    else if( t.IsPrivate() && !t.IsPrivateCreator() )
    {
        //std::cout << de << std::endl;
        std::string owner = ds.GetPrivateCreator( de.GetTag() );
        //std::cout << owner << std::endl;
        gdcmm::DataElement duplicate = de;
        duplicate.GetTag().SetPrivateCreator( new_private );
        if( const gdcmm::ByteValue *bv = duplicate.GetByteValue() )
        {
            // Warning: when doing : duplicate = de, only the pointer to the ByteValue is passed
            // (to avoid large memory duplicate). We need to explicitly duplicate the bytevalue ourselves:
            gdcmm::ByteValue *dupbv = new gdcmm::ByteValue( bv->GetPointer(),
                bv->GetLength() );
            // Let's recognize the duplicated ASCII-type elements:
            if( duplicate.GetVR() & gdcmm::VR::VRASCII )
                dupbv->Fill( 'X' );
            duplicate.SetValue( *dupbv );
        }
        dup.Insert( duplicate );
    }
    start = t;
    // move to next possible 'public' element
    start.SetElement( (uint16_t)(start.GetElement() + 1) );
}

gdcmm::DataSet::ConstIterator it = dup.Begin();
for( ; it != dup.End(); ++it )
{
    ds.Insert( *it );
}

gdcmm::Writer w;
w.SetFile( file );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 29.41 ELSCINT1WaveToText.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

```

```

=====*/
#include "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' << " " << '\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] << '\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] << '\t' << buffer[i+1] << std::endl;
            if (buffer[i+75] == -16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\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] << '\t' << buffer[i+1] << std::endl;
        }
        else
        {
            if (buffer[i+75] == 0)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == 16384)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
            if (buffer[i+75] == 256)
                text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 0 << '\t' << " " << '\t' << " " << '\t' << " " << '\t' << " " << std::endl;
        }
    }
}

```

```

        if (buffer[i+75] == -32768)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\t' << " "
            << '\t' << buffer[i+74] << '\t' << " " << '\t' << " "
            << std::endl;
        if (buffer[i+75] == -16384)
            text_file << buffer[i+74] << '\t' << buffer[i+75] << '\t' << 1 << '\t' << " " << '\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' << " "
            << '\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;
}

```

## 29.42 EncapsulateFileInRawData.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmAnonymizer.h"
#include "gdcmWriter.h"
#include "gdcmUIDGenerator.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmSystem.h"

#include "magic.h" // libmagic, API to file command line tool

/*
 * Let say you want to encapsulate a file type that is not defined in DICOM (exe, zip, png)
 * PNG is a bad example, unless it contains transparency (which has been deprecated).
 * It will take care of dispatching each chunk to an appropriate data item (pretty much like
 * WaveformData)
 */

```



```

* Usage:
* ./EncapsulateFileInRawData large_input_file.exe large_input_file.dcm
*/

// TODO:
// $ file -bi /tmp/gdcm-2.1.0.pdf
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " inputfile output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    if( !gdcm::System::FileExists( filename ) ) return 1;

    size_t s = gdcm::System::FileSize(filename);
    if( !s ) return 1;

    magic_t cookie = magic_open(MAGIC_NONE);
    const char * file_type = magic_file(cookie, filename);
    if( !file_type ) return 1;
    magic_close(cookie);

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    //gdcm::DataSet &ds = file.GetDataSet();
    //w.SetCheckFileMetaInformation( true );
    w.SetFileName( 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;
}

```

## 29.43 ExtractEncapsulatedFile.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
* This example shows how one from C# context can extract a binary blob
* and write out as a file.
* This example is meant for pdf encapsulated file, but can be adapted for other type
* of binary blob.

```

```

*
* DICOM file is:
* ...
* (0042,0010) ST (no value available) # 0, 0 DocumentTitle
* (0042,0011) OB 25\50\44\46\2d\31\2e\32\20\0d\25\e2\e3\cf\d3\20\0d\31\30\20\30\20... # 40718, 1
* EncapsulatedDocument
* (0042,0012) LO [application/pdf] # 16, 1 MIMETimeTypeOfEncapsulatedDocument
* ...
*
* Usage:
* $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
* $ mono bin/ExtractEncapsulatedFile.exe some_pdf_encapsulated.dcm
*/
using System;
using gdcm;

public class ExtractEncapsulatedFile
{
    public static int Main(string[] args)
    {
        string file = args[0];
        Reader reader = new Reader();
        reader.SetFileName( file );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }

        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();
        Tag tencapsulated_stream = new Tag(0x0042,0x0011); // Encapsulated Document
        if( !ds.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;
    }
}

```

## 29.44 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 &nesteddds = item.GetNestedDataSet();

    if( ! nesteddds.FindDataElement( gdcm::Tag( 0x0400,0x0520 ) ) ) return 1;

    const gdcm::DataElement &EncryptedContent = nesteddds.
        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;
}

```

## 29.45 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;
}

```

## 29.46 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];

        // 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();

        // 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 ];

        // 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;
}
}

```

## 29.47 ExtractImageRegion.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.

=====*/

/*
 * This small code shows how to use the gdcml.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 *
 * Usage:
 * $ LD_LIBRARY_PATH=. CLASSPATH=gdcml.jar:. java ExtractImageRegion input.dcm
 */
import gdcml.*;
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");
        }
    }
}

```

## 29.48 ExtractImageRegionWithLUT.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/

/*
 * This small code shows how to use the gdcm.ImageRegionReader API
 * In this example we are taking each frame by frame and dump them to
 * /tmp/frame.raw.
 * Furthermore we are applying the LUT on this image.
 * Special care should be taken in case the image is not PALETTE COLOR
 *
 * Usage:
 * $ bin/ExtractImageRegionWithLUT.exe input.dcm
 *
 * Example:
 * $ bin/ExtractImageRegionWithLUT.exe gdcmData/rle16loo.dcm
 * $ md5sum /tmp/frame_rgb.raw
 * 73bf61325fdb6e2830244a2b7b0c4ae2 /tmp/frame_rgb.raw
 * $ gdcimg --depth 16 --spp 3 --size 600,430 /tmp/frame_rgb.raw rgb.dcm
 * $ gdcviewer rgb.dcm
 */
using System;
using gdcm;

public class ExtractImageRegion
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        // instantiate the reader:
        gdcm.ImageRegionReader reader = new
            gdcm.ImageRegionReader();
        reader.SetFileName( filename );

        // pull DICOM info:
        if (!reader.ReadInformation()) return 1;
        // Get file infos
        gdcm.File f = reader.GetFile();

        gdcm.LookupTable lut = reader.GetImage().GetLUT();

        // get some info about image
        UIntArrayType dims = ImageHelper.GetDimensionsValue(f);
        PixelFormat pf = ImageHelper.GetPixelFormatValue (f);
        int pixelsize = pf.GetPixelSize();
    }
}

```

```

// buffer to get the pixels
byte[] buffer = new byte[ dims[0] * dims[1] * pixelsize ];

// output buffer for the RGB decoded image:
byte[] buffer2 = new byte[ dims[0] * dims[1] * pixelsize * 3 ];

// define a simple box region.
BoxRegion box = new BoxRegion();
for (uint z = 0; z < dims[2]; z++)
{
    // Define that I want the image 0, full size (dimx x dimy pixels)
    // and do that for each z:
    box.SetDomain(0, dims[0] - 1, 0, dims[1] - 1, z, z);
    //System.Console.WriteLine( box.ToString() );
    reader.SetRegion( box );

    // reader will try to load the uncompressed image region into buffer.
    // the call returns an error when buffer.Length is too small. For instance
    // one can call:
    // uint buf_len = reader.ComputeBufferLength(); // take into account pixel size
    // to get the exact size of minimum buffer
    if (reader.ReadIntoBuffer(buffer, (uint)buffer.Length))
    {
        if( !lut.Decode( buffer2, (uint)buffer2.Length, buffer, (uint)buffer.Length ) )
        {
            throw new Exception("can't decode");
        }

        using (System.IO.Stream stream =
            System.IO.File.Open(@"tmp/frame_rgb.raw",
                System.IO.FileMode.Create))
        {
            System.IO.BinaryWriter writer = new System.IO.BinaryWriter(stream);
            writer.Write(buffer2);
        }
    }
    else
    {
        throw new Exception("can't read pixels error");
    }
}

return 0;
}
}

```

## 29.49 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 <openjpeg.h>
#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 "gdcmMediaStorage.h"
#include "gdcmWriter.h"
#include "gdcmItem.h"
#include "gdcmImageReader.h"
#include "gdcmAttribute.h"
#include "gdcmFile.h"
#include "gdcmTag.h"
#include "gdcmTransferSyntax.h"
#include "gdcmUIDGenerator.h"
#include "gdcmAnonymizer.h"
#include "gdcmStreamImageWriter.h"
#include "gdcmImageHelper.h"
#include "gdcmTrace.h"

void error_callback(const char *msg, void *) {
    (void)msg;
}
void warning_callback(const char *msg, void *) {
    (void)msg;
}
void info_callback(const char *msg, void *) {
    (void)msg;
}

bool Write_Resolution(gdcm::StreamImageWriter & theStreamWriter, const char *
    filename, int res, std::ostream& of, int flag, gdcm::SequenceOfItems *sq, int
    No_Of_Resolutions)
{
    std::ifstream is;
    is.open( filename, std::ios::binary );
    opj_dparameters_t parameters; /* decompression parameters */
    opj_event_mgr_t event_mgr; /* event manager */
    opj_dinfo_t* dinfo; /* handle to a decompressor */
    opj_cio_t *cio;
    opj_image_t *image = NULL;
    // FIXME: Do some stupid work:
    is.seekg( 0, std::ios::end);
    std::streampos buf_size = is.tellg();
    char *dummy_buffer = new char[(unsigned int)buf_size];
    is.seekg(0, std::ios::beg);
    is.read( dummy_buffer, buf_size);
    unsigned char *src = (unsigned char*)dummy_buffer;
    uint32_t file_length = (uint32_t)buf_size; // 32bits truncation should be ok since DICOM cannot have
        larger than 2Gb image

    /* configure the event callbacks (not required) */
    memset(&event_mgr, 0, sizeof(opj_event_mgr_t));
    event_mgr.error_handler = error_callback;
    event_mgr.warning_handler = warning_callback;
    event_mgr.info_handler = info_callback;

    /* set decoding parameters to default values */
    opj_set_default_decoder_parameters(&parameters);

    // default blindly copied
    parameters.cp_layer=0;
    parameters.cp_reduce= res;
    // parameters.decod_format=-1;
    // parameters.cod_format=-1;

    const char jp2magic[] = "\x00\x00\x00\x0C\x6A\x50\x20\x20\x0D\x0A\x87\x0A";
    if( memcmp( src, jp2magic, sizeof(jp2magic) ) == 0 )
    {
        /* JPEG-2000 compressed image data ... sigh */
        // gdcmData/ELSCINT1_JP2vsJ2K.dcm
        // gdcmData/MAROTECH_CT_JP2Lossy.dcm
        //gdcmWarningMacro( "J2K start like JPEG-2000 compressed image data instead of codestream" );
        parameters.decod_format = 1; //JP2_CFMT;
        //assert(parameters.decod_format == JP2_CFMT);
    }
    else
    {
        /* JPEG-2000 codestream */

```

```

    //parameters.decod_format = J2K_CFMT;
    //assert(parameters.decod_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);
    //gdcmErrorMacro( "opj_decode failed" );
    return 1;
}

    opj_cp_t * cp = ((opj_jp2_t*)dinfo->jp2_handle)->j2k->cp;
    opj_tcp_t * tcp = &cp->tcps[0];
    opj_tccp_t * tccp = &tcp->tccps[0];
    /* std::cout << "\n No of Cols In Image" << image->x1;
    std::cout << "\n No of Rows In Image" << image->y1;
    std::cout << "\n No of Components in Image" << image->numcomps;
    std::cout << "\n No of Resolutions"<< tccp->numresolutions << "\n";
*/

    opj_j2k_t* j2k = NULL;
    opj_jp2_t* jp2 = NULL;
    jp2 = (opj_jp2_t*)dinfo->jp2_handle;
    int reversible = jp2->j2k->cp->tcps->tccps->qmfbid;
    //std::cout << reversible;
    int compno = 0;
    opj_image_comp_t *comp = &image->comps[compno];
    int Dimensions[2];
    Dimensions[0] = comp->w;
    Dimensions[1] = comp->h;
    opj_cio_close(cio);
    unsigned long len = Dimensions[0]*Dimensions[1] * image->numcomps;
    //std::cout << "\nTest" <<image->comps[0].factor;
    char *raw = new char[len];
    for (unsigned int compno = 0; compno < (unsigned int)image->numcomps; compno++)
    {
        opj_image_comp_t *comp = &image->comps[compno];

        int w = image->comps[compno].w;
        int h = image->comps[compno].h;
        uint8_t *data8 = (uint8_t*)raw + compno;
        for (int i = 0; i < w * h ; i++)
        {
            int v = image->comps[compno].data[i];
            *data8 = (uint8_t)v;
            data8 += image->numcomps;
        }
    }

    gdcm::Writer w;
    gdcm::File &file = w.GetFile();
    gdcm::DataSet &ds = file.GetDataSet();

    file.GetHeader().SetDataSetTransferSyntax(
        gdcm::TransferSyntax::ExplicitVRLittleEndian );

    gdcm::UIDGenerator uid;
    gdcm::DataElement de( gdcm::Tag(0x8,0x18) ); // SOP Instance UID
    de.SetVR( gdcm::VR::UI );
    const char *u = uid.Generate();
    de.SetByteValue( u, strlen(u) );
    ds.Insert( de );

    gdcm::DataElement del( gdcm::Tag(0x8,0x16) );
    del.SetVR( gdcm::VR::UI );

```

```

gdcM::MediaStorage ms( gdcM::MediaStorage::CTImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()));
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcM::DataElement de2( gdcM::Tag(0x28,0x04) );
//de.SetTag(gdcM::Tag(0x28,0x04));
de2.SetVR( gdcM::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcM::Attribute<0x0028,0x0010> row = {image->comps[0].w};
//row.SetValue(512);
ds.Insert( row.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcM::Attribute<0x0028,0x0011> col = {image->comps[0].h};
ds.Insert( col.GetAsDataElement() );
gdcM::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0002> at1 = {image->numcomps};
ds.Insert( at1.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcM::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );

if (flag == 1)
{
    for (int i=0; i < No_Of_Resolutions; i++)
    {
        int a = 1;
        int b = 1;

        while(a!=(No_Of_Resolutions)-i))
        {
            b = b*2;
            a = a+1;
        }
        uint16_t row = (image->y1)/b;
        uint16_t col = (image->x1)/b;
        //std::cout << row;
        gdcM::Element<gdcM::VR::IS,gdcM::VM::VM1> el2;
        el2.SetValue(i+1);
        gdcM::DataElement rfn = el2.GetAsDataElement(); //ulr --> upper
            left row
        rfn.SetTag( gdcM::Tag(0x0008,0x1160) );

        gdcM::Element<gdcM::VR::US,gdcM::VM::VM2> el;
        el.SetValue(1,0);
        el.SetValue(1,1);
        gdcM::DataElement ulr = el.GetAsDataElement(); //ulr --> upper
            left col/row
        ulr.SetTag( gdcM::Tag(0x0048,0x0201) );

        gdcM::Element<gdcM::VR::US,gdcM::VM::VM2> el1;
        el1.SetValue(col,0);
        el1.SetValue(row,1);
        gdcM::DataElement brr = el1.GetAsDataElement();
        brr.SetTag( gdcM::Tag(0x0048,0x0202) ); //brr --> bottom right col/row
        gdcM::Item it;
        gdcM::DataSet &nds = it.GetNestedDataSet();
        nds.Insert( rfn );
        nds.Insert( ulr );
        nds.Insert( brr );

        sq->AddItem(it);
    }

    gdcM::Writer w1;
    gdcM::File &file1 = w1.GetFile();

```

```

gdcmm::DataSet &ds1 = file1.GetDataSet();
file1.GetHeader().SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian );

gdcmm::UIDGenerator uid1;
gdcmm::DataElement dea( gdcmm::Tag(0x8,0x18) ); // SOP Instance UID
dea.SetVR( gdcmm::VR::UI );
const char *u1 = uid1.Generate();
dea.SetByteValue( u1, strlen(u1) );
ds1.Insert( dea );

gdcmm::DataElement deb( gdcmm::Tag(0x8,0x16) );
deb.SetVR( gdcmm::VR::UI );
gdcmm::MediaStorage msl(
    gdcmm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
deb.SetByteValue( msl.GetString(), strlen( msl.GetString() ) );
ds1.Insert( deb );

const char mystr1[] = "MONOCHROME2 ";
gdcmm::DataElement dec( gdcmm::Tag(0x28,0x04) );
//de.SetTag( gdcmm::Tag(0x28,0x04) );
dec.SetVR( gdcmm::VR::CS );
dec.SetByteValue( mystr1, strlen( mystr1 ) );
ds1.Insert( dec );

gdcmm::Attribute<0x0028,0x0010> row1 = {image->y1};
//row.SetValue(512);
ds1.Insert( row1.GetAsDataElement() );
// w.SetCheckFileMetaInformation( true );
gdcmm::Attribute<0x0028,0x0011> col1 = {image->x1};
ds1.Insert( col1.GetAsDataElement() );
gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames1 = {tccp->numresolutions};
ds1.Insert( Number_Of_Frames1.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0100> ata = {8};
ds1.Insert( ata.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0002> atb = {image->numcomps};
ds1.Insert( atb.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0101> atc = {8};
ds1.Insert( atc.GetAsDataElement() );

gdcmm::Attribute<0x0028,0x0102> atd = {7};
ds1.Insert( atd.GetAsDataElement() );

theStreamWriter.SetFile(file1);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::VR::SQ);
//des.SetVR(gdcmm::VM::VM1);
des.SetValue(*sq);
des.SetVLToUndefined();

ds1.Insert( des );

if (!theStreamWriter.WriteImageInformation()){
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}

theStreamWriter.SetFile(file);

if (!theStreamWriter.CanWriteFile()){
    delete [] raw;
    std::cout << "Not able to write";
    return 0; //this means that the file was unwritable, period.
    //very similar to a ReadImageInformation failure
}
else
    std::cout<<"\nable to read";

// Important to write here
std::vector<unsigned int> extent = gdcmm::ImageHelper::GetDimensionsValue
    (file);

```

```

unsigned short xmax = extent[0];
unsigned short ymax = extent[1];
unsigned short theChunkSize = 4;
unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = extent[2];
std::cout << "\n"<<xmax << "\n" << ymax<<"\n"<<zmax<<"\n" << image->numcomps<<"\n";

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.
for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" <<len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(raw[prevLen]), len);
        std::cout << "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z= " << z <<
std::endl;
            delete [] raw;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
    }
}
delete raw;

delete[] src; //FIXME

if(dinfo) {
    opj_destroy_decompress(dinfo);
}

opj_image_destroy(image);

return true;
}

bool Different_Resolution( gdcm::StreamImageWriter & theStreamWriter, const char *
    filename, int res, std::ostream& of)
{
    //std::vector<std::string>::const_iterator it = filenames.begin();
    bool b = true;
    int flag = 1;

    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new
        gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    for(int i = res-1; i>=0; --i)
    {
        b = b && Write_Resolution( theStreamWriter, filename, i, of ,flag,sq,res);
        // b = b && Get_Resolution( theStreamWriter, filename, i, of ,0);
        flag = 0;
    }
    //b = b && Get_Lowest_Resolution( writer, sq, filename, res-1 );
    //b = b && PopulateSingleFile( 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;
}

```

## 29.50 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;
}
}

```

## 29.51 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 del( gdcm::Tag(0x8,0x16) );
    del.SetVR( gdcm::VR::UI );
    gdcm::MediaStorage ms(
        gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
    );
    del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
    ds.Insert( del );

    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() );

```



```

gdcmm::Attribute<0x0028,0x0103> at5 = {0};
ds.Insert( at5.GetAsDataElement() );

//de.SetTag(gdcmm::Tag(0x7fe0,0x0010));
//ds.Insert(de);

gdcmm::StreamImageWriter theStreamWriter;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
    gdcmm::SequenceOfItems();
sq->SetLengthToUndefined();

uint16_t row1 = 256;
uint16_t col1 = 256;
//std::cout << row;

gdcmm::Element<gdcmm::VR::IS,gdcmm::VM::VM1> el2;
el2.SetValue(1);
gdcmm::DataElement rfn = el2.GetAsDataElement(); //rfn --->
    reference frame number
rfn.SetTag( gdcmm::Tag(0x0008,0x1160) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> el;
el.SetValue(1,0);
el.SetValue(1,1);
gdcmm::DataElement ulr = el.GetAsDataElement(); //ulr --> upper
    left col/row
ulr.SetTag( gdcmm::Tag(0x0048,0x0201) );

gdcmm::Element<gdcmm::VR::US,gdcmm::VM::VM2> ell;
ell.SetValue(col1,0);
ell.SetValue(row1,1);
gdcmm::DataElement brr = ell.GetAsDataElement();
brr.SetTag( gdcmm::Tag(0x0048,0x0202) ); //brr --> bottom right col/row

gdcmm::Item it;
gdcmm::DataSet &nds = it.GetNestedDataSet();
nds.Insert( rfn );
nds.Insert( ulr );
nds.Insert( brr );

sq->AddItem(it);

gdcmm::DataElement des( gdcmm::Tag(0x0048,0x0200) );
des.SetVR(gdcmm::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 =
    gdcmm::ImageHelper::GetDimensionsValue(file);

unsigned short xmax = extent[0];
unsigned short ymax = extent[1];
unsigned short theChunkSize = 1;
unsigned short ychunk = extent[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = extent[2];

```

```

std::cout << xmax << ymax << zmax;

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.
for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" << len;
        char* finalBuffer = new char[len];
        memcpy(finalBuffer, &(buffer[prevLen]), len);
        std::cout << "\nable to write";
        if (!theStreamWriter.Write(finalBuffer, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z = " << z <<
std::endl;
            delete [] buffer;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer;
        prevLen += len;
    }
}
delete buffer;

uint16_t firstTag1 = 0xfffe;
uint16_t secondTag1 = 0xe0dd;
uint32_t thirdTag1 = 0x00000000;
//uint16_t fourthTag1 = 0xffff;
const int theBufferSize1 = 2*sizeof(uint16_t)+sizeof(uint32_t);
char* tmpBuffer2 = new char[theBufferSize1];
memcpy(&(tmpBuffer2[0]), &firstTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[sizeof(uint16_t)]), &secondTag1, sizeof(uint16_t));
memcpy(&(tmpBuffer2[2*sizeof(uint16_t)]), &thirdTag1, sizeof(uint32_t));
//memcpy(&(tmpBuffer2[3*sizeof(uint16_t)]), &fourthTag1, sizeof(uint16_t));
assert( of && !of.eof() && of.good() );
of.write(tmpBuffer2, theBufferSize1);
of.flush();
assert( of );

return 0;
}

```

## 29.52 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;
    }
}

```

## 29.53 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" );
}
}

```

## 29.54 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.TType.
            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;
}
}

```

## 29.55 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 gdcmlData/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;
    }
}

```

## 29.56 FindAllPatientName.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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*',gdcm.VL(2))
33
34 ds = gdcm.DataSet()
35 ds.Insert(de)
36
37 cnf = gdcm.CompositeNetworkFunctions()
38 theQuery = cnf.ConstructQuery (gdcm.ePatientRootType,gdcm.ePatient,ds)
39
40 #print theQuery.ValidateQuery()
41
42 # prepare the variable for output
43 ret = gdcm.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]

```

## 29.57 FixBrokenJ2K.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 "gdcmSequenceOfFragments.h"
#include "gdcmFile.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".
 */
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();
    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();

    const char sig[] = "\x00\x00\x00\x00\x6A\x70\x32\x63";
    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;
}

```

## 29.58 FixCommaBug.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 "Success!"
77 sys.exit(0) # success

```

## 29.59 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 false;
    }

    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() );
}

gdcmm::DataElement pixeldata( gdcmm::Tag(0x7fe0,0x0010) );
pixeldata.SetVR( gdcmm::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(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian);

gdcmm::Writer writer;
writer.SetFileName( outfilename );
writer.SetFile( reader.GetFile() );
writer.Write();

std::cout << "Success !" << std::endl;

return 0;
}

```

## 29.60 gdcmmorthoplanes.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 "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 "gdcmSystem.h"
#include "gdcmDirectory.h"
#include "gdcmIPPSorter.h"

#ifdef vtkFloatingPointType
#define vtkFloatingPointType float
#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;
    ippszspacing = 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
    ippszspacing = reader->GetOutput()->GetSpacing()[2];
    ippszspacing = 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], ippszspacing );
    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 wl[2];
planeWidgetZ->GetWindowLevel(wl);

// Add a 2D image to test the GetReslice method
//
vtkImageMapToColors* colorMap = vtkImageMapToColors::New();
colorMap->PassAlphaToOutputOff();
colorMap->SetActiveComponent(0);
colorMap->SetOutputFormatToLuminance();
#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();
v16->Delete();
reader->Delete();

return 0;
}

```

## 29.61 gdcmreslice.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "vtkRenderer.h"
#include "vtkAssembly.h"
#include "vtkImageFlip.h"
#include "vtkImageReslice.h"
#include "vtkRenderWindow.h"
#include "vtkAnnotatedCubeActor.h"
#include "vtkTransform.h"
#include "vtkAxesActor.h"
#include "vtkTextProperty.h"
#include "vtkCaptionActor2D.h"
#include "vtkPropAssembly.h"
#include "vtkOrientationMarkerWidget.h"
#include "vtkRenderWindowInteractor.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkImageData.h"
#include "vtkLookupTable.h"
#include "vtkTexture.h"
#include "vtkPlaneSource.h"

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();
#ifdef (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();
#ifdef (VTK_MAJOR_VERSION >= 6)
    planeMapper->SetInputConnection(plane->GetOutputPort());
#else
    planeMapper->SetInput(plane->GetOutput());
#endif

// Actor
vtkActor* planeActor = vtkActor::New();
planeActor->SetTexture(texture);
planeActor->SetMapper(planeMapper);
planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

// 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;
}

```

## 29.62 gdcmrtionplan.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkImageData.h"
#include "vtkPointData.h"
#include "vtkPolyData.h"
#include "vtkProperty.h"
#include "vtkPolyDataMapper.h"
#include "vtkActor.h"
#include "vtkRenderer.h"
#include "vtkCellArray.h"
#include "vtkPoints.h"
#include "vtkDoubleArray.h"
#include <vtkXMLImageDataWriter.h>
#include <vtkXMLPolyDataWriter.h>
#include <vtkRenderWindowInteractor.h>
#include <vtkImageColorViewer.h>

#include "gdcmReader.h"
#include "gdcmAttribute.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];

    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,03a2) SQ                                     # u/1,1 Ion Beam Sequence
  (ffff,e000) na (Item with undefined length)        #
    (0008,1040) LO [Test]                            # 4,1 Institutional Department Name
    (300a,00b2) SH (no value)                        # 0,1 Treatment Machine Name
    (300a,00b3) CS [MU]                              # 2,1 Primary Dosimeter Unit
    (300a,00c0) IS [1 ]                             # 2,1 Beam Number
    (300a,00c2) LO [1 ]                             # 2,1 Beam Name
    (300a,00c4) CS [STATIC]                          # 6,1 Beam Type
    (300a,00c6) CS [PROTON]                          # 6,1 Radiation Type
    (300a,00ce) CS [TREATMENT ]                     # 10,1 Treatment Delivery Type
    */

```

```

(300a,00d0) IS [ 0 ] # 2,1 Number of Wedges
(300a,00e0) IS [ 1 ] # 2,1 Number of Compensators
(300a,00ed) IS [ 0 ] # 2,1 Number of Boli
(300a,00f0) IS [ 1 ] # 2,1 Number of Blocks
(300a,0110) IS [ 2 ] # 2,1 Number of Control Points
(300a,02ea) SQ # u/1,1 Ion Range Compensator Sequence
(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\3
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,e00d)
*/
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();
gdcmm::Tag tbeamsq(0x300a,0x03a2);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &tbeamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.
    GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item &item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item &item = sqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x02ea);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorsq = nestedds.
    GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq.
    GetValueAsSQ();
const gdcmm::Item &item2 = ssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorthicknessdata = nestedds2.
    GetDataElement( tcompensatorthicknessdata );
// std::cout << compensatorthicknessdata << std::endl;
gdcmm::Attribute<0x300a,0x00ec> at;
at.SetFromDataElement( compensatorthicknessdata );
const double* pts = at.GetValues();
// (300a,00e7) IS [35] # 2,1 Compensator Rows
gdcmm::Attribute<0x300a,0x00e7> at1;
const gdcmm::DataElement &compensatorrows = nestedds2.
    GetDataElement( at1.GetTag() );
at1.SetFromDataElement( compensatorrows );
std::cout << at1.GetValue() << std::endl;
// (300a,00e8) IS [37] # 2,1 Compensator Columns
gdcmm::Attribute<0x300a,0x00e8> at2;
const gdcmm::DataElement &compensatorcols = nestedds2.
    GetDataElement( at2.GetTag() );

```

```

at2.SetFromDataElement( compensatorcols );
std::cout << at2.GetValue() << std::endl;

    // (300a,00e9) DS [3.679991\4.249288 ] # 18,2 Compensator Pixel Spacing
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);

#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\40.
    2\37.4\43.0\37.1\44.7\36] # 1934,2-2n Block Data
    (fffe,e00d)
    (fffe,e0dd)

*/
gdcM::Tag tblocksq(0x300a,0x03a6);
if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcM::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << blocksq << std::endl;
gdcM::SmartPointer<gdcM::SequenceOfItems> sssqi = blocksq.
    GetValueAsSQ();
const gdcM::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcM::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;
}

```

## 29.63 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 "gdcmReader.h"
#include "gdcmAttribute.h"

/*
This example is just for fun. We found a fake RT Ion Plan Storage and simply extracted the viz stuff for
VTK
but this is rather a RT Plan storage
*/
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " filename.dcm outfile.vti\n";
        return 1;
    }
    const char * filename = argv[1];
    const char * outfilename = argv[2];

    gdcm::Reader reader;

```

```

reader.SetFileName( filename );
if( !reader.Read() )
{
    return 1;
}

gdcmm::MediaStorage ms;
ms.SetFromFile( reader.GetFile() );
if( ms != gdcmm::MediaStorage::RTIonPlanStorage )
{
    return 1;
}

/*
(300a,00b0) SQ                                     # u/1,1 Beam Sequence
(fffe,e000) na (Item with undefined length)
(300a,00b2) SH (no value)                           # 0,1 Treatment Machine Name
(300a,00c0) IS [1 ]                                  # 2,1 Beam Number
(300a,00c2) LO [1 ]                                  # 2,1 Beam Name
(300a,00c4) CS [STATIC]                             # 6,1 Beam Type
(300a,00c6) CS [PROTON]                             # 6,1 Radiation Type
(300a,00ce) CS [TREATMENT ]                         # 10,1 Treatment Delivery Type
(300a,00e0) IS [1 ]                                 # 2,1 Number of Compensators
(300a,00e3) SQ                                     # u/1,1 Compensator Sequence
(fffe,e000) na (Item with undefined length)
(300a,00e1) SH [lucite]                             # 6,1 Material ID
(300a,00e4) IS [1 ]                                 # 2,1 Compensator Number
(300a,00e5) SH [75hdhe5 ]                          # 8,1 Compensator ID
(300a,00e7) IS [35]                                 # 2,1 Compensator Rows
(300a,00e8) IS [37]                                 # 2,1 Compensator Columns
(300a,00e9) DS [3.679991\4.249288 ]                 # 18,2 Compensator Pixel Spacing
(300a,00ea) DS [-76.00\62.50]                       # 12,2 Compensator Position
(300a,00ec) DS
[52.13\52.13\52.13\53.18\54.04\54.04\47.11\40.06\40.06\38.79\34.87\33.28\33.28\33.28\
33.28\35.43\35.43\34.54\34.54\34.71\36.10\38.62\44.88\44.88\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
(fffe,e00d)
(fffe,e000) na (Item with undefined length)
(fffe,e00d)
(fffe,e0dd)
*/
const gdcmm::DataSet& ds = reader.GetFile().GetDataSet();
gdcmm::Tag tbeamsq(0x300a,0x00b0);
if( !ds.FindDataElement( tbeamsq ) )
{
    return 1;
}
const gdcmm::DataElement &beamsq = ds.GetDataElement( tbeamsq );
//std::cout << beamsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> sqi = beamsq.
    GetValueAsSQ();
if( !sqi || !sqi->GetNumberOfItems() )
{
    return 1;
}

//for(unsigned int pd = 0; pd < sqi->GetNumberOfItems(); ++pd)
// {
//const gdcmm::Item & item = sqi->GetItem(1); // Item start at #1
const gdcmm::Item & item = sqi->GetItem(2); // Item start at #1
const gdcmm::DataSet& nestedds = item.GetNestedDataSet();
//std::cout << nestedds << std::endl;
gdcmm::Tag tcompensatorsq(0x300a,0x00e3);
if( !nestedds.FindDataElement( tcompensatorsq ) )
{
    return 1;
}
const gdcmm::DataElement &compensatorsq = nestedds.
    GetDataElement( tcompensatorsq );
//std::cout << compensatorsq << std::endl;
gdcmm::SmartPointer<gdcmm::SequenceOfItems> ssqi = compensatorsq.
    GetValueAsSQ();
const gdcmm::Item & item2 = ssqi->GetItem(1); // Item start at #1
const gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//std::cout << nestedds2 << std::endl;
gdcmm::Tag tcompensatorthicknessdata(0x300a,0x00ec);
if( !nestedds2.FindDataElement( tcompensatorthicknessdata ) )
{
    return 1;
}

```

```

    }
    const 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 );
    //img->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\40.
    (fffe,e00d)
    (fffe,e000) na (Item with undefined length)
    (fffe,e00d)
    (fffe,e0dd)
    */
    gdcm::Tag tblocksq(0x300a,0x00f4);

```

```

if( !nestedds.FindDataElement( tblocksq ) )
{
    return 1;
}
const gdcm::DataElement &blocksq = nestedds.GetDataElement( tblocksq );
//std::cout << blocksq << std::endl;
gdcm::SmartPointer<gdcm::SequenceOfItems> sssqi = blocksq.
    GetValueAsSQ();
const gdcm::Item & item3 = sssqi->GetItem(1); // Item start at #1
const gdcm::DataSet& nestedds3 = item3.GetNestedDataSet();

gdcm::Tag tblockdata(0x300a,0x0106);
if( !nestedds3.FindDataElement( tblockdata ) )
{
    return 1;
}
const gdcm::DataElement &tblockdata = nestedds3.
    GetDataElement( tblockdata );
// std::cout << tblockdata << std::endl;
gdcm::Attribute<0x300a,0x0106> at_;
at_.SetFromDataElement( tblockdata );

vtkDoubleArray *scalars = vtkDoubleArray::New();
scalars->SetNumberOfComponents(3);

gdcm::Attribute<0x300a,0x0104> bnpts; // IS [179 ] # 4,1 Block Number of
    Points
if( !nestedds3.FindDataElement( bnpts.GetTag() ) )
{
    return 1;
}
const gdcm::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;
}

```

## 29.64 gdcmscene.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 "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)
{
#ifdef VTK_MAJOR_VERSION >= 6
    append->AddInputConnection( reader->GetOutputPort(i) );
#else
    append->AddInput( reader->GetOutput(i) );
#endif
}

vtkPolyDataWriter * writer = vtkPolyDataWriter::New();
#ifdef 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() );
#ifdef 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;
}

```

## 29.65 gdcmttexture.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"

#include "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();
#ifdef (VTK_MAJOR_VERSION >= 6)
    planeMapper->SetInputConnection(plane->GetOutputPort());
#else
    planeMapper->SetInput(plane->GetOutput());
#endif

// Actor
vtkActor* planeActor = vtkActor::New();
planeActor->SetTexture(texture);
planeActor->SetMapper(planeMapper);
planeActor->PickableOn();

// Final rendering with simple interactor:
vtkRenderer *ren = vtkRenderer::New();
vtkRenderWindow *renwin = vtkRenderWindow::New();
renwin->AddRenderer(ren);
vtkRenderWindowInteractor *iren = vtkRenderWindowInteractor::New();
iren->SetRenderWindow(renwin);
ren->AddActor(planeActor);
ren->SetBackground(0,0,0.5);

vtkAnnotatedCubeActor* cube = vtkAnnotatedCubeActor::New();
cube->SetXPlusFaceText ( "L" );
cube->SetXMinusFaceText ( "R" );
cube->SetYPlusFaceText ( "A" );
cube->SetYMinusFaceText ( "P" );
cube->SetZPlusFaceText ( "H" );
cube->SetZMinusFaceText ( "F" );

vtkAxesActor* axes2 = vtkAxesActor::New();
// simulate a left-handed coordinate system
//
vtkTransform *transform = vtkTransform::New();
transform->Identity();
//transform->RotateY(180);
reader->GetDirectionCosines()->Print(std::cout);
transform->Concatenate(reader->GetDirectionCosines());
//axes2->SetShaftTypeToCylinder();
axes2->SetUserTransform( transform );
//cube->SetUserTransform( transform ); // 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;
}

```

## 29.66 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 "vtkGDCMImageReader.h"
#include "vtkPiecewiseFunction.h"
#include "vtkColorTransferFunction.h"
#include "vtkVolume.h"
#include "vtkVolumeProperty.h"
#include "vtkVolumeTextureMapper3D.h"
#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;
}

```

## 29.67 GenAIIVR.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 gdcm::Dict;
    Dict::ConstIterator beg = dict.Begin();
    Dict::ConstIterator end = dict.End();
    Dict::ConstIterator it;
    for( it = beg; it != end; ++it)
    {
        const gdcm::Tag &t = it->first;
        const gdcm::DictEntry &de = it->second;
        const gdcm::VR &vr_de = de.GetVR();
        if( vr == vr_de && !de.GetRetired() && t.GetGroup() >= 0x8 )
        {
            return t;
        }
    }
    return gdcm::Tag(0xffff,0xffff);
}

struct rnd_gen {
    rnd_gen(char const* r = "abcdefghijklmnopqrstuvwxyz0123456789")
        : range(r), len(std::strlen(r)) { }

    char operator ()() const {
        return range[static_cast<std::size_t>(std::rand() * (1.0 / ((double)RAND_MAX + 1.0)) * (double)len)];
    }
private:
    char const* range;
    std::size_t len;
}

```

```

};

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    const char *outfilename = argv[1];
    static const gdcm::Global &g = gdcm::Global::GetInstance();
    static const gdcm::Dicts &dictionaries = g.GetDicts();
    static const gdcm::Dict &pubdict = dictionaries.GetPublicDict();
    using gdcm::VR;
    using gdcm::Tag;

    gdcm::Writer w;

    gdcm::File &f = w.GetFile();
    gdcm::DataSet &ds = f.GetDataSet();

    gdcm::FileExplicitFilter fef;
    //fef.SetChangePrivateTags( true );
    fef.SetFile( w.GetFile() );
    if( !fef.Change() )
    {
        std::cerr << "Failed to change" << std::endl;
        return 1;
    }

    gdcm::SmartPointer<gdcm::SequenceOfItems> sq = new
        gdcm::SequenceOfItems();
    sq->SetLengthToUndefined();

    // gdcm::DummyValueGenerator dv;

    const std::size_t len = 10;
    char ss[len+1];
    ss[len] = '\0';

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcm::DataElement owner( gdcm::Tag(0x4d4d, 0x10) );
    owner.SetByteValue(owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcm::VR::LO );

    // Create an item
    gdcm::Item it;
    it.SetVLToUndefined();
    gdcm::DataSet &nds = it.GetNestedDataSet();
    // nds.Insert(owner);
    // nds.Insert(de);

    // Insert sequence into data set
    gdcm::DataElement des( gdcm::Tag(0x4d4d, 0x1001) );
    des.SetVR(gdcm::VR::SQ);
    des.SetValue(*sq);
    des.SetVLToUndefined();

    ds.Insert(owner);
    ds.Insert(des);

    // avoid INVALID = 0
    for(int i = 1; i < 27; ++i)
    {
        VR vr = (VR::VRType)(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

```

```

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 );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR::UI );
gdcmm::MediaStorage ms( gdcmm::MediaStorage::RawDataStorage
);
de.SetByteValue( ms.GetString(), (uint32_t)strlen(ms.GetString()) );
ds.Insert( de );

gdcmm::FileMetaInformation &fmi = f.GetHeader();
//fmi.SetDataSetTransferSyntax( gdcmm::TransferSyntax::ImplicitVRLittleEndian );
fmi.SetDataSetTransferSyntax(
    gdcmm::TransferSyntax::ExplicitVRLittleEndian );

w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if ( !w.Write() )
{
    return 1;
}

return 0;
}

```

## 29.68 GenerateDICOMDIR.cs

This is a C# example on how to use `gdcmm::DICOMDIRGenerator`

```

/*=====
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.

=====*/

/*
 * Simple C# example to show how to use DICOMDIRGenerator
 *
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcmm/debug-gcc/bin
 * $ mono bin/GenerateDICOMDIR.exe path output_filename
 */
using System;
using gdcmm;

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();

        gdcmm.DICOMDIRGenerator gen = new DICOMDIRGenerator();

```

```

    gen.SetFilenames( filenames );
    gen.SetDescriptor( descriptor );
    if( !gen.Generate() )
    {
        return 1;
    }

    gdcmm.FileMetaInformation.SetSourceApplicationEntityTitle( "GenerateDICOMDIR" );
    gdcmm.Writer writer = new Writer();
    writer.SetFile( gen.GetFile() );
    writer.SetFileName( outfilename );
    if( !writer.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 29.69 GenerateRTSTRUCT.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 "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 "gdcmmDirectoryHelper.h"

using namespace gdcmm;

//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::FileNamesType theRTSeries =
        DirectoryHelper::GetRTStructSeriesUIDs(theDirName);

    gdcm::Directory theDir;
    theDir.Load(argv[1]);

    if (theRTSeries.empty())
    {
        std::cerr << "No RTStructs found for the test, ending." << std::endl;
        return 1;
    }

    for (size_t q = 0; q < theRTSeries.size(); q++)
    {
        Directory::FileNamesType theRTNames =
            DirectoryHelper::GetFileNamesFromSeriesUIDs(theDirName,
                theRTSeries[q]);

        if (theRTNames.empty()){
            std::cerr << "Unable to load RT Series " << theRTSeries[q] << ", continuing. " << std::endl;
            continue;
        }

        vtkGDCMPolyDataReader * reader =
            vtkGDCMPolyDataReader::New();
        reader->SetFileName( theRTNames[0].c_str() );
        reader->Update();

        //std::cout << reader->GetMedicalImageProperties()->GetStudyDate() << std::endl;

        vtkGDCMPolyDataWriter * writer =
            vtkGDCMPolyDataWriter::New();
        int numMasks = reader->GetNumberOfOutputPorts() + 1; //add a blank one in
        writer->SetNumberOfInputPorts( numMasks );
        std::string thePotentialName = theDirName + "/" + "GDCMTestRTStruct." + theRTSeries[q] + ".dcm";
        gdcm::Directory::FileNamesType theFileNames = theDir.
            GetFileNames();
        //keep renaming the output until we get something that doesn't overwrite what was there already
        int count = 0;
        while (std::find(theFileNames.begin(), theFileNames.end(), thePotentialName) != theFileNames.end())
        {
            char buff[255];
            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;

```



```

}
```

## 29.70 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;
}

```

## 29.71 GenFakeIdentifyFile.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
        gdcmm::Item it;
        it.SetVLToUndefined();
        gdcmm::DataSet &nds = it.GetNestedDataSet();
        // Insert sequence into data set
        assert(de.GetVR() == gdcmm::VR::SQ );
        gdcmm::SmartPointer<gdcmm::SequenceOfItems> sq = new
            gdcmm::SequenceOfItems();
        sq->SetLengthToUndefined();
        de.SetValue(*sq);
        de.SetVLToUndefined();
        //ds.Insert( de);

        if( !toremove )
        {
            nds.Insert( CreateFakeElement( balcptags[count], true ) );
            countglobal++;
        }
        else
        {
            gdcmm::Attribute<0x0008,0x0000> at1 = { 0 }; // This element has no
                reason to be 'anonymized'...
            nds.Insert( at1.GetAsDataElement() );
            gdcmm::Attribute<0x000a,0x0000> at2 = { 0 };
            nds.Insert( at2.GetAsDataElement() );
        }
        sq->AddItem(it);
    }
    return de;
}

/*
*/
int main(int argc, char *argv[])
{
    if( argc < 2 )
    {
        std::cerr << argv[0] << " output.dcm" << std::endl;
        return 1;
    }
    using gdcmm::Tag;
    using gdcmm::VR;
    const char *outfilename = argv[1];

    std::vector<gdcmm::Tag> balcptags =
        gdcmm::Anonymizer::GetBasicApplicationLevelConfidentialityProfileAttributes
            ();

    gdcmm::Writer w;
    gdcmm::File &f = w.GetFile();
    gdcmm::DataSet &ds = f.GetDataSet();

    // Add attribute that need to be anonymized:
    std::vector<gdcmm::Tag>::const_iterator it = balcptags.begin();
    for(; it != balcptags.end(); ++it)
    {
        ds.Insert( CreateFakeElement( *it, true ) );
    }

    // Add attribute that do NOT need to be anonymized:
    static const gdcmm::Global &g = gdcmm::Global::GetInstance();
    static const gdcmm::Dicts &dicts = g.GetDicts();
    static const gdcmm::Dict &pubdict = dicts.GetPublicDict();

    using gdcmm::Dict;
    Dict::ConstIterator dictit = pubdict.Begin();
    for(; dictit != pubdict.End(); ++dictit)
    {
        const gdcmm::Tag &dicttag = dictit->first;
        if( dicttag == Tag(0x6e65,0x6146) ) break;
        //const gdcmm::DictEntry &dictentry = dictit->second;
        ds.Insert( CreateFakeElement( dicttag, false ) );
    }
    ds.Remove( gdcmm::Tag(0x400,0x500) );
    ds.Remove( gdcmm::Tag(0x12,0x62) );
    ds.Remove( gdcmm::Tag(0x12,0x63) );

```

```

// Make sure to override any UID stuff
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( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

## 29.72 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;
}

```

## 29.73 GenLongSeqs.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 "gdcmlItem.h"
#include "gdcmlImageReader.h"
#include "gdcmlSequenceOfItems.h"
#include "gdcmlFile.h"
#include "gdcmlTag.h"

/*
 * This example is used to generate the file:
 *
 *
 * There is a flaw in the DICOM design where it is assumed that Sequence can be
 * either represented as undefined length or defined length. This should work
 * in most case, but the undefined length is a little more general and can
 * store sequence of items that a defined length cannot.
 * We need to make sure that we can store numerous Item in a SQ
 *
 * Warning: do not try to compute the group length elements !
 * Warning: You may need a 64bits machine for this example to work.
 */
int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];
    gdcml::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        return 1;
    }

    gdcml::File &file = reader.GetFile();
    gdcml::DataSet &ds = file.GetDataSet();

    // Create a Sequence
    gdcml::SmartPointer<gdcml::SequenceOfItems> sq = new
        gdcml::SequenceOfItems();
    sq->SetLengthToUndefined();

    const char owner_str[] = "GDCM CONFORMANCE TESTS";
    gdcml::DataElement owner( gdcml::Tag(0x4d4d, 0x10) );
    owner.SetByteValue( owner_str, (uint32_t)strlen(owner_str));
    owner.SetVR( gdcml::VR::LO );

    size_t nitems = 1000;
    nitems += std::numeric_limits<uint32_t>::max();
    for( unsigned int idx = 0; idx < nitems; ++idx)
    {
        // Create a dataelement
        //gdcml::DataElement de( gdcml::Tag(0x4d4d, 0x1002) );
        //de.SetByteValue(ptr, ptr_len);

```

```

    //de.SetVR( gdcmm::VR::OB );

    // Create an item
    gdcmm::Item it;
    it.SetVLToUndefined();
    //gdcmm::DataSet &nds = it.GetNestedDataSet();
    //nds.Insert(owner);
    //nds.Insert(de);

    sq->AddItem(it);
}

// Insert sequence into data set
gdcmm::DataElement des( gdcmm::Tag(0x4d4d,0x1001) );
des.SetVR(gdcmm::VR::SQ);
des.SetValue(*sq);
des.SetVLToUndefined();

ds.Insert(owner);
ds.Insert(des);

gdcmm::Writer w;
w.SetFile( file );
//w.SetCheckFileMetaInformation( true );
w.SetFileName( outfilename );
if (!w.Write() )
{
    return 1;
}

return 0;
}

```

## 29.74 GenSeqs.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmmReader.h"
#include "gdcmmWriter.h"
#include "gdcmmItem.h"
#include "gdcmmImageReader.h"
#include "gdcmmSequenceOfItems.h"
#include "gdcmmFile.h"
#include "gdcmmTag.h"

/*
 * This example is used to generate the file:
 *
 * gdcmmConformanceTests/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 case, 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( outfile );
    if ( !w.Write() )
    {
        return 1;
    }

    return 0;
}

```

## 29.75 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;
    }
}

```

## 29.76 GetJPEGSamplePrecision.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example is a little helper to detect the famous SIEMENS JPEG lossless compressed image
 * where DICOM is declared as:
 *
 * (0028,0100) US 16 # 2,1 Bits Allocated
 * (0028,0101) US 12 # 2,1 Bits Stored
 * (0028,0102) US 11 # 2,1 High Bit
 * (0028,0103) US 0 # 2,1 Pixel Representation
 *
 * But where JPEG is:
 *
 * JPEG_SOF_Parameters:
 * SamplePrecision = 16
 * nLines = 192
 * nSamplesPerLine = 192
 * nComponentsInFrame = 1
 * component 0
 * ComponentIdentifier = 1
 * HorizontalSamplingFactor = 1
 * VerticalSamplingFactor = 1
 * QuantizationTableDestinationSelector = 0
 *
 * This case is valid. One simply has to use the 16bits jpeg decoder to decode the 12bits stored image.
 * This used to be an issue in GDCM 1.2.x (fixed in GDCM 1.2.5)
 *
 * The main return 0 (no error) when the file read is actually a potential problem. At the end of the main
 * function, the jpeg stream is stored in the filename specified as second argument
 */

#include "gdcmImageReader.h"
#include "gdcmSequenceOfFragments.h"
#include "gdcmJPEGCodec.h"

#include <iostream>
#include <fstream>

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.jpg" << std::endl;
        return 1;
    }
    const char *filename = argv[1];
    const char *outfilename = argv[2];

    gdcm::ImageReader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    const gdcm::File &file = reader.GetFile();
    const gdcm::Image &image = reader.GetImage();

    const gdcm::TransferSyntax &ts = file.GetHeader().
        GetDataSetTransferSyntax();

    if( ts != gdcm::TransferSyntax::JPEGLosslessProcess14 && ts !=
        gdcm::TransferSyntax::JPEGLossLessProcess14_1 )
    {

```

```

    std::cerr << "Input is not a lossless JPEG" << std::endl;
    return 1;
}

// the dataset is the the set of element we are interested in:
const gdcm::DataSet &ds = file.GetDataSet();

const gdcm::Tag rawTag(0x7fe0, 0x0010); // Default to Pixel Data
const gdcm::DataElement& pdde = ds.GetDataElement( rawTag );
const gdcm::SequenceOfFragments *sf = pdde.
    GetSequenceOfFragments();
if( sf )
{
    std::ofstream output(outfilename, std::ios::binary);
    sf->WriteBuffer(output);
}
else
{
    std::cerr << "Error" << std::endl;
    return 1;
}

gdcm::JPEGCodec jpeg;
std::ifstream is(outfilename, std::ios::binary);
gdcm::PixelFormat pf ( gdcm::PixelFormat::UINT8 ); // let's
    pretend it's a 8bits jpeg
jpeg.SetPixelFormat( pf );
gdcm::TransferSyntax ts_jpg;
bool b = jpeg.GetHeaderInfo( is, ts_jpg );
if( !b )
{
    return 1;
}

//jpeg.Print( std::cout );
if( jpeg.GetPixelFormat().GetBitsAllocated() != image.
    GetPixelFormat().GetBitsAllocated()
|| jpeg.GetPixelFormat().GetBitsStored() != image.
    GetPixelFormat().GetBitsStored() )
{
    std::cerr << "There is a mismatch in between DICOM declared Pixel Format and Sample Precision used in
        the JPEG stream" << std::endl;
    return 0;
}

std::cout << jpeg.GetPixelFormat() << std::endl;
std::cout << image.GetPixelFormat() << std::endl;

return 1;
}

```

## 29.77 GetPortionCSAHeader.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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
66     bv = gdt.GetByteValue();
67     #print bv
68     str = bv.GetPointer()
69     print str.split("\\")

```

## 29.78 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;

```

```

gdcmm::Attribute<0x0018,0x601c> atX1;
gdcmm::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;
}

```

## 29.79 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.GetDataSet();
    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(0x7fel,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(0x7fel,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(0x7fel,0x37,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq4(0x7fel,0x43,"GEMS_Ultrasound_MovieGroup_001");
const PrivateTag tseq5(0x7fel,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;
#ifdef 0
    {
        std::ofstream out( "/tmp/mo4", std::ios::binary );
        out.write( bv4->GetPointer(), bv4->GetLength());
        out.close();
    }
#endif
    const ByteValue *bv5 = seq5.GetByteValue();
#ifdef 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( gdcm::PhotometricInterpretation::MONOCHROME2
    );

im->SetDataElement( fakedata );

gdcm::ImageWriter w;
w.SetImage( *im );
DataSet &dataset = w.GetFile().GetDataSet();

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 );
dataset.Replace( de );

de.SetTag( Tag(0x8,0x16) ); // SOP Class UID
de.SetVR( VR:UI );
gdcm::MediaStorage ms(
    gdcm::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;
}

```



## 29.80 headsq2dcm.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17 python headsq2dcm.py -D /path/to/VTKData
18 """
19
20 import vtk
21 import vtkgdcm
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/headsq/quarter")
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 = vtkgdcm.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()

```

## 29.81 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(vtkgdc.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        Kitware.VTK.vtkImageData imgout = new Kitware.VTK.vtkImageData( rawCppThis.Handle, false, false);
        return imgout;
    }

    static vtkgdc.vtkImageData ConnectActivizToSWIG(Kitware.VTK.vtkImageData imgin)
    {
        HandleRef rawCppThis = imgin.GetCppThis();
        vtkgdc.vtkImageData imgout = new vtkgdc.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

        vtkgdc.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;
}
}

```

## 29.82 HelloActiviz2.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * Usage:
 * $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
 * $ mono ./bin/HelloActiviz2.exe gdcmlData/test.acr bla.png bla2.dcm
 */

/*
 * From the outside view, no-one can detect that object pass to/from
 * vtkGDCMImageWriter/vtkGDCMImageReader are not Activiz object.
 *
 * TODO: Test Command/Observer
 */
public class HelloActiviz2
{
    public static int Main(string[] args)
    {
        string filename = args[0];
        string outfilename = args[1];
        string outfilename2 = args[2];

        vtkGDCMImageReader reader = new Kitware.VTK.GDCM.vtkGDCMImageReader();
        reader.SetFileName( filename );

        // When calling multiple times creation of C# object from the same C++ object it triggers a:
        //error: potential refcounting error: Duplicate rawCppThis - weak reference that is still alive. Attempting
        //to add '0x00b2dc10' again.
        //    Allowing new wrapped object to take over table key...
        //    Original object should *not* have been destroyed while we still had it in our table without
        //    notifying us...
        //reader.GetOutput();
        //reader.GetOutput();

        System.Console.WriteLine( reader.ToString() ); // Test the ToString compat with Activiz

        vtkGDCMImageWriter writer = new vtkGDCMImageWriter();
        writer.SetInput( reader.GetOutput() );
        writer.SetFileName( outfilename2 );
        writer.Write();

        System.Console.WriteLine( reader.GetOutput().ToString() ); // Test the ToString compat with Activiz

        System.Console.WriteLine( writer.ToString() ); // Test the ToString compat with Activiz

        vtkPNGWriter pngwriter = new vtkPNGWriter();
        pngwriter.SetInput( reader.GetOutput() );
        pngwriter.SetFileName( outfilename );
        pngwriter.Write();

        // at that point the .Write() should have triggered an Update() on the reader:
        if( reader.GetImageFormat() == vtkgdcml.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;
}

```

## 29.83 HelloActiviz3.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/HelloActiviz3.exe ~/Creatis/gdcmData/test.acr
 */
public class HelloActiviz3
{
    public static int Main(string[] args)
    {
        string filename = args[0];

        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);

        reader.SetFileNames(array);
        reader.Update();

        //System.Console.Write(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkImageViewer2 viewer = vtkImageViewer2.New();
        viewer.SetInput(reader.GetOutput());
        viewer.SetupInteractor(iren);
        viewer.SetSize(600, 600);
        viewer.Render();

        iren.Initialize();
        iren.Start();

        return 0;
    }
}

```

## 29.84 HelloActiviz4.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

/*
 * $ 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;
    }
}

```

## 29.85 HelloActiviz5.cs

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcms.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
using Kitware.VTK;
using Kitware.VTK.GDCM;

// The command line arguments are:
// -I      => run in interactive mode; unless this is used, the program will
//          not allow interaction and exit
// -D <path> => path to the data; the data should be in <path>/Data/

/*

```

```

* $ export MONO_PATH=/usr/lib/cli/Activiz.NET:/usr/lib/cli/Kitware.mummy.Runtime-1.0
* $ mono ./bin/HelloActiviz5.exe -I
*/
public class HelloActiviz5
{
    public static int Main(string[] args)
    {
        vtkTesting testHelper = vtkTesting.New();
        for ( int cc = 0; cc < args.Length; cc++ )
        {
            //testHelper.AddArguments(argc,const_cast<const char **>(argv));
            //System.Console.Write( "args: " + args[cc] + "\n" );
            testHelper.AddArgument( args[cc] );
        }
        if ( testHelper.IsFlagSpecified("-D") != 0 )
        {
            string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();
            if( VTK_DATA_ROOT != null )
            {
                //System.Console.Write( "VTK_DATA_ROOT: " + VTK_DATA_ROOT + "\n" );
                testHelper.SetDataRoot(VTK_DATA_ROOT);
                testHelper.AddArgument("-D");
                testHelper.AddArgument(VTK_DATA_ROOT);
            }
        }

        string dataRoot = testHelper.GetDataRoot();
        string filename = dataRoot;
        filename += "/Data/mr.001";

        vtkDirectory dir = vtkDirectory.New();
        if( dir.FileIsDirectory( dataRoot ) == 0 )
        {
            filename = vtkGDCMTesting.GetGDCMDataRoot() + "/test.acr";
        }
        //System.Console.Write( "dataRoot: " + dataRoot + "\n" );
        System.Console.Write( "filename being used is: " + filename + "\n" );

        vtkGDCMImageReader reader = vtkGDCMImageReader.New();
        vtkStringArray array = vtkStringArray.New();
        array.InsertNextValue(filename);
        reader.SetFileNames(array);
        reader.Update();

        System.Console.Write(reader.GetOutput());

        vtkRenderWindowInteractor iren = vtkRenderWindowInteractor.New();

        vtkRenderer ren1 = vtkRenderer.New();
        vtkRenderWindow renWin = vtkRenderWindow.New();
        renWin.AddRenderer(ren1);

        vtkImageActor actor = vtkImageActor.New();

        vtkImageMapToWindowLevelColors coronalColors = vtkImageMapToWindowLevelColors.New();
        coronalColors.SetInput(reader.GetOutput());

        actor.SetInput(coronalColors.GetOutput());

        ren1.AddActor(actor);
        iren.SetRenderWindow(renWin);

        iren.Initialize();

        renWin.Render();

        int retVal = testHelper.IsInteractiveModeSpecified();

        if( retVal != 0 )
        {
            iren.Start();
        }

        return 0;
    }
}

```

## 29.86 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 );
    }
}

```

## 29.87 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( outfile );
//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: " << outfile << std::endl;
    return 1;
}

return 0;
}

```

## 29.88 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() == vtkgdcml.VTK_LUMINANCE ) // MONOCHROME2
        {
            System.Console.WriteLine( "Image is MONOCHROME2" ); //
        }

        // Just for fun, invert the direction cosines, output should reflect that:
        vtkMatrix4x4 dircos = reader.GetDirectionCosines();
        dircos.Invert();

        string outfilename = args[1];
        vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
        writer.SetMedicalImageProperties( reader.GetMedicalImageProperties() );
        writer.SetDirectionCosines( dircos );
        writer.SetShift( reader.GetShift() );
        writer.SetScale( reader.GetScale() );
        writer.SetImageFormat( reader.GetImageFormat() );
        writer.SetFileName( outfilename );
        //writer.SetInputConnection( reader.GetOutputPort() ); // new
        writer.SetInput( reader.GetOutput() ); // old
        writer.Write();

        return 0;
    }
}

```

## 29.89 HelloVTKWorld.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 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:gdcm.jar:. java HelloVTKWorld gdcmData/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() == vtkgdcmtk.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 );
}
}

```

## 29.90 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 vtkgdcmtk;

```

```

/*
 * This test only test the SWIG/VTK part, you do not need Activiz
 */
public class HelloWorld2
{
    public static int Main(string[] args)
    {
        string VTK_DATA_ROOT = vtkGDCMTesting.GetVTKDataRoot();

        vtkVolume16Reader reader = vtkVolume16Reader.New();
        reader.SetDataDimensions(64, 64);
        reader.SetDataByteOrderToLittleEndian();
        reader.SetFilePrefix(VTK_DATA_ROOT + "/Data/headsq/quarter");
        reader.SetImageRange(1, 93);
        reader.SetDataSpacing(3.2, 3.2, 1.5);

        vtkImageCast cast = vtkImageCast.New();
        cast.SetInput( reader.GetOutput() );
        cast.SetOutputScalarTypeToUnsignedChar();

        // By default this is creating a Multiframe Grayscale Word Secondary Capture Image Storage
        vtkGDCMImageWriter writer = vtkGDCMImageWriter.New();
        writer.SetFileName( "headsq.dcm" );
        writer.SetInput( reader.GetOutput() );
        // cast -> Multiframe Grayscale Byte Secondary Capture Image Storage
        // writer.SetInput( cast.GetOutput() );
        writer.SetFileDimensionality( 3 );
        writer.Write();

        return 0;
    }
}

```

## 29.91 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( outfilename );
writer.SetFile( file );
if( !writer.Write() )
{
    std::cerr << "Could not write: " << outfilename << std::endl;
    return 1;
}

return 0;
}

/*
 * (*) static type, means that extra DICOM information VR & VM are computed at compilation time.
 * The compiler is deducing those values from the template arguments of the class.
 */

```

## 29.92 HelloWorld.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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     # Instanciate 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))

```

## 29.93 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;
}

```

## 29.94 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

```

```

gdcmm::DataSet& nestedds2 = item2.GetNestedDataSet();
//item2.SetVLTToUndefined();
//std::cout << nestedds2 << std::endl;
// (3006,0050) DS [43.57636\65.52504\10.0\46.043102\62.564945\10.0\49.126537\60.714... # 398,48
    ContourData
gdcmm::Tag tcontourdata(0x3006,0x0050);
const gdcmm::DataElement & contourdata = nestedds2.
    GetDataElement( tcontourdata );
//std::cout << contourdata << std::endl;

//const gdcmm::ByteValue *bv = contourdata.GetByteValue();
gdcmm::Attribute<0x3006,0x0046> ncontourpoints;
ncontourpoints.Set( nestedds2 );

gdcmm::Attribute<0x3006,0x0050> at;
at.SetFromDataElement( contourdata );
const double* pts = at.GetValues();
unsigned int npts = at.GetNumberOfValues() / 3;

std::vector<double> out( pts, pts + npts * 3 );
std::vector<double> out2;

//const unsigned int niter = 7;
const unsigned int niter = 8;
for( unsigned int i = 0; i < niter; ++i)
{
    //bool b =
    interpolate(&out[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 );
nestedds2.Replace( at_interpolate.GetAsDataElement() );
nestedds2.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;
}

```

## 29.95 MagnifyFile.cxx

```

/*=====

```



```

Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "vtkGDCMImageReader.h"
#include "vtkGDCMImageWriter.h"
#include "vtkImageData.h"
#include "vtkImageMagnify.h"
#include "vtkImageCast.h"

#include "gdcmTesting.h"
#include "gdcmSystem.h"

// This is a simple test to magnify an image that is known to give excellent
// compression ratio. This will be our test for those large image
int main(int, char *[])
{
    const char *directory = gdcm::Testing::GetDataRoot();
    if(!directory) return 1;
    std::string file = std::string(directory) + "/test.acr";
    std::cout << file << std::endl;
    if( !gdcm::System::FileExists( file.c_str() ) ) return 1;

    vtkGDCMImageReader *reader = vtkGDCMImageReader::New();
    reader->SetFileName( file.c_str() );
    reader->Update();
    //reader->GetOutput()->Print( std::cout );

    vtkImageCast *cast = vtkImageCast::New();
    #if (VTK_MAJOR_VERSION >= 6)
        cast->SetInputConnection( reader->GetOutputPort() );
    #else
        cast->SetInput( reader->GetOutput() );
    #endif
    cast->SetOutputScalarTypeToUnsignedShort();

    vtkImageMagnify *magnify = vtkImageMagnify::New();
    #if (VTK_MAJOR_VERSION >= 6)
        magnify->SetInputConnection( cast->GetOutputPort() );
    #else
        magnify->SetInput( cast->GetOutput() );
    #endif
    magnify->SetInterpolate( 1 );
    magnify->SetInterpolate( 0 );
    int factor = 100;
    magnify->SetMagnificationFactors (factor, factor, 1);

    vtkGDCMImageWriter *writer = vtkGDCMImageWriter::New();
    writer->SetFileName( "/tmp/bla.dcm" );
    #if (VTK_MAJOR_VERSION >= 6)
        writer->SetInputConnection( magnify->GetOutputPort() );
    #else
        writer->SetInput( magnify->GetOutput() );
    #endif
    writer->SetImageFormat( reader->GetImageFormat() );
    writer->SetMedicalImageProperties( reader->GetMedicalImageProperties() );
    writer->SetDirectionCosines( reader->GetDirectionCosines() );
    writer->SetShift( reader->GetShift() );
    writer->SetScale( reader->GetScale() );
    writer->Write();

    // TODO:
    //vtkImageAppendComponents.h

    reader->Delete();
    magnify->Delete();
    writer->Delete();

    return 0;
}

```

## 29.96 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;
    }
}

```

## 29.97 ManipulateFile.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

```

```

11 #     PURPOSE.  See the above copyright notice for more information.
12 #
13 #####
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     file1 = sys.argv[1]
33     file2 = sys.argv[2]
34
35     r = gdcm.Reader()
36     r.SetFileName( file1 )
37     if not r.Read():
38         sys.exit(1)
39
40     ano = gdcm.Anonymizer()
41     ano.SetFile( r.GetFile() )
42     ano.RemovePrivateTags()
43     ano.Remove( gdcm.Tag(0x0032,0x1030) )
44     ano.Remove( gdcm.Tag(0x008,0x14) )
45     ano.Remove( gdcm.Tag(0x008,0x1111) )
46     ano.Remove( gdcm.Tag(0x008,0x1120) )
47     ano.Remove( gdcm.Tag(0x008,0x1140) )
48     ano.Remove( gdcm.Tag(0x10,0x21b0) )
49     ano.Empty( gdcm.Tag(0x10,0x10) )
50     ano.Empty( gdcm.Tag(0x10,0x20) )
51     ano.Empty( gdcm.Tag(0x10,0x30) )
52     ano.Empty( gdcm.Tag(0x20,0x10) )
53     ano.Empty( gdcm.Tag(0x32,0x1032) )
54     ano.Empty( gdcm.Tag(0x32,0x1033) )
55     ano.Empty( gdcm.Tag(0x40,0x241) )
56     ano.Empty( gdcm.Tag(0x40,0x254) )
57     ano.Empty( gdcm.Tag(0x40,0x253) )
58     ano.Empty( gdcm.Tag(0x40,0x1001) )
59     ano.Empty( gdcm.Tag(0x8,0x80) )
60     ano.Empty( gdcm.Tag(0x8,0x50) )
61     ano.Empty( gdcm.Tag(0x8,0x1030) )
62     ano.Empty( gdcm.Tag(0x8,0x103e) )
63     ano.Empty( gdcm.Tag(0x18,0x1030) )
64     ano.Empty( gdcm.Tag(0x38,0x300) )
65     g = gdcm.UIDGenerator()
66     ano.Replace( gdcm.Tag(0x0008,0x0018), g.Generate() )
67     ano.Replace( gdcm.Tag(0x0020,0x00d), g.Generate() )
68     ano.Replace( gdcm.Tag(0x0020,0x00e), g.Generate() )
69     ano.Replace( gdcm.Tag(0x0020,0x052), g.Generate() )
70     #ano.Replace( gdcm.Tag(0x0008,0x0016), "1.2.840.10008.5.1.4.1.1.7.2" )
71     """
72     ano.Remove( gdcm.Tag(0x0018,0x0020) ) # ScanningSequence
73     ano.Remove( gdcm.Tag(0x0018,0x0021) ) # SequenceVariant
74     ano.Remove( gdcm.Tag(0x0018,0x0022) ) # ScanOptions
75     ano.Remove( gdcm.Tag(0x0018,0x0023) ) # MRAcquisitionType
76     ano.Remove( gdcm.Tag(0x0018,0x0050) ) # SliceThickness
77     ano.Remove( gdcm.Tag(0x0018,0x0080) ) # RepetitionTime
78     ano.Remove( gdcm.Tag(0x0018,0x0081) ) # EchoTime
79     ano.Remove( gdcm.Tag(0x0018,0x0088) ) # SpacingBetweenSlices
80     ano.Remove( gdcm.Tag(0x0018,0x0091) ) # EchoTrainLength
81     ano.Remove( gdcm.Tag(0x0018,0x1164) ) # ImagerPixelSpacing
82
83     ano.Remove( gdcm.Tag(0x0020,0x0032) ) # Image Position (Patient)
84     ano.Remove( gdcm.Tag(0x0020,0x0037) ) # Image Orientation (Patient)
85     ano.Remove( gdcm.Tag(0x0020,0x0052) ) # Frame of Reference UID
86     ano.Remove( gdcm.Tag(0x0020,0x1040) ) # Position Reference Indicator
87
88     ano.Replace( gdcm.Tag(0x0028,0x0301), "NO" ) # Burned In Annotation
89
90     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)

```

## 29.98 ManipulateSequence.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcM.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 = gdcmm.Tag(0x0040,0x0170) # 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 = gdcmm.Tag(0x0008,0x0104)
64             if nestedds2.FindDataElement( tcm ):
65                 cm = nestedds2.GetDataElement( tcm )
66                 mystr = "GDCM was here"
67                 cm.SetByteValue( mystr, gdcmm.VL( len(mystr) ) )
68
69     w = gdcmm.Writer()
70     w.SetFile( f )
71     w.SetFileName( file2 )
72     if not w.Write():
73         sys.exit(1)

```

## 29.99 MergeFile.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 gdcmm
28
29 if __name__ == "__main__":
30
31     file1 = sys.argv[1]
32     file2 = sys.argv[2]
33
34     r1 = gdcmm.ImageReader()
35     r1.SetFileName( file1 )
36     if not r1.Read():
37         sys.exit(1)
38
39     r2 = gdcmm.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 = gdcmm.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)

```

## 29.100 MergeTwoFiles.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcml.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/
/*
 * This example 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 gdcmlData/012345.002.050.dcm gdcmlData/test.acr merge.dcm
 */

#include "gdcmlReader.h"
#include "gdcmlImageReader.h"
#include "gdcmlImageWriter.h"
#include "gdcmlWriter.h"
#include "gdcmlDataSet.h"
#include "gdcmlAttribute.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
    gdcml::ImageReader reader1;
    reader1.SetFileName( file1 );
    if( !reader1.Read() )
    {
        return 1;
    }

    // Read file2
    gdcml::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() ==
        gdcml::TransferSyntax::ImplicitVRLittleEndian )
    {
        reader2.GetImage().SetTransferSyntax(
            gdcml::TransferSyntax::ImplicitVRLittleEndian );
    }

    gdcml::ImageWriter writer;
    writer.SetFileName( file3 );

```

```

writer.SetFile( reader1.GetFile() );
// ImageWriter will always use all of gdcm::Image information an override anything wrong from
// reader1.GetFile(), including the Transfer Syntax
writer.SetImage( reader2.GetImage() );

gdcm::DataSet &ds = reader1.GetFile().GetDataSet();

// Make sure that SOPInstanceUID are different
// Simply removing it is sufficient as gdcm::ImageWriter will generate one by default
// if not found.
ds.Remove( gdcm::Tag(0x0008,0x0018) );
if( !writer.Write() )
{
    return 1;
}

return 0;
}

```

## 29.101 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;
}
}

```

## 29.102 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], ipzspacing );

    // 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.SetInput( change.GetOutput() );

    //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 );

```

```

    renl.ResetCamera();

    iren.SetRenderWindow( renWin );

    // interact with data
    renWin.Render();

    iren.Start();
}
}

```

## 29.103 MpegVideoInfo.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.

=====*/
/*
 * 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/mpegghdrs.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/mpegghdrs.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 gdcml;

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;
    #endregion
}

```

```

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;
}

```

## 29.104 MPRViewer.java

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
import vtk.*;
import gdcm.*;
import java.io.File;

/*
 * Compilation:
 * CLASSPATH=vtkgdcm.jar:/usr/share/java/vtk.jar javac MPRViewer.java
 *
 * Usage:
 * LD_LIBRARY_PATH=/usr/lib/jvm/java-6-openjdk/jre/lib/amd64/xawt:/usr/lib/jni:. CLASSPATH=/usr/share/java/
 *   vtk.jar:vtkgdcm.jar:gdcm.jar:. java MPRViewer BRAINX
 */
public class MPRViewer
{
    static {
        // VTK
        System.loadLibrary("vtkCommonJava");
    }
}

```

```

    System.loadLibrary("vtkFilteringJava");
    System.loadLibrary("vtkIOJava");
    System.loadLibrary("vtkImagingJava");
    System.loadLibrary("vtkGraphicsJava");
    System.loadLibrary("vtkRenderingJava");
    // VTK-GDCM
    System.loadLibrary("vtkgdcmJava");
}

static FilenamesType fns = new FilenamesType();

public static void process(String path)
{
    fns.add( path );
}

// Process only files under dir
public static void visitAllFiles(File dir)
{
    if (dir.isDirectory())
    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public static void main(String[] args) throws Exception
{
    String dirname = args[0];
    if( !PosixEmulation.FileIsDirectory( dirname ) )
    {
        return;
    }

    File dir = new File(dirname);
    visitAllFiles(dir);

    IPPSorter ipp = new IPPSorter();
    ipp.SetComputeZSpacing( true );
    ipp.SetZSpacingTolerance( 1e-3 );
    boolean b = ipp.Sort( fns );
    if(!b)
    {
        throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );

    // A simple vtkInteractorStyleImage example for
    // 3D image viewing with the vtkImageResliceMapper.
    //
    // Drag Left mouse button to window/level
    // Shift-Left drag to rotate (oblique slice)
    // Shift-Middle drag to slice through image
    // OR Ctrl-Right drag to slice through image

    // Create the RenderWindow, Renderer

```



```

vtkRenderer ren1 = new vtkRenderer();
vtkRenderWindow renWin = new vtkRenderWindow();
renWin.AddRenderer(ren1);

vtkImageResliceMapper im = new vtkImageResliceMapper();
im.SetInputConnection(change.GetOutputPort());
im.SliceFacesCameraOn();
im.SliceAtFocalPointOn();
im.BorderOff();

vtkImageProperty ip = new vtkImageProperty();
ip.SetColorWindow(2000);
ip.SetColorLevel(1000);
ip.SetAmbient(0.0);
ip.SetDiffuse(1.0);
ip.SetOpacity(1.0);
ip.SetInterpolationTypeToLinear();

vtkImageSlice ia = new vtkImageSlice();
ia.SetMapper(im);
ia.SetProperty(ip);

ren1.AddViewProp(ia);
ren1.SetBackground(0.1,0.2,0.4);
renWin.SetSize(300,300);

vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
vtkInteractorStyleImage style = new vtkInteractorStyleImage();
style.SetInteractionModeToImage3D();
iren.SetInteractorStyle(style);
renWin.SetInteractor(iren);

// render the image
renWin.Render();
vtkCamera cam1 = ren1.GetActiveCamera();
cam1.ParallelProjectionOn();
ren1.ResetCameraClippingRange();
renWin.Render();

iren.Start();
}
}

```

## 29.105 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=vtkgdcm.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:vtkgdcm.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("vtkgdcmJava");
}

static FilenamesType fns = new FilenamesType();

public static void process(String path)
{
    fns.add( path );
}

// Process only files under dir
public static void visitAllFiles(File dir)
{
    if (dir.isDirectory())
    {
        String[] children = dir.list();
        for (int i=0; i<children.length; i++)
        {
            visitAllFiles(new File(dir, children[i]));
        }
    }
    else
    {
        process(dir.getPath());
    }
}

public void dointer(vtkImagePlaneWidget current_widget)
{
    int cstat = current_widget.GetCursorDataStatus();
    double[] v = current_widget.GetCurrentCursorPosition();
    //System.out.println( cstat );
    //System.out.println( v[0] );
    //System.out.println( v[1] );
    //System.out.println( v[2] );
    planeWidgetX.SetSliceIndex( (int)v[0] );
    planeWidgetY.SetSliceIndex( (int)v[1] );
    planeWidgetZ.SetSliceIndex( (int)v[2] );
    planeWidgetX.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetY.GetCurrentRenderer().ResetCameraClippingRange();
    planeWidgetZ.GetCurrentRenderer().ResetCameraClippingRange();
}

public void startinterX()
{
    dointer( planeWidgetX );
}

public void interX()
{
    dointer( planeWidgetX );
}

public void endinterX()
{
}

public void startinterY()
{
    dointer( planeWidgetY );
}

public void interY()
{
    dointer( planeWidgetY );
}

public void endinterY()
{
}

public void startinterZ()
{
    dointer( planeWidgetZ );
}

public void interZ()
{
    dointer( planeWidgetZ );
}

public void endinterZ()
{
}

```

```

        //System.out.println( "endinter" );
    }

    public static void AlignCamera(int slice_number, vtkImagePlaneWidget current_widget)
    {
        vtkImageData image = (vtkImageData)current_widget.GetInput();
        vtkRenderer ren = current_widget.GetCurrentRenderer();
        double[] origin = image.GetOrigin();
        double ox = origin[0];
        double oy = origin[1];
        double oz = origin[2];

        int wextent[] = image.GetWholeExtent();
        int xmin = wextent[0];
        int xmax = wextent[1];
        int ymin = wextent[2];
        int ymax = wextent[3];
        int zmin = wextent[4];
        int zmax = wextent[5];

        double[] spacing = image.GetSpacing();
        double sx = spacing[0];
        double sy = spacing[1];
        double sz = spacing[2];

        double cx = ox + (0.5 * (xmax - xmin)) * sx;
        double cy = oy + (0.5 * (ymax - ymin)) * sy;
        double cz = oz + (0.5 * (zmax - zmin)) * sz;
        double vx = 0, vy = 0, vz = 0;
        double nx = 0, ny = 0, nz = 0;
        int iaxis = current_widget.GetPlaneOrientation();
        if ( iaxis == 0 ) {
            vz = -1;
            nx = ox + xmax * sx;
            cx = ox + slice_number * sx;
        }
        else if ( iaxis == 1 ) {
            vz = -1;
            ny = oy + ymax * sy;
            cy = oy + slice_number * sy;
        }
        else {
            vy = 1;
            nz = oz + zmax * sz;
            cz = oz + slice_number * sz;
        }
        double px = cx + nx * 2;
        double py = cy + ny * 2;
        double pz = cz + nz * 3;

        vtkCamera camera = ren.GetActiveCamera();
        camera.SetViewUp(vx, vy, vz);
        camera.SetFocalPoint(cx, cy, cz);
        camera.SetPosition(px, py, pz);
        camera.OrthogonalizeViewUp();
        ren.ResetCameraClippingRange();
    }

    private vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
    private vtkImagePlaneWidget planeWidgetY = new vtkImagePlaneWidget();
    private vtkImagePlaneWidget planeWidgetZ = new vtkImagePlaneWidget();

    public void config()
    {
        //System.out.println( "config" );
        planeWidgetX.GetCurrentRenderer().ResetCamera();
        planeWidgetY.GetCurrentRenderer().ResetCamera();
        planeWidgetZ.GetCurrentRenderer().ResetCamera();
    }

    public void Run(String dirname)
    {
        File dir = new File(dirname);
        visitAllFiles(dir);

        IPPSorter ipp = new IPPSorter();
        ipp.SetComputeZSpacing( true );
        ipp.SetZSpacingTolerance( 1e-3 );
        boolean b = ipp.Sort( fns );
        if (!b)
        {

```

```

        //throw new Exception("Could not scan");
    }
    double ippzspacing = ipp.GetZSpacing();

    FilenamesType sorted = ipp.GetFilenames();
    vtkStringArray files = new vtkStringArray();
    long nfiles = sorted.size();
    //for( String f : sorted )
    for (int i = 0; i < nfiles; i++) {
        String f = sorted.get(i);
        files.InsertNextValue( f );
    }
    vtkGDCMImageReader reader = new vtkGDCMImageReader();
    reader.SetFileNames( files );
    reader.Update(); // get spacing value

    double[] spacing = reader.GetOutput().GetSpacing();

    vtkImageChangeInformation change = new vtkImageChangeInformation();
    change.SetInputConnection( reader.GetOutputPort() );
    change.SetOutputSpacing( spacing[0], spacing[1], ippzspacing );
    change.Update();

    System.out.println( change.GetOutput().toString() );

    vtkRenderer ren1 = new vtkRenderer();
    ren1.SetViewport(0., 0., 0.333, 1);
    ren1.SetBackground(0.1,0.2,0.4);
    vtkRenderer ren2 = new vtkRenderer();
    ren2.SetViewport(0.333, 0., 0.667, 1);
    ren2.SetBackground(0.1,0.2,0.4);
    vtkRenderer ren3 = new vtkRenderer();
    ren3.SetViewport(0.667, 0., 1., 1.);
    ren3.SetBackground(0.1,0.2,0.4);

    vtkRenderWindow renWin = new vtkRenderWindow();
    renWin.AddRenderer(ren1);
    renWin.AddRenderer(ren2);
    renWin.AddRenderer(ren3);

    vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
    iren.SetRenderWindow(renWin);

    vtkInteractorStyleImage style = new vtkInteractorStyleImage();
    iren.SetInteractorStyle( style );

    vtkCellPicker picker = new vtkCellPicker();
    picker.SetTolerance(0.005);

    vtkProperty ipwProp = new vtkProperty();

    //vtkImagePlaneWidget planeWidgetX = new vtkImagePlaneWidget();
    planeWidgetX.SetInteractor(iren);
    planeWidgetX.SetCurrentRenderer(ren1);
    planeWidgetX.SetDefaultRenderer(ren1);
    planeWidgetX.RestrictPlaneToVolumeOn();
    planeWidgetX.SetTexturePlaneProperty(ipwProp);
    //planeWidgetX.GetPlaneProperty().SetColor(1,0,0);
    //planeWidgetX.TextureInterpolateOff();
    //planeWidgetX.SetResliceInterpolateToNearestNeighbour();
    planeWidgetX.SetInput(change.GetOutput());
    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.SetInput(change.GetOutput());
    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.SetInput(change.GetOutput());
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 );
}
}

```

## 29.106 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.ucTOFInflow = 0x4
sAngio.ucRephasedImage = 0x1
sAngio.ucPhaseImage = 0x1
sEllipticalFilter.ucMode = 0x1
sPat.lAccelFactPE = 1
sPat.lAccelFact3D = 1
sPat.ucPATMode = 0x1
sPat.ucRefScanMode = 0x1
ucAutoMovie = 0x1
ucDisableChangeStoreImages = 0x1
ucReconstructionMode = 0x1
ucPHAPSMODE = 0x1
ucDixon = 0x1
lAverages = 2
adFlipAngleDegree[0] = 30
lScanTimeSec = 103
lTotalScanTimeSec = 112
dRefSNR = 165404.1473
dRefSNR_VOI = 165404.1473
tdefaultEVAProt = "%SiemensEvaDefProt%\Inline\Inline.evp"
tcurrentEVAProt = "%CURRENT EVAProt%\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:

...
<ParamFuncor."<TlmapFuncor">
{
    <Class> "<TlmapFuncor@IceImagePostProcFuncors">

    <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;
}

return 0;
}

```

## 29.107 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;
    }
}

```

## 29.108 NewSequence.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####

```

```

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)

```

## 29.109 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;
}

```

## 29.110 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;
}

```

## 29.111 PhilipsPrivateRescaleInterceptSlope.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18 python
19 """
20
21 import gdcm
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 = gdcm.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 = gdcm.PrivateTag(0x2005,0x09,"Philips MR Imaging DD 005")
42 tag2 = gdcm.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 gdcm gives us a reference
48 e11 = gdcm.DataElement( ds.GetDataElement( tag1 ) )
49 print e11
50 e12 = gdcm.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( gdcm.Tag(0x0028,0x1052) )
57 e12.SetTag( gdcm.Tag(0x0028,0x1053) )
58
59 ds.Insert( e11 )
60 ds.Insert( e12 )
61
62 w = gdcm.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"

```

## 29.112 PlaySound.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18     python PlaySound.py input.dcm
19 """
20
21 import gdcm
22 import sys
23
24 #filename = "/home/mmalaterre/Creatis/gdcmDataExtra/gdcmNonImageData/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(0x5400,0x01010)
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()

```

## 29.113 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;

```

```
}

```

## 29.114 PrivateDict.py

```
1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 """
17
18 import gdcm
19 import sys,os
20
21 if __name__ == "__main__":
22     #gdcm.Trace.DebugOn()
23     globInst = gdcm.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( gdcm.Tag(0x0029,0x0010) ,"SIEMENS CSA HEADER" )
34     pd = d.GetPrivateDict()
35     print pd.GetDictEntry( gdcm.PrivateTag(0x0029,0x0010,"SIEMENS CSA HEADER") )

```

## 29.115 PublicDict.cxx

```
/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.
=====*/

/*
 * Dummy example to show GDCM Dict(s) API (Part 6) + Collected Private Attributes:
 */

#include "gdcmGlobal.h"
#include "gdcmDicts.h"
#include "gdcmDict.h"
#include "gdcmCSAHeader.h"
#include "gdcmPrivateTag.h"

int main(int , char *[])
{
    const gdcm::Global& g = gdcm::Global::GetInstance(); // sum of all
        knowledge !
    const gdcm::Dicts &dicts = g.GetDicts();
    const gdcm::Dict &pub = dicts.GetPublicDict(); // Part 6

    //std::cout << pub << std::endl;

```

```

// 3 different ways to access the same information

// 1. From the public dict only:
gdcm::Tag patient_name(0x10,0x10);
const gdcm::DictEntry &entry1 = pub.GetDictEntry(patient_name);
std::cout << entry1 << std::endl;

// 2. From all dicts:
const gdcm::DictEntry &entry2 = dicts.GetDictEntry(patient_name);
std::cout << entry2 << std::endl;

// 3. This solution is the most flexible solution as you can request using the same
// API either a public tag or a private tag
const char *strowner = 0;
const gdcm::DictEntry &entry3 = dicts.GetDictEntry(patient_name,strowner);
std::cout << entry3 << std::endl;

// Private attributes:

// try with a private tag now:
const gdcm::PrivateTag &private_tag =
    gdcm::CSAHeader::GetCSAImageHeaderInfoTag();
//std::cout << private_tag << std::endl;
const gdcm::DictEntry &entry4 = dicts.GetDictEntry(private_tag,private_tag.
    GetOwner());
std::cout << entry4 << std::endl;

// Let's pretend that private lookup is on 0x10xx elements:
gdcm::PrivateTag dummy = private_tag;
dummy.SetElement( (uint16_t)(0x1000 + dummy.GetElement()) );
const gdcm::DictEntry &entry5 = dicts.GetDictEntry(dummy,dummy.
    GetOwner());
std::cout << entry5 << std::endl;

return 0;
}

```

## 29.116 QIDO-RS.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
#include "gdcmReader.h"
#include "gdcmWriter.h"
#include "gdcmJSON.h"

/*
 * Simple QIDO-RS round-trip to test implementation of gdcm::JSON
 * See Supl66 for details
 */
int main(int argc, char *argv[])
{
    if( argc < 2 ) return 1;
    using namespace gdcm;
    const char *filename = argv[1];
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() ) return 1;

    gdcm::JSON json;
    json.PrettyPrintOn();
    std::stringstream ss;
    const gdcm::File & f = reader.GetFile();
    json.Code( f.GetDataSet(), ss);
}

```



```

std::cout << ss.str() << std::endl;

gdcmm::Writer w;
gdcmm::File & ff = w.GetFile();
ff.GetHeader().SetDataSetTransferSyntax(
    gdcmm::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;
}

```

## 29.117 ReadAndDumpDICOMDIR.cxx

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

Copyright (c) 2006-2011 Mathieu Malaterre
All rights reserved.
See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.

This software is distributed WITHOUT ANY WARRANTY; without even
the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the above copyright notice for more information.

=====*/
/*
 * This example shows how to read and dump a DICOMDIR File
 *
 * Thanks:
 * Tom Marynowski (lordglub gmail) for contributing this example
 */
#include "gdcmmReader.h"
#include "gdcmmMediaStorage.h"

typedef std::set<gdcmm::DataElement> DataElementSet;
typedef DataElementSet::const_iterator ConstIterator;

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 << "Could not read: " << filename << std::endl;
        return 1;
    }
    std::stringstream strm;

    gdcmm::File &file = reader.GetFile();
    gdcmm::DataSet &ds = file.GetDataSet();
    gdcmm::FileMetaInformation &fmi = file.GetHeader();

    gdcmm::MediaStorage ms;
    ms.SetFromFile(file);
    if( ms != gdcmm::MediaStorage::MediaStorageDirectoryStorage
        )
    {
        std::cout << "This file is not a DICOMDIR" << std::endl;
        return 1;
    }

    if( fmi.FindDataElement( gdcmm::Tag(0x0002, 0x0002)) )
    {
        strm.str("");
        fmi.GetDataElement( gdcmm::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 << "          STUDY UID : " << strm.str() << std::endl;

                    //STUDY DATE
                    strm.str("");
                    if (sqi->GetItem(itemused).FindDataElement(
                    gdcm::Tag (0x0008, 0x0020)))
                        sqi->GetItem(itemused).GetDataElement(
                        gdcm::Tag (0x0008, 0x0020)).GetValue().Print(strm);

```

```

std::cout << "          STUDY DATE : " << strm.str() << std::endl;

//STUDY DESCRIPTION
strm.str("");
if (sqi->GetItem(itemused).FindDataElement(
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;
}

```

## 29.118 ReadAndDumpDICOMDIR.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 # File: ReadAndDumpDICOMDIR.py
14 #
15 # Author: Lukas Batteau (lbatteau gmail)
16 #
17 # This example shows how to read and dump a DICOMDIR File.
18 # Based on Tom Marynowski's (lordglub gmail) example.
19 #
20 # Usage:
21 # python ReadAndDumpDICOMDIR.py [DICOMDIR file]
22 #####
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 = gdcm.MediaStorage()
58     mediaStorage.SetFromFile(file)
59     if (gdcm.MediaStorage.GetMSType(str(mediaStorage)) !=
gdcm.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
130         # Print study description
131         if (item.FindDataElement(gdcm.Tag(0x0008, 0x1030))):
132             value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x1030)).GetValue(
133             ))
134             print value
135
136         # Next
137         itemNr = itemNr + 1
138         item = sequence.GetItem(itemNr)
139         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
140             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).
141             GetValue())
142
143         # SERIES
144         while (value.strip() == "SERIES"):
145             print value.strip()
146
147         # Print series UID
148         if (item.FindDataElement(gdcm.Tag(0x0020, 0x000e))):
149             value = str(item.GetDataElement(gdcm.Tag(0x0020, 0x000e)).
150             GetValue())
151             print value
152
153         # Print series modality
154         if (item.FindDataElement(gdcm.Tag(0x0008, 0x0060))):
155             value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x0060)).
156             GetValue())
157             print "Modality"
158             print value
159
160         # Print series description
161         if (item.FindDataElement(gdcm.Tag(0x0008, 0x103e))):
162             value = str(item.GetDataElement(gdcm.Tag(0x0008, 0x103e)).
163             GetValue())
164             print "Description"
165             print value
166
167         # Next
168         itemNr = itemNr + 1
169         item = sequence.GetItem(itemNr)
170         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
171             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1430)).
172             GetValue())
173
174         # IMAGE
175         while (value.strip() == "IMAGE"):
176             print value.strip()
177
178         # Print image UID
179         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1511))):
180             value = str(item.GetDataElement(gdcm.Tag(0x0004, 0x1511)).
181             GetValue())
182             print value
183
184         # Next
185         if (itemNr < sequence.GetNumberOfItems()):
186             itemNr = itemNr + 1
187         else:
188             break
189
190         item = sequence.GetItem(itemNr)
191         if (item.FindDataElement(gdcm.Tag(0x0004, 0x1430))):
192             value = str(item.GetDataElement(
193             gdcm.Tag(0x0004, 0x1430)).GetValue())
194
195         # Next
196         itemNr = itemNr + 1

```

## 29.119 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];

    // Instantiate the reader:
    gdcm::Reader reader;
    reader.SetFileName( filename );
    if( !reader.Read() )
    {
        std::cerr << "Could not read: " << filename << std::endl;
        return 1;
    }

    // The output of gdcm::Reader is a gdcm::File
    gdcm::File &file = reader.GetFile();

    // the dataset is the the set of element we are interested in:
    gdcm::DataSet &ds = file.GetDataSet();

    const gdcm::Global& g = gdcm::Global::GetInstance();
    const gdcm::Dicts &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 &de1 =
    pubdict.GetDictEntryByName("Patient Name", tPatientName);

    std::cout << "Found: " << tPatientName << std::endl;

    // Indeed the attribute could not be found. Since DICOM 2003, Patient Name
    // has become Patient's Name.

    Tag tPatientsName;
    //const DictEntry &de2 =
    pubdict.GetDictEntryByName("Patient's Name", tPatientsName);

    std::cout << "Found: " << tPatientsName << std::endl;

    // Let's try to read an arbitrary DICOM Attribute:
    Tag tDoseGridScaling;
    //const DictEntry &de3 =
    pubdict.GetDictEntryByName("Dose Grid Scaling", tDoseGridScaling);

    std::cout << "Found: " << tDoseGridScaling << std::endl;

    if( ds.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 ieee double)
Attribute<0x3004,0x000e>::ArrayType v = at.
    GetValue();
std::cout << "DoseGridScaling=" << v << std::endl;
}

return 0;
}

```

## 29.120 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;
}

```



## 29.121 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);
        }
    }
}

```

## 29.122 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;
}

```

## 29.123 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;
}

```

## 29.124 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.SetInput( reader.GetOutput() );
writer.SetFileName( "ReadSeriesIntoVTK.mhd" );
writer.Write();

System.out.println("Success writing: " + writer.GetFileName() );
}
}

```

## 29.125 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;
}

```

## 29.126 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.WriteLine( "GetReferenceCount: " + testing1.GetReferenceCount() + "\n");
            System.Console.WriteLine( "GetReferenceCount: " + testing2.GetReferenceCount() + "\n");
            System.Console.WriteLine( "GetReferenceCount: " + testing3.GetReferenceCount() + "\n");
        }

        using (vtkGDCMImageReader reader3 = new vtkGDCMImageReader())
        {
            System.Console.WriteLine( "GetReferenceCount: " + reader3.GetReferenceCount() + "\n");
        }

        using (vtkGDCMImageWriter writer3 = vtkGDCMImageWriter.New())
        {
            System.Console.WriteLine( "GetReferenceCount: " + writer3.GetReferenceCount() + "\n");
        }

        // C# destructor will call ->Delete on all C++ object as expected.
        return 0;
    }
}

```

## 29.127 ReformatFile.cs

This is a C++ example on how to use [gdcml::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;
    }
}

```

```

    }
}

```

## 29.128 RemovePrivateTags.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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     # Instanciate 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)
51     # DICOM file
52     # (application level)

```

## 29.129 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;
        }
    }
}

```

## 29.130 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 );
        gdcmm::Directory d;
        d.Load(fileName);
        gdcmm::Directory::FileNamesType const &files = d.
        GetFileNames();

        gdcmm::IPPSorter s;
        s.SetComputeZSpacing( true );
        s.SetZSpacingTolerance( 1e-3 );
        bool b = s.Sort( files );
        if( !b )
        {
            std::cerr << "Failed to sort:" << fileName << std::endl;
            //return ;
        }
        //std::cout << "Sorting succeeded:" << std::endl;
        //s.Print( std::cout );

        //std::cout << "Found z-spacing:" << std::endl;
        //std::cout << s.GetZSpacing() << std::endl;
        double ippzspacing = s.GetZSpacing();

```

```

const std::vector<std::string> & sorted = s.GetFilesNames();
vtkStringArray *vtkfiles = vtkStringArray::New();
std::vector< std::string >::const_iterator it = sorted.begin();
for( ; it != sorted.end(); ++it)
{
    const std::string &f = *it;
    vtkfiles->InsertNextValue( f.c_str() );
}

    //_reader->SetDirectoryName(fileName);
    //_reader->SetFileNames( d->GetFiles() );
    _reader->SetFileNames( vtkfiles );
    _reader->Update();

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], appzspacing );
    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.
    _imageView = vtkImageViewer2::New();
    _imageView->SetupInteractor(_interactor);
    _imageView->SetSize(400, 400);
    _imageView->SetColorWindow(1024);
    _imageView->SetColorLevel(800);
    _imageView->SetInputConnection(_reslice->GetOutputPort());
    _imageView->GetImageActor()->SetOpacity(0.5);

    _annotation = vtkTextActor::New();
    _annotation->SetTextScaleModeToViewport();
    _imageView->GetRenderer()->AddActor(_annotation);

    // Add the cut shape actor to the renderer.
    _imageView->GetRenderer()->AddActor(_ROIActor);

    // Set up the key handler.
    vtkSmartPointer<KeyCallback> callback = vtkSmartPointer<KeyCallback>::New();
    callback->SetCallbackData(this);
    _interactor->AddObserver(vtkCommand::KeyPressEvent, callback);

    _interactor->Initialize();
}

void Start()
{
    _interactor->Start();
}

void ResetOrientation()
{
    vtkSmartPointer<vtkMatrix4x4> matrix =
        vtkSmartPointer<vtkMatrix4x4>::New();
    matrix->Identity();

    SetOrientation(matrix);
}

```

```

// Make sure the orientation of the vtkImageReslice and
// vtkTransform are in sync.
void SetOrientation(vtkMatrix4x4* matrix)
{
    _reslice->SetResliceAxes(matrix);
    _reslice->Update();

    vtkMatrix4x4* inverse = vtkMatrix4x4::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    _transform->SetMatrix(inverse);
    _transform->Update();
}

// Set the current slice of the current view.
void SetSlice(int slice)
{
    std::stringstream posString;

    double    center[3];
    double    spacing[3];
    double    origin[3];
    double    point[4];
    double    newPoint[4];

    vtkImageData* imageData;
    int newSlice;

    // Try to make sure the extents of the reslice are updated.
    // PROBLEM: It doesn't seem to work when changing the orientation.
    imageData=vtkImageData::SafeDownCast(_reslice->GetOutput());
#ifdef (VTK_MAJOR_VERSION >= 6)
    assert(0);
#else
    imageData->UpdateInformation();
#endif

    // Let vtkImageViewer2 handle the slice limits.
    _imageView->SetSlice(slice);
    newSlice=GetSlice();

    imageData->GetCenter(center);
    imageData->GetSpacing(spacing);
    imageData->GetOrigin(origin);

    // Compute the position of the center of the slice based on the
    // spacing of the slices. The resliced axis will always
    // be the "Z" axis.
    point[0]=center[0];
    point[1]=center[1];
    point[2]=(newSlice * spacing[2]) + origin[2];
    point[3]=1.0;

    // Convert the coordinate from the reslice coordinate system to the
    // original image coordinate system.
    // PROBLEM: Logically this seems like it should have been multiplied
    // by the inverse to translate from the resliced coordinate system to
    // the original coordinate system. However, multiplying by the inverse
    // sticks the plane in the wrong place completely. Using the original
    // matrix at least gets the Z coordinate right.
    vtkMatrix4x4* matrix=_reslice->GetResliceAxes();
    vtkSmartPointer<vtkMatrix4x4> inverse =
        vtkSmartPointer<vtkMatrix4x4>::New();
    vtkMatrix4x4::Invert(matrix, inverse);

    matrix->MultiplyPoint(point, newPoint);
    _plane->SetOrigin(newPoint[0], newPoint[1], newPoint[2]);

    // Annotate the image.
    posString << "Position: (" << newPoint[0] << ", " << newPoint[1]
        << ", " << newPoint[2] << ") Slice: " << newSlice;
    _annotation->SetInput(posString.str());

    _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());
    if (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;
}

```

## 29.131 ReWriteSCAsMR.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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( gdcm.Tag(0x0008,0x1150) ):
33         ReferencedSOPClassUID = nestedds.GetDataElement( gdcm.Tag(0x0008,0x1150) ).
34         raw = ReferencedSOPClassUID.GetByteValue().GetPointer()
35         uids = gdcm.UIDs()
36         # what is the actual object we are looking at ?
37         ms = gdcm.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 = gdcm.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         gdcm.ImageHelper.SetForcePixelSpacing( True )
63         mrspacing = gdcm.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         gdcm.ImageHelper.SetForceRescaleInterceptSlope( True )
70         ris = gdcm.ImageHelper.GetRescaleInterceptSlopeValue(
         r.GetFile() )
71         image.SetIntercept( ris[0] )
72         image.SetSlope( ris[1] )
73
74     outfilename = sys.argv[2]
75     w = gdcm.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)

```

## 29.132 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(
        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 = "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;
}

```

## 29.133 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 )
#ifdef 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)
{
#ifdef 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();
#ifdef 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;
}

```



## 29.134 ScanDirectory.cs

This is a C# example on how to use [gdcm::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:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/ScanDirectory.exe /path/to/gdcmData/
 */
using System;
using gdcm;

public class ScanDirectory
{
    public static int Main(string[] args)
    {
        {
            string directory = args[0];
            Tag t = new Tag(0x8,0x8);

            Directory d = new Directory();
            uint nfiles = d.Load( directory );
            if(nfiles == 0) return 1;
            //System.Console.WriteLine( "Files:\n" + d.toString() );

            //Scanner s = new Scanner();
            SmartPtrScan sscan = Scanner.New();
            Scanner s = sscan.__ref__();
            SimpleSubjectWatcher watcher = new SimpleSubjectWatcher(s, "MySimple");
            s.AddTag( t );
            bool b = s.Scan( d.GetFileNames() );
            if(!b) return 1;

            System.Console.WriteLine( "Scan:\n" + s.toString() );

            System.Console.WriteLine( "success" );
            return 0;
        }
    }
}
```

## 29.135 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");
}

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" );
}
}

```

## 29.136 ScanDirectory.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 import gdcm
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 = gdcmm.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)

```

## 29.137 SendFileSCU.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.

=====*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Perso/gdcmm-gcc/bin
 * $ mono bin/SendFileSCU.exe server port input.dcm
 */
using System;
using gdcmm;

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;
        }
    }
}

```

## 29.138 SimplePrint.cs

This is a C# example on how to use `gdcmm::SWIGDataSet`

```

/*=====
Program: GDCM (Grassroots DICOM). A DICOM library

```

Copyright (c) 2006-2011 Mathieu Malaterre  
 All rights reserved.  
 See Copyright.txt or <http://gdcm.sourceforge.net/Copyright.html> for details.

This software is distributed WITHOUT ANY WARRANTY; without even  
 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR  
 PURPOSE. See the above copyright notice for more information.

```

=====*/
/*
  Convertor convertor = new Convertor();
  int a = convertor.Convert<int>( some_int_blob );
  double b = convertor.Convert<double>( some_double_blob );
*/

/*
 * Usage:
 * $ export LD_LIBRARY_PATH=$HOME/Projects/gdcm/debug-gcc/bin
 * $ mono bin/SimplePrint.exe gdcmData/012345.002.050.dcm
 */
using System;
using gdcm;

public class SimplePrint
{
    public static void RecurseDataSet(File f, DataSet ds, string indent)
    {
        CSharpDataSet cds = new CSharpDataSet(ds);
        while(!cds.IsAtEnd())
        {
            DataElement de = cds.GetCurrent();
            // Compute VR from the toplevel file, and the currently processed dataset:
            VR vr = DataSetHelper.ComputeVR(f, ds, de.GetTag() );

            if( vr.Compatible( new VR(VR.VRType.SQ) ) )
            {
                uint uvl = (uint)de.GetVL(); // Test cast is ok
                System.Console.WriteLine( indent + de.GetTag().toString() + ":" + uvl ); // why not ?
                //SequenceOfItems sq = de.GetSequenceOfItems();
                // GetValueAsSQ handle more cases than GetSequenceOfItems
                SmartPtrSQ sq = de.GetValueAsSQ();
                uint n = sq.GetNumberOfItems();
                for( uint i = 1; i <= n; i++) // item starts at 1, not 0
                {
                    Item item = sq.GetItem( i );
                    DataSet nested = item.GetNestedDataSet();
                    RecurseDataSet( f, nested, indent + " " );
                }
            }
            else
            {
                {
                    System.Console.WriteLine( indent + de.toString() );
                }
            }
            cds.Next();
        }
    }

    public static int Main(string[] args)
    {
        string filename = args[0];
        Reader reader = new Reader();
        reader.SetFileName( filename );
        bool ret = reader.Read();
        if( !ret )
        {
            return 1;
        }
        File f = reader.GetFile();
        DataSet ds = f.GetDataSet();

        RecurseDataSet( f, ds, "" );

        return 0;
    }
}

```

## 29.139 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;
        }
    }
}

```

## 29.140 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 "gdcmScanner.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::Scanner> sp = new
        gdcm::Scanner;
    gdcm::Scanner &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),
    };
    s.AddTag( tag_array[0] );
    s.AddTag( tag_array[1] );
    s.AddTag( tag_array[2] );

    gdcm::Directory::FileNamesType filenames;
    filenames.push_back( filename );
    filenames.push_back( filename_invalid );

    if( !s.Scan( filenames ) )
    {
        return 1;
    }

    //s.Print( std::cout );

    if( s.IsKey( filename ) )
    {
        std::cout << "INFO:" << filename << " is a proper Key for the Scanner (this is a DICOM file)" <<
            std::endl;
    }

    if( !s.IsKey( filename_invalid ) )
    {

```

```

    std::cout << "INFO:" << filename_invalid << " is not a proper Key for the Scanner (this is either not a
        DICOM file or file does not exist)" << std::endl;
}

gdcmm::Scanner::TagToValue const &ttv = s.GetMapping(filename);

const gdcmm::Tag *ptag = tag_array;
for( ; ptag != tag_array + 3; ++ptag )
{
    gdcmm::Scanner::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;
}

```

## 29.141 SortImage.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 "gdcmmSorter.h"
#include "gdcmmScanner.h"
#include "gdcmmDataSet.h"
#include "gdcmmAttribute.h"

bool mysort(gdcmm::DataSet const & ds1, gdcmm::DataSet const & ds2 )
{
    //gdcmm::Attribute<0x0020,0x0013> at1; // Instance Number
    gdcmm::Attribute<0x0018,0x1060> at1; // Trigger Time
    gdcmm::Attribute<0x0020,0x0032> at1l; // Image Position (Patient)
    at1.Set( ds1 );
    at1l.Set( ds1 );
    //gdcmm::Attribute<0x0020,0x0013> at2;
    gdcmm::Attribute<0x0018,0x1060> at2;
    gdcmm::Attribute<0x0020,0x0032> at2l;
    at2.Set( ds2 );
    at2l.Set( ds2 );
    if( at1l == at2l )
    {
        return at1 < at2;
    }
    return at1l < at2l;
}

```

```

}

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.GetFilesNames() );

    std::cout << "Sorter:" << std::endl;
    sorter.Print( std::cout );

    gdcm::Sorter sorter2;
    sorter2.SetSortFunction( mysort_part1 );
    sorter2.StableSort( dir.GetFilesNames() );
    sorter2.SetSortFunction( mysort_part2 );
    sorter2.StableSort( sorter2.GetFilesNames() ); // IMPORTANT
    sorter2.SetSortFunction( mysort_dummy );
    sorter2.StableSort( sorter2.GetFilesNames() ); // IMPORTANT

    std::cout << "Sorter2:" << std::endl;
    sorter2.Print( std::cout );

    gdcm::Scanner s;
    s.AddTag( gdcm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( dir.GetFilesNames() );

    //s.Print( std::cout );

    // Count how many different IPP there are:
    const gdcm::Scanner::ValueType &values = s.GetValues();
    size_t nvalues = values.size();
    std::cout << "There are " << nvalues << " different type of values" << std::endl;

    //std::cout << "nfiles=" << nfiles << std::endl;
    if( nfiles % nvalues != 0 )
    {
        std::cerr << "Impossible: this is a not a proper series" << std::endl;
        return 1;
    }
    std::cout << "Series is composed of " << (nfiles/nvalues) << " different 3D volumes" << std::endl;

    return 0;
}

```

## 29.142 SortImage.py

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 #       This software is distributed WITHOUT ANY WARRANTY; without even
10 #       the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #       PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 """
16 Usage:
17
18   python SortImage.py dirname
19 """
20
21 import gdcm
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 = gdcm.Directory()
36     d.Load( dirname )
37
38     print d
39
40     sorter = gdcm.Sorter()
41     sorter.SetSortFunction( MySort )
42     #sorter.AddObserver( "ProgressEvent", PrintProgress )
43     sorter.Sort( d.GetFileNames() )
44
45     print "Sorter:"
46     print sorter

```

## 29.143 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;
    }
}

```

## 29.144 StandardizeFiles.cs

This is a C++ example on how to use [gdcm::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;
    }
}

```

## 29.145 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 del( gdcm::Tag(0x8,0x16) );
del.SetVR( gdcm::VR::UI );
gdcm::MediaStorage ms(
    gdcm::MediaStorage::VLWholeSlideMicroscopyImageStorage
);
del.SetByteValue( ms.GetString(), strlen(ms.GetString()) );
ds.Insert( del );

const char mystr[] = "MONOCHROME2 ";
gdcm::DataElement de2( gdcm::Tag(0x28,0x04) );
//de.SetTag(gdcm::Tag(0x28,0x04));
de2.SetVR( gdcm::VR::CS );
de2.SetByteValue(mystr, strlen(mystr));
ds.Insert( de2 );

gdcm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds.Insert( Number_Of_Frames.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0010> row = {extent[0]/a}; //
ds.Insert( row.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0011> col = {extent[1]/a}; //
ds.Insert( col.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0100> at = {8};
ds.Insert( at.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0002> at1 = {1}; //
ds.Insert( at1.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0101> at2 = {8};
ds.Insert( at2.GetAsDataElement() );

gdcm::Attribute<0x0028,0x0102> at3 = {7};
ds.Insert( at3.GetAsDataElement() );
/*
ds1.Remove( gdcm::Tag(0x0028,0x0008) );

```



```

gdcmm::Attribute<0x0028,0x0008> Number_Of_Frames = {1};
ds1.Insert( Number_Of_Frames.GetAsDataElement() );
*/
theStreamWriter.SetFile(file);

if (!theStreamWriter.WriteImageInformation())
{
    std::cerr << "unable to write image information" << std::endl;
    return 1; //the CanWrite function should prevent getting here, else,
    //that's a test failure
}
std::vector<unsigned int> extent1 = gdcmm::ImageHelper::GetDimensionsValue
(file);

unsigned short xmax = extent1[0];
unsigned short ymax = extent1[1];
unsigned short theChunkSize = 1;
unsigned short ychunk = extent1[1]/theChunkSize; //go in chunk sizes of theChunkSize
unsigned short zmax = 1;

std::cout<< "\n Row: "<<extent1[0] <<"\n Col :"<< extent1[1]<< "\n Resolution :"<< extent1[2] <<
std::endl;

if (xmax == 0 || ymax == 0)
{
    std::cerr << "Image has no size, unable to write zero-sized image." << std::endl;
    return 0;
}

int z, y, nexty;
unsigned long prevLen = 0; //when going through the char buffer, make sure to grab
//the bytes sequentially. So, store how far you got in the buffer with each iteration.

for (z = 0; z < zmax; ++z){
    for (y = 0; y < ymax; y += ychunk){
        nexty = y + ychunk;
        if (nexty > ymax) nexty = ymax;
        theStreamWriter.DefinePixelExtent(0, xmax, y, nexty, z, z+1);
        unsigned long len = theStreamWriter.DefineProperBufferLength();
        std::cout << "\n" <<len;
        char* finalBuffer1 = new char[len];
        memcpy(finalBuffer1, &(finalBuffer[prevLen]), len);
        std::cout << "\nable to write";

        if (!theStreamWriter.Write(finalBuffer1, len)){
            std::cerr << "writing failure:" << "output.dcm" << " at y = " << y << " and z= " << z <<
std::endl;
            delete [] finalBuffer1;
            delete [] finalBuffer;
            return 1;
        }
        delete [] finalBuffer1;
        prevLen += len;
    }
}
delete [] finalBuffer;
std::cout << "all is set";

return true;
}

int main(int argc, char *argv[])
{
    if( argc < 3 )
    {
        std::cerr << argv[0] << " input.dcm output.dcm Resolution" << std::endl;
        return 1;
    }

    const char *filename = argv[1];
    const char *outfilename = argv[2];
    char *res = argv[3];

    int resolution = atoi(res);

    gdcmm::StreamImageWriter theStreamWriter;

    std::ofstream of;
    of.open( outfile, std::ios::out | std::ios::binary );

```

```

theStreamWriter.SetStream(of);

// else
// First of get rid of warning/debug message
gdcm::Trace::DebugOn();
gdcm::Trace::WarningOn();

if(!StreamImageRead( theStreamWriter, filename, outfilename, resolution))
    return 1;

uint16_t firstTag1 = 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;
}

```

## 29.146 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);
}

```

```

std::cout << std::hex << "v1: " << v1 << std::endl;
if( v1 != 0x4000000 )
{
    return 1;
}

return 0;
}

int TestByteSwap(int , char *[])
{
    gdcm::SwapCode sc = gdcm::SwapCode::Unknown;
    if ( gdcm::ByteSwap<uint16_t>::SystemIsBigEndian() )
    {
        sc = gdcm::SwapCode::BigEndian;
    }
    else if ( gdcm::ByteSwap<uint16_t>::SystemIsLittleEndian() )
    {
        sc = gdcm::SwapCode::LittleEndian;
    }
    if( sc == gdcm::SwapCode::Unknown )
    {
        return 1;
    }

    std::cout << "sc: " << sc << std::endl;

    uint16_t t = 0x1234;
    gdcm::ByteSwap<uint16_t>::SwapFromSwapCodeIntoSystem(
        t, sc);
    if( sc == gdcm::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 == gdcm::SwapCode::LittleEndian )
    {
        if( t != 0x1234 )
        {
            std::cerr << std::hex << "t: " << t << std::endl;
            return 1;
        }
    }
}

union { char n[2]; uint16_t tn; } ul6;
memcpy(ul6.n, &t, 2 );
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
    (&ul6.tn, sc, 1);
uint16_t tn = ul6.tn;
if( sc == gdcm::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 == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x1234 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;
        return 1;
    }
}
gdcm::ByteSwap<uint16_t>::SwapRangeFromSwapCodeIntoSystem
    (&ul6.tn, gdcm::SwapCode::BigEndian, 1);
tn = ul6.tn;
if( sc == gdcm::SwapCode::LittleEndian )
{
    if( tn != 0x3412 )
    {
        std::cerr << std::hex << "tn: " << tn << std::endl;

```

```

        return 1;
    }
}
else if ( sc == gdcm::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,2);
if ( array[0] != 0x3412 )
{
    return 1;
}

return 0;
}

```

## 29.147 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 != gdcmm::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
    gdcmm::Trace::DebugOff();
    gdcmm::Trace::WarningOff();
    int r = 0, i = 0;
    const char *filename;
    const char * const *filenames = gdcmm::Testing::GetFileNames();
    while( (filename = filenames[i]) )
    {
        r += TestRead( filename );
        ++i;
    }

    return r;
}

```

## 29.148 TestReader.py

This is a C++ example on how to use `gdcmm::Reader`

```

1 #####
2 #
3 #   Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 #   Copyright (c) 2006-2011 Mathieu Malaterre
6 #   All rights reserved.
7 #   See Copyright.txt or http://gdcmm.sourceforge.net/Copyright.html for details.
8 #
9 #   This software is distributed WITHOUT ANY WARRANTY; without even
10 #   the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 #   PURPOSE. See the above copyright notice for more information.
12 #
13 #####
14
15 import gdcmm
16 import os,sys
17
18 def TestRead(filename, verbose = False):
19     r = gdcmm.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         sucess += TestRead( filename, True )
31     except:
32         # loop over all files:
33         gdcmm.Trace.DebugOff()
34         gdcmm.Trace.WarningOff()
35         t = gdcmm.Testing()
36         nfiles = t.GetNumberOfFileNames()
37         for i in range(0,nfiles):
38             filename = t.GetFileName(i)
39             sucess += TestRead( filename )
40
41
42     # Test succeed ?
43     sys.exit(sucess == 0)

```

## 29.149 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 *scalarpointer;
// 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 gdcm::Image &image = reader.GetImage();
        unsigned long len = image.GetBufferLength();
        char * pointer = params->scalarpoiner;

    #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;
}

```

## 29.150 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 "gdcmlODs.h"
#include "gdcmlOD.h"
#include "gdcmlMacros.h"
#include "gdcmlODEntry.h"
#include "gdcmlModules.h"
#include "gdcmlModule.h"
#include "gdcmlAnonymizer.h"
#include "gdcmlDicts.h"

int main(int , char *[])
{
    using namespace gdcml;
    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 =
        gdcml::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;
}

```

## 29.151 uid\_unique.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 "gdcmUIDGenerator.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 );
    }
}

```

## 29.152 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 )
{

```

```

    gdcmm::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> iop1;
    gdcmm::Attribute<0x0020,0x0037> iop1;
    iop1.Set( ds1 );
    iop1.Set( ds1 );
    gdcmm::Attribute<0x0020,0x0032> iop2;
    gdcmm::Attribute<0x0020,0x0037> iop2;
    iop2.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]*iop1[i];
    double dist2 = 0;
    for( int i = 0; i < 3; ++i) dist2 += normal[i]*iop2[i];

    std::cout << dist1 << ", " << dist2 << std::endl;
    return dist1 < dist2;
}

int main(int argc, char *argv[])
{
    const char *extradataroot = gdcmm::Testing::GetDataExtraRoot();
    std::string dir1;
    if( argc < 2 )
    {
        if( !extradataroot )
        {
            return 1;
        }
        dir1 = extradataroot;
        dir1 += "/gdcmmSampleData/ForSeriesTesting/VariousIncidences/ST1";
    }
    else
    {
        dir1 = argv[1];
    }

    gdcmm::Directory d;
    d.Load( dir1.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 gdcmm::Tag t2(0x0020,0x000e); // Series Instance UID
//const gdcmm::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:
gdcmm::Directory::FileNamesType l2 = s0.GetKeys();
const size_t nfiles2 = l2.size();
std::cout << nfiles2 << std::endl;

if ( nfiles2 > nfiles )
{
    return 1;
}

gdcmm::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;
{
    gdcmm::Scanner s;
    s.AddTag( gdcmm::Tag(0x20,0x32) ); // Image Position (Patient)
    //s.AddTag( gdcmm::Tag(0x20,0x37) ); // Image Orientation (Patient)
    s.Scan( d.GetFileNames() );

    //s.Print( std::cout );

    const gdcmm::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;
}

gdcmm::Directory::FileNamesType sorted_files = sorter.
    GetFileNames();

// Which means we can take nvalues files at a time and execute gdcmm::IPPSorter on it:
gdcmm::IPPSorter ippsorter;
gdcmm::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;
}

```

## 29.153 WriteBuffer.py

```

1 #####
2 #
3 # Program: GDCM (Grassroots DICOM). A DICOM library
4 #
5 # Copyright (c) 2006-2011 Mathieu Malaterre
6 # All rights reserved.
7 # See Copyright.txt or http://gdcm.sourceforge.net/Copyright.html for details.
8 #
9 # This software is distributed WITHOUT ANY WARRANTY; without even
10 # the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 # PURPOSE. See the above copyright notice for more information.
12 #
13 #####
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 (fffe,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 (fffe,e00d) na (ItemDelimitationItem) # 0, 0 ItemDelimitationItem
32 (fffe,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 (fffe,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

AE  
    gdcmm::VR, [865](#)

AES128\_CIPHER  
    gdcmm::CryptographicMessageSyntax, [266](#)

AES192\_CIPHER  
    gdcmm::CryptographicMessageSyntax, [266](#)

AES256\_CIPHER  
    gdcmm::CryptographicMessageSyntax, [266](#)

ALGOType\_END  
    gdcmm::Segment, [658](#)

ARGB  
    gdcmm::PhotometricInterpretation, [578](#)

AS  
    gdcmm::VR, [865](#)

AT  
    gdcmm::VR, [865](#)

AUTOMATIC  
    gdcmm::Segment, [658](#)

AXIAL  
    gdcmm::Orientation, [553](#)

AmbulatoryECGWaveformStorage  
    gdcmm::MediaStorage, [517](#)  
    gdcmm::UIDs, [782](#)

Audio  
    gdcmm::MediaStorage, [518](#)

AudioSRStorageTrialRetired  
    gdcmm::UIDs, [783](#)

BLUE  
    gdcmm::LookupTable, [505](#)

backslash  
    gdcmm, [128](#)

BadBigEndian  
    gdcmm::SwapCode, [737](#)

BadLittleEndian  
    gdcmm::SwapCode, [737](#)

BasicAnnotationBoxSOPClass  
    gdcmm::UIDs, [781](#)

BasicColorImageBoxSOPClass  
    gdcmm::UIDs, [781](#)

BasicColorPrintManagementMetaSOPClass  
    gdcmm::UIDs, [781](#)

BasicFilmBoxSOPClass  
    gdcmm::UIDs, [781](#)

BasicFilmSessionSOPClass  
    gdcmm::UIDs, [781](#)

BasicGrayscaleImageBoxSOPClass  
    gdcmm::UIDs, [781](#)

BasicGrayscalePrintManagementMetaSOPClass  
    gdcmm::UIDs, [781](#)

BasicPrintImageOverlayBoxSOPClassRetired  
    gdcmm::UIDs, [782](#)

BasicStudyContentNotificationSOPClassRetired  
    gdcmm::UIDs, [781](#)

BasicTextSR  
    gdcmm::MediaStorage, [517](#)

BasicTextSRStorage  
    gdcmm::UIDs, [783](#)

BasicVoiceAudioWaveformStorage  
    gdcmm::MediaStorage, [517](#)  
    gdcmm::UIDs, [782](#)

BigEndian  
    gdcmm::SwapCode, [737](#)

black  
    gdcmm::terminal, [139](#)

BlendingSoftcopyPresentationStateStorageSOPClass  
    gdcmm::UIDs, [783](#)

blink  
    gdcmm::terminal, [139](#)

blue  
    gdcmm::terminal, [140](#)

BreastImagingRelevantPatientInformationQuery  
    gdcmm::UIDs, [784](#)

BreastTomosynthesisImageStorage  
    gdcmm::MediaStorage, [518](#)  
    gdcmm::UIDs, [786](#)

bright  
    gdcmm::terminal, [139](#)

C\_CANCEL\_RQ  
    gdcmm::network::DIMSE, [329](#)

C\_ECHO\_RQ  
    gdcmm::network::DIMSE, [329](#)

C\_ECHO\_RSP  
    gdcmm::network::DIMSE, [329](#)

C\_FIND\_RQ  
    gdcmm::network::DIMSE, [328](#)

C\_FIND\_RSP  
    gdcmm::network::DIMSE, [328](#)

C\_GET\_RQ  
    gdcmm::network::DIMSE, [328](#)

C\_GET\_RSP



- gdcm::network::DIMSE, 328
- C\_MOVE\_RQ
  - gdcm::network::DIMSE, 328
- C\_MOVE\_RSP
  - gdcm::network::DIMSE, 329
- C\_STORE\_RQ
  - gdcm::network::DIMSE, 328
- C\_STORE\_RSP
  - gdcm::network::DIMSE, 328
- CALIBRATED
  - gdcm::Spacing, 702
- CAPIC
  - gdcm::CryptoFactory, 264
- CMYK
  - gdcm::PhotometricInterpretation, 578
- CONDENSED\_STYLE
  - gdcm::Printer, 611
- CONSOLE
  - gdcm::terminal, 140
- CORONAL
  - gdcm::Orientation, 553
- CS
  - gdcm::VR, 865
- CSANonImageStorage
  - gdcm::MediaStorage, 517
- CT\_private\_ELE
  - gdcm::TransferSyntax, 766
- CTImageStorage
  - gdcm::MediaStorage, 516
  - gdcm::UIDs, 782
- CardiacElectrophysiologyWaveformStorage
  - gdcm::MediaStorage, 517
  - gdcm::UIDs, 782
- CardiacRelevantPatientInformationQuery
  - gdcm::UIDs, 785
- ChestCADSRStorage
  - gdcm::UIDs, 784
- ColorSoftcopyPresentationStateStorageSOPClass
  - gdcm::UIDs, 783
- ComprehensiveSR
  - gdcm::MediaStorage, 517
- ComprehensiveSRStorage
  - gdcm::UIDs, 783
- ComprehensiveSRStorageTrialRetired
  - gdcm::UIDs, 783
- ComputedRadiographyImageStorage
  - gdcm::MediaStorage, 516
  - gdcm::UIDs, 782
- Conditional
  - gdcm::Usage, 847
- cyan
  - gdcm::terminal, 140
- DA
  - gdcm::VR, 865
- DATASET\_FORMAT
  - gdcm::CSAHeader, 273
- DEFAULT
  - gdcm::CryptoFactory, 264
- DES3\_CIPHER
  - gdcm::CryptographicMessageSyntax, 266
- DETECTOR
  - gdcm::Spacing, 702
- DICOMApplicationContextName
  - gdcm::UIDs, 781
- DICOMControlledTerminology
  - gdcm::UIDs, 781
- DICOMUIDRegistry
  - gdcm::UIDs, 781
- DICT\_DEBUG
  - gdcm::DictConverter, 320
- DICT\_DEFAULT
  - gdcm::DictConverter, 320
- DICT\_XML
  - gdcm::DictConverter, 320
- DS
  - gdcm::VR, 866
- DT
  - gdcm::VR, 866
- DeflatedExplicitVRLittleEndian
  - gdcm::TransferSyntax, 766
  - gdcm::UIDs, 779
- DeformableSpatialRegistrationStorage
  - gdcm::UIDs, 783
- DetachedInterpretationManagementSOPClassRetired
  - gdcm::UIDs, 781
- DetachedPatientManagementMetaSOPClassRetired
  - gdcm::UIDs, 781
- DetachedPatientManagementSOPClass
  - gdcm::MediaStorage, 517
- DetachedPatientManagementSOPClassRetired
  - gdcm::UIDs, 781
- DetachedResultsManagementMetaSOPClassRetired
  - gdcm::UIDs, 781
- DetachedResultsManagementSOPClassRetired
  - gdcm::UIDs, 781
- DetachedStudyManagementMetaSOPClassRetired
  - gdcm::UIDs, 781
- DetachedStudyManagementSOPClass
  - gdcm::MediaStorage, 517
- DetachedStudyManagementSOPClassRetired
  - gdcm::UIDs, 781
- DetachedVisitManagementSOPClass
  - gdcm::MediaStorage, 517
- DetachedVisitManagementSOPClassRetired
  - gdcm::UIDs, 781
- DetailSRStorageTrialRetired
  - gdcm::UIDs, 783

- dicomAETitle
  - gdcm::UIDs, [785](#)
- dicomApplicationCluster
  - gdcm::UIDs, [785](#)
- dicomAssociationAcceptor
  - gdcm::UIDs, [785](#)
- dicomAssociationInitiator
  - gdcm::UIDs, [785](#)
- dicomAuthorizedNodeCertificateReference
  - gdcm::UIDs, [785](#)
- dicomConfigurationRoot
  - gdcm::UIDs, [785](#)
- dicomDescription
  - gdcm::UIDs, [785](#)
- dicomDevice
  - gdcm::UIDs, [785](#)
- dicomDeviceName
  - gdcm::UIDs, [785](#)
- dicomDeviceSerialNumber
  - gdcm::UIDs, [785](#)
- dicomDevicesRoot
  - gdcm::UIDs, [785](#)
- dicomHostname
  - gdcm::UIDs, [785](#)
- dicomInstalled
  - gdcm::UIDs, [785](#)
- dicomInstitutionAddress
  - gdcm::UIDs, [785](#)
- dicomInstitutionDepartmentName
  - gdcm::UIDs, [785](#)
- dicomInstitutionName
  - gdcm::UIDs, [785](#)
- dicomIssuerOfPatientID
  - gdcm::UIDs, [785](#)
- dicomManufacturer
  - gdcm::UIDs, [785](#)
- dicomManufacturerModelName
  - gdcm::UIDs, [785](#)
- dicomNetworkAE
  - gdcm::UIDs, [785](#)
- dicomNetworkConnection
  - gdcm::UIDs, [786](#)
- dicomNetworkConnectionReference
  - gdcm::UIDs, [785](#)
- dicomPort
  - gdcm::UIDs, [785](#)
- dicomPreferredCalledAETitle
  - gdcm::UIDs, [785](#)
- dicomPreferredCallingAETitle
  - gdcm::UIDs, [785](#)
- dicomPrimaryDeviceType
  - gdcm::UIDs, [785](#)
- dicomRelatedDeviceReference
  - gdcm::UIDs, [785](#)
- dicomSOPClass
  - gdcm::UIDs, [785](#)
- dicomSoftwareVersion
  - gdcm::UIDs, [785](#)
- dicomStationName
  - gdcm::UIDs, [785](#)
- dicomSupportedCharacterSet
  - gdcm::UIDs, [785](#)
- dicomTLSCyphersuite
  - gdcm::UIDs, [785](#)
- dicomThisNodeCertificateReference
  - gdcm::UIDs, [785](#)
- dicomTransferCapability
  - gdcm::UIDs, [786](#)
- dicomTransferRole
  - gdcm::UIDs, [785](#)
- dicomTransferSyntax
  - gdcm::UIDs, [785](#)
- dicomUniqueAETitle
  - gdcm::UIDs, [786](#)
- dicomUniqueAETitlesRegistryRoot
  - gdcm::UIDs, [785](#)
- dicomVendorData
  - gdcm::UIDs, [785](#)
- DigitalIntraoralXRayImageStorageForPresentation
  - gdcm::UIDs, [782](#)
- DigitalIntraoralXRayImageStorageForProcessing
  - gdcm::MediaStorage, [516](#)
  - gdcm::UIDs, [782](#)
- DigitalIntraoralXrayImageStorageForPresentation
  - gdcm::MediaStorage, [516](#)
- DigitalMammographyImageStorageForPresentation
  - gdcm::MediaStorage, [516](#)
- DigitalMammographyImageStorageForProcessing
  - gdcm::MediaStorage, [516](#)
- DigitalMammographyXRayImageStorageForPresentation
  - gdcm::UIDs, [782](#)
- DigitalMammographyXRayImageStorageForProcessing
  - gdcm::UIDs, [782](#)
- DigitalXRayImageStorageForPresentation
  - gdcm::MediaStorage, [516](#)
  - gdcm::UIDs, [782](#)
- DigitalXRayImageStorageForProcessing
  - gdcm::MediaStorage, [516](#)
  - gdcm::UIDs, [782](#)
- dim
  - gdcm::terminal, [139](#)
- DuplicateAttributeError
  - gdcm::Parser, [563](#)
- eAABORTPDURceivedOpen
  - gdcm::network, [137](#)
- eAABORTRequest
  - gdcm::network, [137](#)

- eAASSOCIATE\_RQPDUreceived
  - gdcmm::network, [137](#)
- eAASSOCIATERequestLocalUser
  - gdcmm::network, [137](#)
- eAASSOCIATEresponseAccept
  - gdcmm::network, [137](#)
- eAASSOCIATEresponseReject
  - gdcmm::network, [137](#)
- eARELEASE\_RPPDUReceived
  - gdcmm::network, [137](#)
- eARELEASE\_RQPDUReceivedOpen
  - gdcmm::network, [137](#)
- eARELEASERequest
  - gdcmm::network, [137](#)
- eARELEASEResponse
  - gdcmm::network, [137](#)
- eARTIMTimerExpired
  - gdcmm::network, [137](#)
- eASSOCIATE\_ACPDUreceived
  - gdcmm::network, [137](#)
- eASSOCIATE\_RJPDUreceived
  - gdcmm::network, [137](#)
- eArabic
  - gdcmm, [127](#)
- eCyrillic
  - gdcmm, [127](#)
- EDGE
  - gdcmm::MeshPrimitive, [526](#)
- eEventDoesNotExist
  - gdcmm::network, [137](#)
- eFind
  - gdcmm, [127](#)
- eGB18030
  - gdcmm, [127](#)
- eGreek
  - gdcmm, [127](#)
- eHebrew
  - gdcmm, [127](#)
- elImage
  - gdcmm, [127](#)
- eJapanese
  - gdcmm, [127](#)
- eJapaneseKanjiMultibyte
  - gdcmm, [127](#)
- eJapaneseSupplementaryKanjiMultibyte
  - gdcmm, [127](#)
- eKoreanHangulHanjaMultibyte
  - gdcmm, [127](#)
- eLatin1
  - gdcmm, [127](#)
- eLatin2
  - gdcmm, [127](#)
- eLatin3
  - gdcmm, [127](#)
- eLatin4
  - gdcmm, [127](#)
- eLatin5
  - gdcmm, [127](#)
- eMove
  - gdcmm, [127](#)
- ePDATATFPDU
  - gdcmm::network, [137](#)
- ePDATArequest
  - gdcmm::network, [137](#)
- ePatient
  - gdcmm, [127](#)
- ePatientRootType
  - gdcmm, [128](#)
- eSeries
  - gdcmm, [127](#)
- eSta10ReleaseCollisionAc
  - gdcmm::network, [138](#)
- eSta11ReleaseCollisionRq
  - gdcmm::network, [138](#)
- eSta12ReleaseCollisionAcLocal
  - gdcmm::network, [138](#)
- eSta13AwaitingClose
  - gdcmm::network, [138](#)
- eSta1Idle
  - gdcmm::network, [137](#)
- eSta2Open
  - gdcmm::network, [137](#)
- eSta3WaitLocalAssoc
  - gdcmm::network, [137](#)
- eSta4LocalAssocDone
  - gdcmm::network, [137](#)
- eSta5WaitRemoteAssoc
  - gdcmm::network, [137](#)
- eSta6TransferReady
  - gdcmm::network, [137](#)
- eSta7WaitRelease
  - gdcmm::network, [137](#)
- eSta8WaitLocalRelease
  - gdcmm::network, [138](#)
- eSta9ReleaseCollisionRqLocal
  - gdcmm::network, [138](#)
- eStaDoesNotExist
  - gdcmm::network, [137](#)
- eStudy
  - gdcmm, [127](#)
- eStudyRootType
  - gdcmm, [128](#)
- eThai
  - gdcmm, [127](#)
- eTransportConnConfirmLocal
  - gdcmm::network, [137](#)
- eTransportConnIndicLocal
  - gdcmm::network, [137](#)

eTransportConnectionClosed  
     gdcm::network, 137  
 eUTF8  
     gdcm, 127  
 eUnrecognizedPDURceived  
     gdcm::network, 137  
 EncapsulatedCDASStorage  
     gdcm::MediaStorage, 517  
     gdcm::UIDs, 784  
 EncapsulatedPDFStorage  
     gdcm::MediaStorage, 517  
     gdcm::UIDs, 784  
 EnhancedCTImageStorage  
     gdcm::MediaStorage, 516  
     gdcm::UIDs, 782  
 EnhancedMRImageStorage  
     gdcm::MediaStorage, 516  
     gdcm::UIDs, 782  
 EnhancedSR  
     gdcm::MediaStorage, 517  
 EnhancedSRStorage  
     gdcm::UIDs, 783  
 EnhancedUSVolumeStorage  
     gdcm::MediaStorage, 518  
     gdcm::UIDs, 786  
 EnhancedXAImageStorage  
     gdcm::MediaStorage, 518  
     gdcm::UIDs, 783  
 EnhancedXRImageStorage  
     gdcm::UIDs, 783  
 Explicit  
     gdcm::TransferSyntax, 766  
 ExplicitVRBigEndian  
     gdcm::TransferSyntax, 766  
     gdcm::UIDs, 779  
 ExplicitVRLittleEndian  
     gdcm::TransferSyntax, 766  
     gdcm::UIDs, 779  
 FACET  
     gdcm::MeshPrimitive, 526  
 FD  
     gdcm::VR, 866  
 FL  
     gdcm::VR, 866  
 FLOAT16  
     gdcm::PixelFormat, 581  
 FLOAT32  
     gdcm::PixelFormat, 581  
 FLOAT64  
     gdcm::PixelFormat, 582  
 FujiPrivateCRImageStorage  
     gdcm::MediaStorage, 518  
 GDCM\_DIFFERENT  
     gdcm, 126  
 GDCM\_EQUAL  
     gdcm, 126  
 GDCM\_GREATER  
     gdcm, 127  
 GDCM\_GREATEROREQUAL  
     gdcm, 127  
 GDCM\_LESS  
     gdcm, 127  
 GDCM\_LESSOREQUAL  
     gdcm, 127  
 GEMS  
     gdcm::Dicts, 326  
 GEPrivate3DModelStorage  
     gdcm::MediaStorage, 517  
 GRAY  
     gdcm::LookupTable, 505  
 GREEN  
     gdcm::LookupTable, 505  
 gdcm, 111  
     backslash, 128  
     eArabic, 127  
     eCyrillic, 127  
     eFind, 127  
     eGB18030, 127  
     eGreek, 127  
     eHebrew, 127  
     eImage, 127  
     eJapanese, 127  
     eJapaneseKanjiMultibyte, 127  
     eJapaneseSupplementaryKanjiMultibyte, 127  
     eKoreanHangulHanjaMultibyte, 127  
     eLatin1, 127  
     eLatin2, 127  
     eLatin3, 127  
     eLatin4, 127  
     eLatin5, 127  
     eMove, 127  
     ePatient, 127  
     ePatientRootType, 128  
     eSeries, 127  
     eStudy, 127  
     eStudyRootType, 128  
     eThai, 127  
     eUTF8, 127  
     GDCM\_DIFFERENT, 126  
     GDCM\_EQUAL, 126  
     GDCM\_GREATER, 127  
     GDCM\_GREATEROREQUAL, 127  
     GDCM\_LESS, 127  
     GDCM\_LESSOREQUAL, 127  
     LD\_ALL, 128  
     LD\_NOSEQ, 128  
     LD\_NOSHADOW, 128

- LD\_NOSHADOWSEQ, [128](#)
- operator!=, [128](#)
- operator<<, [128–132](#)
- operator>>, [132](#)
- operator==, [132](#)
- gdcmm::Attribute
  - VMType, [173](#)
- gdcmm::Attribute< Group, Element, TVR, VM::VM1 >
  - VMType, [180](#)
- gdcmm::CSAHeader
  - DATASET\_FORMAT, [273](#)
  - INTERFILE, [273](#)
  - NOMAGIC, [273](#)
  - SV10, [273](#)
  - UNKNOWN, [273](#)
  - ZEROED\_OUT, [273](#)
- gdcmm::CryptoFactory
  - CAP, [264](#)
  - DEFAULT, [264](#)
  - OPENSSL, [264](#)
  - OPENSSL7, [264](#)
- gdcmm::CryptographicMessageSyntax
  - AES128\_CIPHER, [266](#)
  - AES192\_CIPHER, [266](#)
  - AES256\_CIPHER, [266](#)
  - DES3\_CIPHER, [266](#)
- gdcmm::DictConverter
  - DICT\_DEBUG, [320](#)
  - DICT\_DEFAULT, [320](#)
  - DICT\_XML, [320](#)
- gdcmm::Dicts
  - GEMS, [326](#)
  - PHILIPS, [326](#)
  - SIEMENS, [326](#)
- gdcmm::LookupTable
  - BLUE, [505](#)
  - GRAY, [505](#)
  - GREEN, [505](#)
  - RED, [505](#)
  - UNKNOWN, [505](#)
- gdcmm::MediaStorage
  - AmbulatoryECGWaveformStorage, [517](#)
  - Audio, [518](#)
  - BasicTextSR, [517](#)
  - BasicVoiceAudioWaveformStorage, [517](#)
  - BreastTomosynthesisImageStorage, [518](#)
  - CSANonImageStorage, [517](#)
  - CTImageStorage, [516](#)
  - CardiacElectrophysiologyWaveformStorage, [517](#)
  - ComprehensiveSR, [517](#)
  - ComputedRadiographyImageStorage, [516](#)
  - DetachedPatientManagementSOPClass, [517](#)
  - DetachedStudyManagementSOPClass, [517](#)
  - DetachedVisitManagementSOPClass, [517](#)
  - DigitalIntraoralXRayImageStorageForProcessing, [516](#)
  - DigitalIntraoralXrayImageStorageForPresentation, [516](#)
  - DigitalMammographyImageStorageForPresentation, [516](#)
  - DigitalMammographyImageStorageForProcessing, [516](#)
  - DigitalXRayImageStorageForPresentation, [516](#)
  - DigitalXRayImageStorageForProcessing, [516](#)
  - EncapsulatedCDASStorage, [517](#)
  - EncapsulatedPDFStorage, [517](#)
  - EnhancedCTImageStorage, [516](#)
  - EnhancedMRIImageStorage, [516](#)
  - EnhancedSR, [517](#)
  - EnhancedUSVolumeStorage, [518](#)
  - EnhancedXAImageStorage, [518](#)
  - FujiPrivateCRIImageStorage, [518](#)
  - GEPrivate3DModelStorage, [517](#)
  - GeneralECGWaveformStorage, [517](#)
  - GeneralElectricMagneticResonanceImageStorage, [517](#)
  - GrayscaleSoftcopyPresentationStateStorageSOP↔  
Class, [517](#)
  - HangingProtocolStorage, [518](#)
  - HardcopyGrayscaleImageStorage, [517](#)
  - HemodynamicWaveformStorage, [517](#)
  - KeyObjectSelectionDocument, [517](#)
  - LeadECGWaveformStorage, [517](#)
  - MRIImageStorage, [516](#)
  - MRSpectroscopyStorage, [516](#)
  - MS\_END, [518](#)
  - MammographyCADSR, [517](#)
  - MediaStorageDirectoryStorage, [516](#)
  - ModalityPerformedProcedureStepSOPClass, [518](#)
  - MultiframeGrayscaleByteSecondaryCaptureImage↔  
Storage, [516](#)
  - MultiframeGrayscaleWordSecondaryCapture↔  
ImageStorage, [516](#)
  - MultiframeSingleBitSecondaryCaptureImage↔  
Storage, [516](#)
  - MultiframeTrueColorSecondaryCaptureImage↔  
Storage, [517](#)
  - NoObject, [518](#)
  - NuclearMedicineImageStorage, [517](#)
  - NuclearMedicineImageStorageRetired, [516](#)
  - ObjectEnd, [518](#)
  - OphthalmicPhotography8BitImageStorage, [518](#)
  - OphthalmicTomographyImageStorage, [518](#)
  - PDF, [518](#)
  - PETImageStorage, [517](#)
  - Philips3D, [517](#)
  - PhilipsPrivateMRSyntheticImageStorage, [518](#)
  - RTDoseStorage, [517](#)

- RTImageStorage, [517](#)
- RTIonBeamsTreatmentRecordStorage, [518](#)
- RTIonPlanStorage, [518](#)
- RTPlanStorage, [517](#)
- RTStructureSetStorage, [517](#)
- RTTreatmentSummaryRecordStorage, [518](#)
- RawDataStorage, [517](#)
- SecondaryCaptureImageStorage, [516](#)
- Segmentation, [518](#)
- SegmentationStorage, [518](#)
- SpacialFiducialsStorage, [517](#)
- SpacialRegistrationStorage, [517](#)
- StandaloneCurveStorage, [517](#)
- StandaloneModalityLUTStorage, [517](#)
- StandaloneOverlayStorage, [517](#)
- StandaloneVOILUTStorage, [517](#)
- StudyComponentManagementSOPClass, [517](#)
- SurfaceSegmentationStorage, [518](#)
- ToshibaPrivateDataStorage, [517](#)
- URI, [518](#)
- UltrasoundImageStorage, [516](#)
- UltrasoundImageStorageRetired, [516](#)
- UltrasoundMultiFrameImageStorage, [516](#)
- UltrasoundMultiFrameImageStorageRetired, [516](#)
- VLEndoscopicImageStorage, [518](#)
- VLMicroscopicImageStorage, [518](#)
- VLPhotographicImageStorage, [518](#)
- VLWholeSlideMicroscopyImageStorage, [518](#)
- Video, [518](#)
- VideoEndoscopicImageStorage, [517](#)
- Waveform, [518](#)
- XRay3DAngiographicImageStorage, [518](#)
- XRayAngiographicBiPlaneImageStorageRetired, [517](#)
- XRayAngiographicImageStorage, [517](#)
- XRayRadiationDoseSR, [518](#)
- XRayRadiofluoroscopicImageStorage, [517](#)
- gdcmmesh::MeshPrimitive
  - EDGE, [526](#)
  - FACET, [526](#)
  - LINE, [526](#)
  - MPTYPE\_END, [526](#)
  - TRIANGLE, [526](#)
  - TRIANGLE\_FAN, [526](#)
  - TRIANGLE\_STRIP, [526](#)
  - VERTEX, [526](#)
- gdcmmath::Orientation
  - AXIAL, [553](#)
  - CORONAL, [553](#)
  - OBLIQUE, [553](#)
  - SAGITTAL, [553](#)
  - UNKNOWN, [553](#)
- gdcmmath::Overlay
  - Graphics, [557](#)
  - Invalid, [557](#)
  - ROI, [557](#)
- gdcm::Parser
  - DuplicateAttributeError, [563](#)
  - JunkAfterDocElementError, [563](#)
  - NoElementsError, [563](#)
  - NoError, [563](#)
  - NoMemoryError, [563](#)
  - SyntaxError, [563](#)
  - TagMismatchError, [563](#)
  - UndefinedEntityError, [563](#)
  - UnexpectedStateError, [563](#)
- gdcm::PhotometricInterpretation
  - ARGB, [578](#)
  - CMYK, [578](#)
  - HSV, [578](#)
  - MONOCHROME1, [578](#)
  - MONOCHROME2, [578](#)
  - PALETTE\_COLOR, [578](#)
  - PI\_END, [578](#)
  - RGB, [578](#)
  - UNKNOWN, [578](#)
  - YBR\_FULL, [578](#)
  - YBR\_FULL\_422, [578](#)
  - YBR\_ICT, [578](#)
  - YBR\_PARTIAL\_420, [578](#)
  - YBR\_PARTIAL\_422, [578](#)
  - YBR\_RCT, [578](#)
- gdcm::PixelFormat
  - FLOAT16, [581](#)
  - FLOAT32, [581](#)
  - FLOAT64, [582](#)
  - INT12, [581](#)
  - INT16, [581](#)
  - INT32, [581](#)
  - INT8, [581](#)
  - SINGLEBIT, [582](#)
  - UINT12, [581](#)
  - UINT16, [581](#)
  - UINT32, [581](#)
  - UINT8, [581](#)
  - UNKNOWN, [582](#)
- gdcm::Printer
  - CONDENSED\_STYLE, [611](#)
  - VERBOSE\_STYLE, [611](#)
  - XML, [611](#)
- gdcm::STATIC\_ASSERTION\_FAILURE< true >
  - value, [706](#)
- gdcm::Segment
  - ALGOType\_END, [658](#)
  - AUTOMATIC, [658](#)
  - MANUAL, [658](#)
- gdcm::Spacing
  - CALIBRATED, [702](#)
  - DETECTOR, [702](#)

- MAGNIFIED, [702](#)
- UNKNOWN, [702](#)
- gdcmm::Surface
  - NO, [726](#)
  - POINTS, [726](#)
  - STATES\_END, [726](#)
  - SURFACE, [726](#)
  - UNKNOWN, [726](#)
  - VIEWType\_END, [726](#)
  - WIREFRAME, [726](#)
  - YES, [726](#)
- gdcmm::SwapCode
  - BadBigEndian, [737](#)
  - BadLittleEndian, [737](#)
  - BigEndian, [737](#)
  - LittleEndian, [737](#)
  - Unknown, [737](#)
- gdcmm::TransferSyntax
  - CT\_private\_ELE, [766](#)
  - DeflatedExplicitVRLittleEndian, [766](#)
  - Explicit, [766](#)
  - ExplicitVRBigEndian, [766](#)
  - ExplicitVRLittleEndian, [766](#)
  - Implicit, [766](#)
  - ImplicitVRBigEndianACRNEMA, [766](#)
  - ImplicitVRBigEndianPrivateGE, [766](#)
  - ImplicitVRLittleEndian, [766](#)
  - JPEG2000, [766](#)
  - JPEG2000Lossless, [766](#)
  - JPEG2000Part2, [766](#)
  - JPEG2000Part2Lossless, [766](#)
  - JPEGBaselineProcess1, [766](#)
  - JPEGExtendedProcess2\_4, [766](#)
  - JPEGExtendedProcess3\_5, [766](#)
  - JPEGFullProgressionProcess10\_12, [766](#)
  - JPEGLSLossless, [766](#)
  - JPEGLSNearLossless, [766](#)
  - JPEGLosslessProcess14, [766](#)
  - JPEGLosslessProcess14\_1, [766](#)
  - JPEGSpectralSelectionProcess6\_8, [766](#)
  - JPIPRendered, [766](#)
  - MPEG2MainProfile, [766](#)
  - RLELossless, [766](#)
  - TS\_END, [766](#)
  - Unknown, [766](#)
- gdcmm::Type
  - T1, [771](#)
  - T1C, [771](#)
  - T2, [771](#)
  - T2C, [771](#)
  - T3, [771](#)
  - UNKNOWN, [771](#)
- gdcmm::UIDs
  - AmbulatoryECGWaveformStorage, [782](#)
  - AudioSRStorageTrialRetired, [783](#)
  - BasicAnnotationBoxSOPClass, [781](#)
  - BasicColorImageBoxSOPClass, [781](#)
  - BasicColorPrintManagementMetaSOPClass, [781](#)
  - BasicFilmBoxSOPClass, [781](#)
  - BasicFilmSessionSOPClass, [781](#)
  - BasicGrayscaleImageBoxSOPClass, [781](#)
  - BasicGrayscalePrintManagementMetaSOPClass, [781](#)
  - BasicPrintImageOverlayBoxSOPClassRetired, [782](#)
  - BasicStudyContentNotificationSOPClassRetired, [781](#)
  - BasicTextSRStorage, [783](#)
  - BasicVoiceAudioWaveformStorage, [782](#)
  - BlendingSoftcopyPresentationStateStorageSOPClass, [783](#)
  - BreastImagingRelevantPatientInformationQuery, [784](#)
  - BreastTomosynthesisImageStorage, [786](#)
  - CTImageStorage, [782](#)
  - CardiacElectrophysiologyWaveformStorage, [782](#)
  - CardiacRelevantPatientInformationQuery, [785](#)
  - ChestCADSRStorage, [784](#)
  - ColorSoftcopyPresentationStateStorageSOPClass, [783](#)
  - ComprehensiveSRStorage, [783](#)
  - ComprehensiveSRStorageTrialRetired, [783](#)
  - ComputedRadiographyImageStorage, [782](#)
  - DICOMApplicationContextName, [781](#)
  - DICOMControlledTerminology, [781](#)
  - DICOMUIDRegistry, [781](#)
  - DeflatedExplicitVRLittleEndian, [779](#)
  - DeformableSpatialRegistrationStorage, [783](#)
  - DetachedInterpretationManagementSOPClassRetired, [781](#)
  - DetachedPatientManagementMetaSOPClassRetired, [781](#)
  - DetachedPatientManagementSOPClassRetired, [781](#)
  - DetachedResultsManagementMetaSOPClassRetired, [781](#)
  - DetachedResultsManagementSOPClassRetired, [781](#)
  - DetachedStudyManagementMetaSOPClassRetired, [781](#)
  - DetachedStudyManagementSOPClassRetired, [781](#)
  - DetachedVisitManagementSOPClassRetired, [781](#)
  - DetailSRStorageTrialRetired, [783](#)
  - dicomAETitle, [785](#)
  - dicomApplicationCluster, [785](#)
  - dicomAssociationAcceptor, [785](#)
  - dicomAssociationInitiator, [785](#)
  - dicomAuthorizedNodeCertificateReference, [785](#)
  - dicomConfigurationRoot, [785](#)
  - dicomDescription, [785](#)
  - dicomDevice, [785](#)



- dicomDeviceName, [785](#)
- dicomDeviceSerialNumber, [785](#)
- dicomDevicesRoot, [785](#)
- dicomHostname, [785](#)
- dicomInstalled, [785](#)
- dicomInstitutionAddress, [785](#)
- dicomInstitutionDepartmentName, [785](#)
- dicomInstitutionName, [785](#)
- dicomIssuerOfPatientID, [785](#)
- dicomManufacturer, [785](#)
- dicomManufacturerModelName, [785](#)
- dicomNetworkAE, [785](#)
- dicomNetworkConnection, [786](#)
- dicomNetworkConnectionReference, [785](#)
- dicomPort, [785](#)
- dicomPreferredCalledAETitle, [785](#)
- dicomPreferredCallingAETitle, [785](#)
- dicomPrimaryDeviceType, [785](#)
- dicomRelatedDeviceReference, [785](#)
- dicomSOPClass, [785](#)
- dicomSoftwareVersion, [785](#)
- dicomStationName, [785](#)
- dicomSupportedCharacterSet, [785](#)
- dicomTLSCyphersuite, [785](#)
- dicomThisNodeCertificateReference, [785](#)
- dicomTransferCapability, [786](#)
- dicomTransferRole, [785](#)
- dicomTransferSyntax, [785](#)
- dicomUniqueAETitle, [786](#)
- dicomUniqueAETitlesRegistryRoot, [785](#)
- dicomVendorData, [785](#)
- DigitalIntraoralXRayImageStorageForPresentation, [782](#)
- DigitalIntraoralXRayImageStorageForProcessing, [782](#)
- DigitalMammographyXRayImageStorageForPresentation, [782](#)
- DigitalMammographyXRayImageStorageForProcessing, [782](#)
- DigitalXRayImageStorageForPresentation, [782](#)
- DigitalXRayImageStorageForProcessing, [782](#)
- EncapsulatedCDASStorage, [784](#)
- EncapsulatedPDFStorage, [784](#)
- EnhancedCTImageStorage, [782](#)
- EnhancedMRIImageStorage, [782](#)
- EnhancedSRStorage, [783](#)
- EnhancedUSVolumeStorage, [786](#)
- EnhancedXAImageStorage, [783](#)
- EnhancedXRFImageStorage, [783](#)
- ExplicitVRBigEndian, [779](#)
- ExplicitVRLittleEndian, [779](#)
- GeneralECGWaveformStorage, [782](#)
- GeneralPurposePerformedProcedureStepSOPClass, [784](#)
- GeneralPurposeScheduledProcedureStepSOPClass, [784](#)
- GeneralPurposeWorklistInformationModelFIND, [784](#)
- GeneralPurposeWorklistManagementMetaSOPClass, [784](#)
- GeneralRelevantPatientInformationQuery, [784](#)
- GrayscaleSoftcopyPresentationStateStorageSOPClass, [783](#)
- HangingProtocolInformationModelFIND, [785](#)
- HangingProtocolInformationModelMOVE, [785](#)
- HangingProtocolStorage, [785](#)
- HardcopyColorImageStorageSOPClassRetired, [782](#)
- HardcopyGrayscaleImageStorageSOPClassRetired, [782](#)
- HemodynamicWaveformStorage, [782](#)
- ICBM452T1FrameofReference, [781](#)
- ICBMSingleSubjectMRIFrameofReference, [781](#)
- ImageOverlayBoxSOPClassRetired, [782](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM, [779](#)
- InstanceAvailabilityNotificationSOPClass, [784](#)
- JPEG2000ImageCompression, [780](#)
- JPEG2000ImageCompressionLosslessOnly, [780](#)
- JPEG2000Part2MulticomponentImageCompression, [780](#)
- JPEG2000Part2MulticomponentImageCompressionLosslessOnly, [780](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJPEG8BitImageCompression, [779](#)
- JPEGExtendedHierarchicalProcess1618Retired, [780](#)
- JPEGExtendedHierarchicalProcess1719Retired, [780](#)
- JPEGExtendedProcess24DefaultTransferSyntaxforLossyJPEG12BitImageCompressionProcess4only, [779](#)
- JPEGExtendedProcess35Retired, [779](#)
- JPEGFULLProgressionHierarchicalProcess2426Retired, [780](#)
- JPEGFULLProgressionHierarchicalProcess2527Retired, [780](#)
- JPEGFULLProgressionNonHierarchicalProcess1012Retired, [779](#)
- JPEGFULLProgressionNonHierarchicalProcess1113Retired, [779](#)
- JPEGLSLosslessImageCompression, [780](#)
- JPEGLSLossyNearLosslessImageCompression, [780](#)
- JPEGLosslessHierarchicalProcess28Retired, [780](#)
- JPEGLosslessHierarchicalProcess29Retired, [780](#)
- JPEGLosslessNonHierarchicalFirstOrderPredictionProcess14SelectionValue1DefaultTransferSyntaxforLosslessJPEGImageCompression, [780](#)
- JPEGLosslessNonHierarchicalProcess14, [779](#)
- JPEGLosslessNonHierarchicalProcess15Retired, [780](#)



- JPEGSpectralSelectionHierarchicalProcess2022↔  
Retired, [780](#)
- JPEGSpectralSelectionHierarchicalProcess2123↔  
Retired, [780](#)
- JPEGSpectralSelectionNonHierarchicalProcess68↔  
Retired, [779](#)
- JPEGSpectralSelectionNonHierarchicalProcess79↔  
Retired, [779](#)
- JPIPReterenced, [780](#)
- JPIPReterencedDeflate, [780](#)
- KeyObjectSelectionDocumentStorage, [784](#)
- MPEG2MainProfileMainLevel, [780](#)
- MRImageStorage, [782](#)
- MRSpectroscopyStorage, [782](#)
- MammographyCADSRStorage, [783](#)
- MediaCreationManagementSOPClassUID, [782](#)
- MediaStorageDirectoryStorage, [780](#)
- ModalityPerformedProcedureStepNotificationSOP↔  
Class, [781](#)
- ModalityPerformedProcedureStepRetrieveSOP↔  
Class, [781](#)
- ModalityPerformedProcedureStepSOPClass, [781](#)
- ModalityWorklistInformationModelFIND, [784](#)
- MultiframeGrayscaleByteSecondaryCaptureImage↔  
Storage, [782](#)
- MultiframeGrayscaleWordSecondaryCapture↔  
ImageStorage, [782](#)
- MultiframeSingleBitSecondaryCaptureImage↔  
Storage, [782](#)
- MultiframeTrueColorSecondaryCaptureImage↔  
Storage, [782](#)
- NuclearMedicineImageStorage, [783](#)
- NuclearMedicineImageStorageRetired, [782](#)
- OphthalmicPhotography16BitImageStorage, [783](#)
- OphthalmicPhotography8BitImageStorage, [783](#)
- OphthalmicTomographyImageStorage, [783](#)
- PatientRootQueryRetrieveInformationModelFIND,  
[784](#)
- PatientRootQueryRetrieveInformationModelGET,  
[784](#)
- PatientRootQueryRetrieveInformationModelMOVE,  
[784](#)
- PatientStudyOnlyQueryRetrieveInformationModelF↔  
INDRetired, [784](#)
- PatientStudyOnlyQueryRetrieveInformationModel↔  
GETRetired, [784](#)
- PatientStudyOnlyQueryRetrieveInformationModel↔  
MOVERetired, [784](#)
- PositronEmissionTomographyImageStorage, [784](#)
- PresentationLUTSOPClass, [782](#)
- PrintJobSOPClass, [781](#)
- PrintQueueManagementSOPClassRetired, [782](#)
- PrintQueueSOPInstanceRetired, [782](#)
- PrinterConfigurationRetrievalSOPClass, [781](#)
- PrinterSOPClass, [781](#)
- PrinterSOPInstance, [781](#)
- ProceduralEventLoggingSOPClass, [781](#)
- ProceduralEventLoggingSOPInstance, [781](#)
- ProcedureLogStorage, [783](#)
- ProductCharacteristicsQuerySOPClass, [785](#)
- PseudoColorSoftcopyPresentationStateStorageSO↔  
PClass, [783](#)
- PullPrintRequestSOPClassRetired, [782](#)
- PullStoredPrintManagementMetaSOPClassRetired,  
[782](#)
- RFC2557MIMEencapsulation, [780](#)
- RLELossless, [780](#)
- RTBeamsDeliveryInstructionStorageSupplement74↔  
FrozenDraft, [784](#)
- RTBeamsTreatmentRecordStorage, [784](#)
- RTBrachyTreatmentRecordStorage, [784](#)
- RTConventionalMachineVerificationSupplement74↔  
FrozenDraft, [784](#)
- RTDoseStorage, [784](#)
- RTImageStorage, [784](#)
- RTIonBeamsTreatmentRecordStorage, [784](#)
- RTIonMachineVerificationSupplement74FrozenDraft,  
[784](#)
- RTIonPlanStorage, [784](#)
- RTPlanStorage, [784](#)
- RTStructureSetStorage, [784](#)
- RTTreatmentSummaryRecordStorage, [784](#)
- RawDataStorage, [783](#)
- RealWorldValueMappingStorage, [783](#)
- ReferencedColorPrintManagementMetaSOPClass↔  
Retired, [781](#)
- ReferencedGrayscalePrintManagementMetaSOP↔  
ClassRetired, [781](#)
- ReferencedImageBoxSOPClassRetired, [781](#)
- SPM2AVG152PDFFrameofReference, [780](#)
- SPM2AVG152T1FrameofReference, [780](#)
- SPM2AVG152T2FrameofReference, [780](#)
- SPM2AVG305T1FrameofReference, [780](#)
- SPM2BRAINMASKFrameofReference, [780](#)
- SPM2CSFFFrameofReference, [780](#)
- SPM2EPIFrameofReference, [780](#)
- SPM2FILT1FrameofReference, [780](#)
- SPM2GRAYFrameofReference, [780](#)
- SPM2PDFFrameofReference, [780](#)
- SPM2PETFrameofReference, [780](#)
- SPM2SINGLESUBJT1FrameofReference, [780](#)
- SPM2SPECTFrameofReference, [780](#)
- SPM2T1FrameofReference, [780](#)
- SPM2T2FrameofReference, [780](#)
- SPM2TRANSMFrameofReference, [780](#)
- SPM2WHITEFrameofReference, [780](#)
- SecondaryCaptureImageStorage, [782](#)

SegmentationStorage, [783](#)  
 SpatialFiducialsStorage, [783](#)  
 SpatialRegistrationStorage, [783](#)  
 StandaloneCurveStorageRetired, [782](#)  
 StandaloneModalityLUTStorageRetired, [783](#)  
 StandaloneOverlayStorageRetired, [782](#)  
 StandalonePETCurveStorageRetired, [784](#)  
 StandaloneVOILUTStorageRetired, [783](#)  
 StereometricRelationshipStorage, [783](#)  
 StorageCommitmentPullModelSOPClassRetired, [781](#)  
 StorageCommitmentPullModelSOPInstanceRetired, [781](#)  
 StorageCommitmentPushModelSOPClass, [781](#)  
 StorageCommitmentPushModelSOPInstance, [781](#)  
 StorageServiceClass, [781](#)  
 StoredPrintStorageSOPClassRetired, [782](#)  
 StudyComponentManagementSOPClassRetired, [781](#)  
 StudyRootQueryRetrieveInformationModelFIND, [784](#)  
 StudyRootQueryRetrieveInformationModelGET, [784](#)  
 StudyRootQueryRetrieveInformationModelMOVE, [784](#)  
 SubstanceAdministrationLoggingSOPClass, [781](#)  
 SubstanceAdministrationLoggingSOPInstance, [781](#)  
 SubstanceApprovalQuerySOPClass, [785](#)  
 SurfaceSegmentationStorage, [786](#)  
 TalairachBrainAtlasFrameofReference, [780](#)  
 TextSRStorageTrialRetired, [783](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_1, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_10, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_11, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_12, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_13, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_14, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_15, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_16, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_17, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_18, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_19, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_2, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_20, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_21, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_22, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_23, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_24, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_25, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_26, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_27, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_28, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_29, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_3, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_30, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_31, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_4, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_5, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_6, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_7, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_8, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_3\_9, [791](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_1, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_2, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_3, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_4, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_5, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_6, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_7, [792](#)  
 uid\_1\_2\_840\_10008\_15\_0\_4\_8, [792](#)  
 uid\_1\_2\_840\_10008\_1\_1, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2, [786](#)  
 uid\_1\_2\_840\_10008\_1\_20\_1, [787](#)  
 uid\_1\_2\_840\_10008\_1\_20\_1\_1, [787](#)  
 uid\_1\_2\_840\_10008\_1\_20\_2, [787](#)  
 uid\_1\_2\_840\_10008\_1\_20\_2\_1, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_1, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_1\_99, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_2, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_100, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_50, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_51, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_52, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_53, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_54, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_55, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_56, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_57, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_58, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_59, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_60, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_61, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_62, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_63, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_64, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_65, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_66, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_70, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_80, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_81, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_90, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_91, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_92, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_93, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_94, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_95, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_5, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_6\_1, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_6\_2, [787](#)  
 uid\_1\_2\_840\_10008\_1\_3\_10, [787](#)

uid\_1\_2\_840\_10008\_1\_40, [787](#)  
uid\_1\_2\_840\_10008\_1\_40\_1, [787](#)  
uid\_1\_2\_840\_10008\_1\_42, [787](#)  
uid\_1\_2\_840\_10008\_1\_42\_1, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_1, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_10, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_11, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_12, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_13, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_14, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_15, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_16, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_17, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_18, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_2, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_3, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_4, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_5, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_6, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_7, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_8, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_1\_9, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_1, [787](#)  
uid\_1\_2\_840\_10008\_1\_4\_2\_2, [787](#)  
uid\_1\_2\_840\_10008\_1\_9, [787](#)  
uid\_1\_2\_840\_10008\_2\_16\_4, [787](#)  
uid\_1\_2\_840\_10008\_2\_6\_1, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_1\_1, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1, [787](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5, [788](#)  
uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1, [788](#)  
uid\_1\_2\_840\_10008\_4\_2, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_14, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_15, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_2, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_22, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_23, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_25, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_26, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_27, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_29, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_30, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_31, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_32, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_33, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3, [792](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_48\_2, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9, [790](#)

[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_4\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_4\\_2, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_5, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_6, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66\\_2, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66\\_3, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66\\_4, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_66\\_5, 792](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_67, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_6\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_6\\_2, 792](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_7, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_1\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_2, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_2\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_3, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_4, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_4\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_5\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_5\\_2, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_5\\_3, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_5\\_4, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_1\\_6, 792](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_77\\_2, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_7\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_7\\_2, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_7\\_3, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_7\\_4, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_8, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_11, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_2, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_22, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_3, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_33, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_4, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_40, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_50, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_59, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_65, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_88\\_67, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_1\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_1\\_2, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_1\\_3, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_2\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_3\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_1\\_9\\_4\\_1, 789](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_1\\_1, 790](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_1\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_1\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_2\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_2\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_2\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_3\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_3\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_1\\_2\\_3\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_31, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_32, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_32\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_32\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_32\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_33, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_4, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_4\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_4\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_4\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_4\\_4, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_34\\_5, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_37\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_37\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_37\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_38\\_1, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_38\\_2, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_38\\_3, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_41, 791](#)  
[uid\\_1\\_2\\_840\\_10008\\_5\\_1\\_4\\_42, 791](#)  
[UltrasoundImageStorage, 782](#)  
[UltrasoundImageStorageRetired, 782](#)  
[UltrasoundMultiframeImageStorage, 782](#)  
[UltrasoundMultiframeImageStorageRetired, 782](#)  
[UnifiedProcedureStepEventSOPClass, 784](#)  
[UnifiedProcedureStepPullSOPClass, 784](#)  
[UnifiedProcedureStepPushSOPClass, 784](#)  
[UnifiedProcedureStepWatchSOPClass, 784](#)  
[UnifiedWorklistandProcedureStepSOPInstance, 784](#)  
[UnifiedWorklistandProcedureStepServiceClass, 784](#)  
[VLEndoscopicImageStorage, 783](#)  
[VLImageStorageTrialRetired, 783](#)  
[VLMicroscopicImageStorage, 783](#)  
[VLMultiframeImageStorageTrialRetired, 783](#)  
[VLPhotographicImageStorage, 783](#)  
[VLSlideCoordinatesMicroscopicImageStorage, 783](#)  
[VLWholeSlideMicroscopyImageStorage, 786](#)  
[VOILUTBoxSOPClass, 782](#)  
[VerificationSOPClass, 779](#)  
[VideoEndoscopicImageStorage, 783](#)  
[VideoMicroscopicImageStorage, 783](#)  
[VideoPhotographicImageStorage, 783](#)  
[WaveformStorageTrialRetired, 782](#)

- XMLEncoding, [780](#)
- XRay3DAngiographicImageStorage, [783](#)
- XRay3DCraniofacialImageStorage, [783](#)
- XRayAngiographicBiPlaneImageStorageRetired, [783](#)
- XRayAngiographicImageStorage, [783](#)
- XRayRadiationDoseSRStorage, [784](#)
- XRayRadiofluoroscopicImageStorage, [783](#)
- gdcmm::Usage
  - Conditional, [847](#)
  - Invalid, [847](#)
  - Mandatory, [847](#)
  - UserOption, [847](#)
- gdcmm::VM
  - VM0, [861](#)
  - VM1, [861](#)
  - VM10, [861](#)
  - VM12, [861](#)
  - VM16, [861](#)
  - VM18, [861](#)
  - VM1\_2, [862](#)
  - VM1\_3, [862](#)
  - VM1\_32, [862](#)
  - VM1\_4, [862](#)
  - VM1\_5, [862](#)
  - VM1\_8, [862](#)
  - VM1\_99, [862](#)
  - VM1\_n, [862](#)
  - VM2, [861](#)
  - VM24, [861](#)
  - VM256, [862](#)
  - VM28, [861](#)
  - VM2\_2n, [862](#)
  - VM2\_n, [862](#)
  - VM3, [861](#)
  - VM30\_30n, [862](#)
  - VM32, [861](#)
  - VM35, [861](#)
  - VM3\_3n, [862](#)
  - VM3\_4, [862](#)
  - VM3\_n, [862](#)
  - VM4, [861](#)
  - VM47\_47n, [862](#)
  - VM4\_4n, [862](#)
  - VM5, [861](#)
  - VM6, [861](#)
  - VM6\_6n, [862](#)
  - VM7\_7n, [862](#)
  - VM8, [861](#)
  - VM9, [861](#)
  - VM99, [862](#)
  - VM\_END, [862](#)
- gdcmm::VR
  - AE, [865](#)
  - AS, [865](#)
  - AT, [865](#)
  - CS, [865](#)
  - DA, [865](#)
  - DS, [866](#)
  - DT, [866](#)
  - FD, [866](#)
  - FL, [866](#)
  - INVALID, [865](#)
  - IS, [866](#)
  - LO, [866](#)
  - LT, [866](#)
  - OB, [866](#)
  - OB\_OW, [866](#)
  - OD, [866](#)
  - OF, [866](#)
  - OW, [866](#)
  - PN, [866](#)
  - SH, [866](#)
  - SL, [866](#)
  - SQ, [866](#)
  - SS, [866](#)
  - ST, [866](#)
  - TM, [866](#)
  - UI, [866](#)
  - UL, [866](#)
  - UN, [866](#)
  - US, [866](#)
  - US\_SS, [866](#)
  - US\_SS\_OW, [866](#)
  - UT, [866](#)
  - VL16, [866](#)
  - VL32, [866](#)
  - VR\_END, [866](#)
  - VR\_VM1, [866](#)
  - VRALL, [866](#)
  - VRASCII, [866](#)
  - VRBINARY, [866](#)
- gdcmm::XMLPrinter
  - LOADBULKDATA, [937](#)
  - OnlyUUID, [937](#)
- gdcmm::network
  - eAABORTPDUReturnedOpen, [137](#)
  - eAABORTRequest, [137](#)
  - eAASSOCIATE\_RQPDUReturned, [137](#)
  - eAASSOCIATERequestLocalUser, [137](#)
  - eAASSOCIATEResponseAccept, [137](#)
  - eAASSOCIATEResponseReject, [137](#)
  - eARELEASE\_RPPDUReturned, [137](#)
  - eARELEASE\_RQPDUReturnedOpen, [137](#)
  - eARELEASERequest, [137](#)
  - eARELEASEResponse, [137](#)
  - eARTIMTimerExpired, [137](#)
  - eASSOCIATE\_ACPDUReturned, [137](#)
  - eASSOCIATE\_RJPDUReturned, [137](#)

- eEventDoesNotExist, 137
- ePDATATFPDU, 137
- ePDATArequest, 137
- eSta10ReleaseCollisionAc, 138
- eSta11ReleaseCollisionRq, 138
- eSta12ReleaseCollisionAcLocal, 138
- eSta13AwaitingClose, 138
- eSta1Idle, 137
- eSta2Open, 137
- eSta3WaitLocalAssoc, 137
- eSta4LocalAssocDone, 137
- eSta5WaitRemoteAssoc, 137
- eSta6TransferReady, 137
- eSta7WaitRelease, 137
- eSta8WaitLocalRelease, 138
- eSta9ReleaseCollisionRqLocal, 138
- eStaDoesNotExist, 137
- eTransportConnConfirmLocal, 137
- eTransportConnIndicLocal, 137
- eTransportConnectionClosed, 137
- eUnrecognizedPDUReceived, 137
- gdcm::network::DIMSE
  - C\_CANCEL\_RQ, 329
  - C\_ECHO\_RQ, 329
  - C\_ECHO\_RSP, 329
  - C\_FIND\_RQ, 328
  - C\_FIND\_RSP, 328
  - C\_GET\_RQ, 328
  - C\_GET\_RSP, 328
  - C\_MOVE\_RQ, 328
  - C\_MOVE\_RSP, 329
  - C\_STORE\_RQ, 328
  - C\_STORE\_RSP, 328
  - N\_ACTION\_RQ, 329
  - N\_ACTION\_RSP, 329
  - N\_CREATE\_RQ, 329
  - N\_CREATE\_RSP, 329
  - N\_DELETE\_RQ, 329
  - N\_DELETE\_RSP, 329
  - N\_EVENT\_REPORT\_RQ, 329
  - N\_EVENT\_REPORT\_RSP, 329
  - N\_GET\_RQ, 329
  - N\_GET\_RSP, 329
  - N\_SET\_RQ, 329
  - N\_SET\_RSP, 329
- gdcm::terminal
  - black, 139
  - blink, 139
  - blue, 140
  - bright, 139
  - CONSOLE, 140
  - cyan, 140
  - dim, 139
  - green, 139
  - hidden, 139
  - magenta, 140
  - red, 139
  - reset, 139
  - reverse, 139
  - underline, 139
  - VT100, 140
  - white, 140
  - yellow, 139
- GeneralECGWaveformStorage
  - gdcm::MediaStorage, 517
  - gdcm::UIDs, 782
- GeneralElectricMagneticResonanceImageStorage
  - gdcm::MediaStorage, 517
- GeneralPurposePerformedProcedureStepSOPClass
  - gdcm::UIDs, 784
- GeneralPurposeScheduledProcedureStepSOPClass
  - gdcm::UIDs, 784
- GeneralPurposeWorklistInformationModelFIND
  - gdcm::UIDs, 784
- GeneralPurposeWorklistManagementMetaSOPClass
  - gdcm::UIDs, 784
- GeneralRelevantPatientInformationQuery
  - gdcm::UIDs, 784
- Graphics
  - gdcm::Overlay, 557
- GrayscaleSoftcopyPresentationStateStorageSOPClass
  - gdcm::MediaStorage, 517
  - gdcm::UIDs, 783
- green
  - gdcm::terminal, 139
- HSV
  - gdcm::PhotometricInterpretation, 578
- HangingProtocolInformationModelFIND
  - gdcm::UIDs, 785
- HangingProtocolInformationModelMOVE
  - gdcm::UIDs, 785
- HangingProtocolStorage
  - gdcm::MediaStorage, 518
  - gdcm::UIDs, 785
- HardcopyColorImageStorageSOPClassRetired
  - gdcm::UIDs, 782
- HardcopyGrayscaleImageStorage
  - gdcm::MediaStorage, 517
- HardcopyGrayscaleImageStorageSOPClassRetired
  - gdcm::UIDs, 782
- HemodynamicWaveformStorage
  - gdcm::MediaStorage, 517
  - gdcm::UIDs, 782
- hidden
  - gdcm::terminal, 139
- ICBM452T1FrameofReference
  - gdcm::UIDs, 781



- ICBMSingleSubjectMRIFrameofReference
  - gdcm::UIDs, [781](#)
- INT12
  - gdcm::PixelFormat, [581](#)
- INT16
  - gdcm::PixelFormat, [581](#)
- INT32
  - gdcm::PixelFormat, [581](#)
- INT8
  - gdcm::PixelFormat, [581](#)
- INTERFILE
  - gdcm::CSAHeader, [273](#)
- INVALID
  - gdcm::VR, [865](#)
- IS
  - gdcm::VR, [866](#)
- ImageOverlayBoxSOPClassRetired
  - gdcm::UIDs, [782](#)
- Implicit
  - gdcm::TransferSyntax, [766](#)
- ImplicitVRBigEndianACRNEMA
  - gdcm::TransferSyntax, [766](#)
- ImplicitVRBigEndianPrivateGE
  - gdcm::TransferSyntax, [766](#)
- ImplicitVRLittleEndian
  - gdcm::TransferSyntax, [766](#)
- ImplicitVRLittleEndianDefaultTransferSyntaxforDICOM
  - gdcm::UIDs, [779](#)
- InstanceAvailabilityNotificationSOPClass
  - gdcm::UIDs, [784](#)
- Invalid
  - gdcm::Overlay, [557](#)
  - gdcm::Usage, [847](#)
- JPEG2000
  - gdcm::TransferSyntax, [766](#)
- JPEG2000\_COMPRESSION
  - vtkGDCMImageWriter, [886](#)
- JPEG2000ImageCompression
  - gdcm::UIDs, [780](#)
- JPEG2000ImageCompressionLosslessOnly
  - gdcm::UIDs, [780](#)
- JPEG2000Lossless
  - gdcm::TransferSyntax, [766](#)
- JPEG2000Part2
  - gdcm::TransferSyntax, [766](#)
- JPEG2000Part2Lossless
  - gdcm::TransferSyntax, [766](#)
- JPEG2000Part2MulticomponentImageCompression
  - gdcm::UIDs, [780](#)
- JPEG2000Part2MulticomponentImageCompression↔
  - LosslessOnly
  - gdcm::UIDs, [780](#)
- JPEG\_COMPRESSION
  - vtkGDCMImageWriter, [886](#)
- JPEGBaselineProcess1
  - gdcm::TransferSyntax, [766](#)
- JPEGBaselineProcess1DefaultTransferSyntaxforLossyJ↔
  - PEG8BitImageCompression
  - gdcm::UIDs, [779](#)
- JPEGExtendedHierarchicalProcess1618Retired
  - gdcm::UIDs, [780](#)
- JPEGExtendedHierarchicalProcess1719Retired
  - gdcm::UIDs, [780](#)
- JPEGExtendedProcess24DefaultTransferSyntaxfor↔
  - LossyJPEG12BitImageCompressionProcess4only
  - gdcm::UIDs, [779](#)
- JPEGExtendedProcess2\_4
  - gdcm::TransferSyntax, [766](#)
- JPEGExtendedProcess35Retired
  - gdcm::UIDs, [779](#)
- JPEGExtendedProcess3\_5
  - gdcm::TransferSyntax, [766](#)
- JPEGFullProgressionHierarchicalProcess2426Retired
  - gdcm::UIDs, [780](#)
- JPEGFullProgressionHierarchicalProcess2527Retired
  - gdcm::UIDs, [780](#)
- JPEGFullProgressionNonHierarchicalProcess1012↔
  - Retired
  - gdcm::UIDs, [779](#)
- JPEGFullProgressionNonHierarchicalProcess1113↔
  - Retired
  - gdcm::UIDs, [779](#)
- JPEGFullProgressionProcess10\_12
  - gdcm::TransferSyntax, [766](#)
- JPEGLS\_COMPRESSION
  - vtkGDCMImageWriter, [886](#)
- JPEGLSLossless
  - gdcm::TransferSyntax, [766](#)
- JPEGLSLosslessImageCompression
  - gdcm::UIDs, [780](#)
- JPEGLSLossyNearLosslessImageCompression
  - gdcm::UIDs, [780](#)
- JPEGLSNearLossless
  - gdcm::TransferSyntax, [766](#)
- JPEGLosslessHierarchicalProcess28Retired
  - gdcm::UIDs, [780](#)
- JPEGLosslessHierarchicalProcess29Retired
  - gdcm::UIDs, [780](#)
- JPEGLosslessNonHierarchicalFirstOrderPrediction↔
  - Process14SelectionValue1DefaultTransfer↔
  - SyntaxforLosslessJPEGImageCompression
  - gdcm::UIDs, [780](#)
- JPEGLosslessNonHierarchicalProcess14
  - gdcm::UIDs, [779](#)
- JPEGLosslessNonHierarchicalProcess15Retired
  - gdcm::UIDs, [780](#)
- JPEGLosslessProcess14

- gdcmm::TransferSyntax, 766
- JPEGLosslessProcess14\_1
  - gdcmm::TransferSyntax, 766
- JPEGSpectralSelectionHierarchicalProcess2022Retired
  - gdcmm::UIDs, 780
- JPEGSpectralSelectionHierarchicalProcess2123Retired
  - gdcmm::UIDs, 780
- JPEGSpectralSelectionNonHierarchicalProcess68Retired
  - gdcmm::UIDs, 779
- JPEGSpectralSelectionNonHierarchicalProcess79Retired
  - gdcmm::UIDs, 779
- JPEGSpectralSelectionProcess6\_8
  - gdcmm::TransferSyntax, 766
- JPIPIReferenced
  - gdcmm::TransferSyntax, 766
  - gdcmm::UIDs, 780
- JPIPIReferencedDeflate
  - gdcmm::UIDs, 780
- JunkAfterDocElementError
  - gdcmm::Parser, 563
- KeyObjectSelectionDocument
  - gdcmm::MediaStorage, 517
- KeyObjectSelectionDocumentStorage
  - gdcmm::UIDs, 784
- LD\_ALL
  - gdcmm, 128
- LD\_NOSEQ
  - gdcmm, 128
- LD\_NOSHADOW
  - gdcmm, 128
- LD\_NOSHADOWSEQ
  - gdcmm, 128
- LINE
  - gdcmm::MeshPrimitive, 526
- LO
  - gdcmm::VR, 866
- LOADBULKDATA
  - gdcmm::XMLPrinter, 937
- LT
  - gdcmm::VR, 866
- LeadECGWaveformStorage
  - gdcmm::MediaStorage, 517
- LittleEndian
  - gdcmm::SwapCode, 737
- MAGNIFIED
  - gdcmm::Spacing, 702
- MANUAL
  - gdcmm::Segment, 658
- MONOCHROME1
  - gdcmm::PhotometricInterpretation, 578
- MONOCHROME2
  - gdcmm::PhotometricInterpretation, 578
- MPEG2MainProfile
  - gdcmm::TransferSyntax, 766
- MPEG2MainProfileMainLevel
  - gdcmm::UIDs, 780
- MPTType\_END
  - gdcmm::MeshPrimitive, 526
- MRImageStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 782
- MRSpectroscopyStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 782
- MS\_END
  - gdcmm::MediaStorage, 518
- magenta
  - gdcmm::terminal, 140
- MammographyCADSR
  - gdcmm::MediaStorage, 517
- MammographyCADSRStorage
  - gdcmm::UIDs, 783
- Mandatory
  - gdcmm::Usage, 847
- MediaCreationManagementSOPClassUID
  - gdcmm::UIDs, 782
- MediaStorageDirectoryStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 780
- ModalityPerformedProcedureStepNotificationSOPClass
  - gdcmm::UIDs, 781
- ModalityPerformedProcedureStepRetrieveSOPClass
  - gdcmm::UIDs, 781
- ModalityPerformedProcedureStepSOPClass
  - gdcmm::MediaStorage, 518
  - gdcmm::UIDs, 781
- ModalityWorklistInformationModelFIND
  - gdcmm::UIDs, 784
- MultiframeGrayscaleByteSecondaryCaptureImageStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 782
- MultiframeGrayscaleWordSecondaryCaptureImageStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 782
- MultiframeSingleBitSecondaryCaptureImageStorage
  - gdcmm::MediaStorage, 516
  - gdcmm::UIDs, 782
- MultiframeTrueColorSecondaryCaptureImageStorage
  - gdcmm::MediaStorage, 517
  - gdcmm::UIDs, 782
- N\_ACTION\_RQ
  - gdcmm::network::DIMSE, 329
- N\_ACTION\_RSP



- gdcm::network::DIMSE, 329
- N\_CREATE\_RQ
  - gdcm::network::DIMSE, 329
- N\_CREATE\_RSP
  - gdcm::network::DIMSE, 329
- N\_DELETE\_RQ
  - gdcm::network::DIMSE, 329
- N\_DELETE\_RSP
  - gdcm::network::DIMSE, 329
- N\_EVENT\_REPORT\_RQ
  - gdcm::network::DIMSE, 329
- N\_EVENT\_REPORT\_RSP
  - gdcm::network::DIMSE, 329
- N\_GET\_RQ
  - gdcm::network::DIMSE, 329
- N\_GET\_RSP
  - gdcm::network::DIMSE, 329
- N\_SET\_RQ
  - gdcm::network::DIMSE, 329
- N\_SET\_RSP
  - gdcm::network::DIMSE, 329
- NO
  - gdcm::Surface, 726
- NO\_COMPRESSION
  - vtkGDCMImageWriter, 886
- NOMAGIC
  - gdcm::CSAHeader, 273
- NoElementsError
  - gdcm::Parser, 563
- NoError
  - gdcm::Parser, 563
- NoMemoryError
  - gdcm::Parser, 563
- NoObject
  - gdcm::MediaStorage, 518
- NuclearMedicineImageStorage
  - gdcm::MediaStorage, 517
  - gdcm::UIDs, 783
- NuclearMedicineImageStorageRetired
  - gdcm::MediaStorage, 516
  - gdcm::UIDs, 782
- OB
  - gdcm::VR, 866
- OB\_OW
  - gdcm::VR, 866
- OBLIQUE
  - gdcm::Orientation, 553
- OD
  - gdcm::VR, 866
- OF
  - gdcm::VR, 866
- OPENSSL
  - gdcm::CryptoFactory, 264
- OPENSSL7
  - gdcm::CryptoFactory, 264
- OW
  - gdcm::VR, 866
- ObjectEnd
  - gdcm::MediaStorage, 518
- OnlyUUID
  - gdcm::XMLPrinter, 937
- operator!=
  - gdcm, 128
- operator<<
  - gdcm, 128–132
- operator>>
  - gdcm, 132
- operator==
  - gdcm, 132
- OphthalmicPhotography16BitImageStorage
  - gdcm::UIDs, 783
- OphthalmicPhotography8BitImageStorage
  - gdcm::MediaStorage, 518
  - gdcm::UIDs, 783
- OphthalmicTomographyImageStorage
  - gdcm::MediaStorage, 518
  - gdcm::UIDs, 783
- PALETTE\_COLOR
  - gdcm::PhotometricInterpretation, 578
- PDF
  - gdcm::MediaStorage, 518
- PETImageStorage
  - gdcm::MediaStorage, 517
- PHILIPS
  - gdcm::Dicts, 326
- PI\_END
  - gdcm::PhotometricInterpretation, 578
- PN
  - gdcm::VR, 866
- POINTS
  - gdcm::Surface, 726
- PatientRootQueryRetrieveInformationModelFIND
  - gdcm::UIDs, 784
- PatientRootQueryRetrieveInformationModelGET
  - gdcm::UIDs, 784
- PatientRootQueryRetrieveInformationModelMOVE
  - gdcm::UIDs, 784
- PatientStudyOnlyQueryRetrieveInformationModelFIND↔
  - Retired
  - gdcm::UIDs, 784
- PatientStudyOnlyQueryRetrieveInformationModelGET↔
  - Retired
  - gdcm::UIDs, 784
- PatientStudyOnlyQueryRetrieveInformationModelMOVE↔
  - Retired
  - gdcm::UIDs, 784

- Philips3D
  - gdcm::MediaStorage, [517](#)
- PhilipsPrivateMRSyntheticImageStorage
  - gdcm::MediaStorage, [518](#)
- PositronEmissionTomographyImageStorage
  - gdcm::UIDs, [784](#)
- PresentationLUTSOPClass
  - gdcm::UIDs, [782](#)
- PrintJobSOPClass
  - gdcm::UIDs, [781](#)
- PrintQueueManagementSOPClassRetired
  - gdcm::UIDs, [782](#)
- PrintQueueSOPInstanceRetired
  - gdcm::UIDs, [782](#)
- PrinterConfigurationRetrievalSOPClass
  - gdcm::UIDs, [781](#)
- PrinterConfigurationRetrievalSOPInstance
  - gdcm::UIDs, [781](#)
- PrinterSOPClass
  - gdcm::UIDs, [781](#)
- PrinterSOPInstance
  - gdcm::UIDs, [781](#)
- ProceduralEventLoggingSOPClass
  - gdcm::UIDs, [781](#)
- ProceduralEventLoggingSOPInstance
  - gdcm::UIDs, [781](#)
- ProcedureLogStorage
  - gdcm::UIDs, [783](#)
- ProductCharacteristicsQuerySOPClass
  - gdcm::UIDs, [785](#)
- PseudoColorSoftcopyPresentationStateStorageSOP↔
  - Class
  - gdcm::UIDs, [783](#)
- PullPrintRequestSOPClassRetired
  - gdcm::UIDs, [782](#)
- PullStoredPrintManagementMetaSOPClassRetired
  - gdcm::UIDs, [782](#)
- RED
  - gdcm::LookupTable, [505](#)
- RFC2557MIMEencapsulation
  - gdcm::UIDs, [780](#)
- RGB
  - gdcm::PhotometricInterpretation, [578](#)
- RLE\_COMPRESSION
  - vtkGDCMImageWriter, [886](#)
- RLELossless
  - gdcm::TransferSyntax, [766](#)
  - gdcm::UIDs, [780](#)
- ROI
  - gdcm::Overlay, [557](#)
- RTBeamsDeliveryInstructionStorageSupplement74↔
  - FrozenDraft
  - gdcm::UIDs, [784](#)
- RTBeamsTreatmentRecordStorage
  - gdcm::UIDs, [784](#)
- RTBrachyTreatmentRecordStorage
  - gdcm::UIDs, [784](#)
- RTConventionalMachineVerificationSupplement74↔
  - FrozenDraft
  - gdcm::UIDs, [784](#)
- RTDoseStorage
  - gdcm::MediaStorage, [517](#)
  - gdcm::UIDs, [784](#)
- RTImageStorage
  - gdcm::MediaStorage, [517](#)
  - gdcm::UIDs, [784](#)
- RTIonBeamsTreatmentRecordStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [784](#)
- RTIonMachineVerificationSupplement74FrozenDraft
  - gdcm::UIDs, [784](#)
- RTIonPlanStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [784](#)
- RTPlanStorage
  - gdcm::MediaStorage, [517](#)
  - gdcm::UIDs, [784](#)
- RTStructureSetStorage
  - gdcm::MediaStorage, [517](#)
  - gdcm::UIDs, [784](#)
- RTTreatmentSummaryRecordStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [784](#)
- RawDataStorage
  - gdcm::MediaStorage, [517](#)
  - gdcm::UIDs, [783](#)
- RealWorldValueMappingStorage
  - gdcm::UIDs, [783](#)
- red
  - gdcm::terminal, [139](#)
- ReferencedColorPrintManagementMetaSOPClassRetired
  - gdcm::UIDs, [781](#)
- ReferencedGrayscalePrintManagementMetaSOPClass↔
  - Retired
  - gdcm::UIDs, [781](#)
- ReferencedImageBoxSOPClassRetired
  - gdcm::UIDs, [781](#)
- reset
  - gdcm::terminal, [139](#)
- reverse
  - gdcm::terminal, [139](#)
- SAGITTAL
  - gdcm::Orientation, [553](#)
- SH
  - gdcm::VR, [866](#)
- SIEMENS

- gdcm::Dicts, [326](#)
- SINGLEBIT
  - gdcm::PixelFormat, [582](#)
- SL
  - gdcm::VR, [866](#)
- SLICE\_ORIENTATION\_XY
  - vtkImageColorViewer, [907](#)
- SLICE\_ORIENTATION\_XZ
  - vtkImageColorViewer, [907](#)
- SLICE\_ORIENTATION\_YZ
  - vtkImageColorViewer, [907](#)
- SPM2AVG152PDFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2AVG152T1FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2AVG152T2FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2AVG305T1FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2BRAINMASKFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2CSFFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2EPIFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2FILT1FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2GRAYFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2PDFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2PETFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2SINGLESUBJT1FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2SPECTFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2T1FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2T2FrameofReference
  - gdcm::UIDs, [780](#)
- SPM2TRANSMFrameofReference
  - gdcm::UIDs, [780](#)
- SPM2WHITEFrameofReference
  - gdcm::UIDs, [780](#)
- SQ
  - gdcm::VR, [866](#)
- SS
  - gdcm::VR, [866](#)
- ST
  - gdcm::VR, [866](#)
- STATES\_END
  - gdcm::Surface, [726](#)
- SURFACE
  - gdcm::Surface, [726](#)
- SV10
  - gdcm::CSAHeader, [273](#)
- SecondaryCaptureImageStorage
  - gdcm::MediaStorage, [516](#)
  - gdcm::UIDs, [782](#)
- Segmentation
  - gdcm::MediaStorage, [518](#)
- SegmentationStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [783](#)
- SpacialFiducialsStorage
  - gdcm::MediaStorage, [517](#)
- SpacialRegistrationStorage
  - gdcm::MediaStorage, [517](#)
- SpatialFiducialsStorage
  - gdcm::UIDs, [783](#)
- SpatialRegistrationStorage
  - gdcm::UIDs, [783](#)
- StandaloneCurveStorage
  - gdcm::MediaStorage, [517](#)
- StandaloneCurveStorageRetired
  - gdcm::UIDs, [782](#)
- StandaloneModalityLUTStorage
  - gdcm::MediaStorage, [517](#)
- StandaloneModalityLUTStorageRetired
  - gdcm::UIDs, [783](#)
- StandaloneOverlayStorage
  - gdcm::MediaStorage, [517](#)
- StandaloneOverlayStorageRetired
  - gdcm::UIDs, [782](#)
- StandalonePETCurveStorageRetired
  - gdcm::UIDs, [784](#)
- StandaloneVOILUTStorage
  - gdcm::MediaStorage, [517](#)
- StandaloneVOILUTStorageRetired
  - gdcm::UIDs, [783](#)
- StereometricRelationshipStorage
  - gdcm::UIDs, [783](#)
- StorageCommitmentPullModelSOPClassRetired
  - gdcm::UIDs, [781](#)
- StorageCommitmentPullModelSOPInstanceRetired
  - gdcm::UIDs, [781](#)
- StorageCommitmentPushModelSOPClass
  - gdcm::UIDs, [781](#)
- StorageCommitmentPushModelSOPInstance
  - gdcm::UIDs, [781](#)
- StorageServiceClass
  - gdcm::UIDs, [781](#)
- StoredPrintStorageSOPClassRetired
  - gdcm::UIDs, [782](#)
- StudyComponentManagementSOPClass
  - gdcm::MediaStorage, [517](#)
- StudyComponentManagementSOPClassRetired

- gdcmm::UIDs, 781
- StudyRootQueryRetrieveInformationModelFIND
  - gdcmm::UIDs, 784
- StudyRootQueryRetrieveInformationModelGET
  - gdcmm::UIDs, 784
- StudyRootQueryRetrieveInformationModelMOVE
  - gdcmm::UIDs, 784
- SubstanceAdministrationLoggingSOPClass
  - gdcmm::UIDs, 781
- SubstanceAdministrationLoggingSOPInstance
  - gdcmm::UIDs, 781
- SubstanceApprovalQuerySOPClass
  - gdcmm::UIDs, 785
- SurfaceSegmentationStorage
  - gdcmm::MediaStorage, 518
  - gdcmm::UIDs, 786
- SyntaxError
  - gdcmm::Parser, 563
- T1
  - gdcmm::Type, 771
- T1C
  - gdcmm::Type, 771
- T2
  - gdcmm::Type, 771
- T2C
  - gdcmm::Type, 771
- T3
  - gdcmm::Type, 771
- TM
  - gdcmm::VR, 866
- TRIANGLE
  - gdcmm::MeshPrimitive, 526
- TRIANGLE\_FAN
  - gdcmm::MeshPrimitive, 526
- TRIANGLE\_STRIP
  - gdcmm::MeshPrimitive, 526
- TS\_END
  - gdcmm::TransferSyntax, 766
- TagMismatchError
  - gdcmm::Parser, 563
- TalairachBrainAtlasFrameofReference
  - gdcmm::UIDs, 780
- TextSRStorageTrialRetired
  - gdcmm::UIDs, 783
- ToshibaPrivateDataStorage
  - gdcmm::MediaStorage, 517
- UI
  - gdcmm::VR, 866
- UINT12
  - gdcmm::PixelFormat, 581
- UINT16
  - gdcmm::PixelFormat, 581
- UINT32
  - gdcmm::PixelFormat, 581
- UINT8
  - gdcmm::PixelFormat, 581
- UL
  - gdcmm::VR, 866
- UN
  - gdcmm::VR, 866
- UNKNOWN
  - gdcmm::PhotometricInterpretation, 578
- UNKNOWN
  - gdcmm::CSAHeader, 273
  - gdcmm::LookupTable, 505
  - gdcmm::Orientation, 553
  - gdcmm::PixelFormat, 582
  - gdcmm::Spacing, 702
  - gdcmm::Surface, 726
  - gdcmm::Type, 771
- URI
  - gdcmm::MediaStorage, 518
- US
  - gdcmm::VR, 866
- US\_SS
  - gdcmm::VR, 866
- US\_SS\_OW
  - gdcmm::VR, 866
- UT
  - gdcmm::VR, 866
- uid\_1\_2\_840\_10008\_15\_0\_3\_1
  - gdcmm::UIDs, 791
- uid\_1\_2\_840\_10008\_15\_0\_3\_10
  - gdcmm::UIDs, 791
- uid\_1\_2\_840\_10008\_15\_0\_3\_11
  - gdcmm::UIDs, 791
- uid\_1\_2\_840\_10008\_15\_0\_3\_12
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_13
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_14
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_15
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_16
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_17
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_18
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_19
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_2
  - gdcmm::UIDs, 791
- uid\_1\_2\_840\_10008\_15\_0\_3\_20
  - gdcmm::UIDs, 792
- uid\_1\_2\_840\_10008\_15\_0\_3\_21

gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_22  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_23  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_24  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_25  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_26  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_27  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_28  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_29  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_3  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_30  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_31  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_4  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_5  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_6  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_7  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_8  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_3\_9  
gdcmm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_1  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_2  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_3  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_4  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_5  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_6  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_7  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_15\_0\_4\_8  
gdcmm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_1\_1  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_20\_1  
gdcmm::UIDs, [787](#)  
uid\_1\_2\_840\_10008\_1\_20\_1\_1  
gdcmm::UIDs, [787](#)  
uid\_1\_2\_840\_10008\_1\_20\_2  
gdcmm::UIDs, [787](#)  
uid\_1\_2\_840\_10008\_1\_20\_2\_1  
gdcmm::UIDs, [787](#)  
uid\_1\_2\_840\_10008\_1\_2\_1  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_1\_99  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_2  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_100  
gdcmm::UIDs, [787](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_50  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_51  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_52  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_53  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_54  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_55  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_56  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_57  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_58  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_59  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_60  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_61  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_62  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_63  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_64  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_65  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_66  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_70  
gdcmm::UIDs, [786](#)  
uid\_1\_2\_840\_10008\_1\_2\_4\_80

gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_81  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_90  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_91  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_92  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_93  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_94  
   gdcmm::UIDs, [786](#)  
 uid\_1\_2\_840\_10008\_1\_2\_4\_95  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_5  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_6\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_2\_6\_2  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_3\_10  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_40  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_40\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_42  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_42\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_10  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_11  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_12  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_13  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_14  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_15  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_16  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_17  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_18  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_2  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_3

gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_4  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_5  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_6  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_7  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_8  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_1\_9  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_2\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_4\_2\_2  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_1\_9  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_2\_16\_4  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_2\_6\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_1\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_1\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_1\_4  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_2\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_3\_1  
   gdcmm::UIDs, [787](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_3\_2  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_3\_3  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_3\_4  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_3\_5  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_5\_1  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_5\_4  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_5\_5  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_3\_1\_2\_6\_1  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_4\_2  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_1  
   gdcmm::UIDs, [788](#)  
 uid\_1\_2\_840\_10008\_5\_1\_1\_14

gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_15  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_16\_376  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_17\_376  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_18\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_2  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_22  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_23  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_24\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_25  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_26  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_27  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_29  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_30  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_31  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_32  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_33  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_4\_2  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_1\_9\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_10

gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_1  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_104\_2  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_2  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_3  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_11\_4  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_128  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_129  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_1\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_2\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_12\_3  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_2  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_13\_1\_3  
gdcm::UIDs, [792](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_1\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_2\_1  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3  
gdcm::UIDs, [788](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_1\_3\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_20  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_2\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3



gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_3\_1  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_2  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_3  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_4  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_5  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_6  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_7  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_8  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_481\_9  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_1  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_4\_2  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_5  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_1  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_2  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_3  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_4  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_66\_5  
 gdcmm::UIDs, [792](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_67  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_1  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_6\_2  
 gdcmm::UIDs, [792](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1

gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_1\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_2\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_3  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_4\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_2  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_3  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_5\_4  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_1\_6  
 gdcmm::UIDs, [792](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_77\_2  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_1  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_2  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_3  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_7\_4  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_8  
 gdcmm::UIDs, [789](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_1  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_11  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_2  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_22  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_3  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_33  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_4  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_40  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_50  
 gdcmm::UIDs, [790](#)  
 uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_59



gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_65  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_88\_67  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_2  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_1\_3  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_2\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_3\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_1\_9\_4\_1  
gdcm::UIDs, [789](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_1  
gdcm::UIDs, [790](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_1\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_2\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_1\_2\_3\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_31  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_32\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_33  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_2

gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_4\_4  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_34\_5  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_37\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_1  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_2  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_38\_3  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_41  
gdcm::UIDs, [791](#)  
uid\_1\_2\_840\_10008\_5\_1\_4\_42  
gdcm::UIDs, [791](#)  
UltrasoundImageStorage  
gdcm::MediaStorage, [516](#)  
gdcm::UIDs, [782](#)  
UltrasoundImageStorageRetired  
gdcm::MediaStorage, [516](#)  
gdcm::UIDs, [782](#)  
UltrasoundMultiFrameImageStorage  
gdcm::MediaStorage, [516](#)  
UltrasoundMultiFrameImageStorageRetired  
gdcm::MediaStorage, [516](#)  
UltrasoundMultiframeImageStorage  
gdcm::UIDs, [782](#)  
UltrasoundMultiframeImageStorageRetired  
gdcm::UIDs, [782](#)  
UndefinedEntityError  
gdcm::Parser, [563](#)  
underline  
gdcm::terminal, [139](#)  
UnexpectedStateError  
gdcm::Parser, [563](#)  
UnifiedProcedureStepEventSOPClass  
gdcm::UIDs, [784](#)  
UnifiedProcedureStepPullSOPClass

- gdcm::UIDs, [784](#)
- UnifiedProcedureStepPushSOPClass
  - gdcm::UIDs, [784](#)
- UnifiedProcedureStepWatchSOPClass
  - gdcm::UIDs, [784](#)
- UnifiedWorklistandProcedureStepSOPInstance
  - gdcm::UIDs, [784](#)
- UnifiedWorklistandProcedureStepServiceClass
  - gdcm::UIDs, [784](#)
- Unknown
  - gdcm::SwapCode, [737](#)
  - gdcm::TransferSyntax, [766](#)
- UserOption
  - gdcm::Usage, [847](#)
- VERBOSE\_STYLE
  - gdcm::Printer, [611](#)
- VERTEX
  - gdcm::MeshPrimitive, [526](#)
- VIEWType\_END
  - gdcm::Surface, [726](#)
- VL16
  - gdcm::VR, [866](#)
- VL32
  - gdcm::VR, [866](#)
- VLEndoscopicImageStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [783](#)
- VLImageStorageTrialRetired
  - gdcm::UIDs, [783](#)
- VLMicroscopicImageStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [783](#)
- VLMultiframeImageStorageTrialRetired
  - gdcm::UIDs, [783](#)
- VLPhotographicImageStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [783](#)
- VLSlideCoordinatesMicroscopicImageStorage
  - gdcm::UIDs, [783](#)
- VLWholeSlideMicroscopyImageStorage
  - gdcm::MediaStorage, [518](#)
  - gdcm::UIDs, [786](#)
- VM0
  - gdcm::VM, [861](#)
- VM1
  - gdcm::VM, [861](#)
- VM10
  - gdcm::VM, [861](#)
- VM12
  - gdcm::VM, [861](#)
- VM16
  - gdcm::VM, [861](#)
- VM18
  - gdcm::VM, [861](#)
- gdcm::VM, [861](#)
- VM1\_2
  - gdcm::VM, [862](#)
- VM1\_3
  - gdcm::VM, [862](#)
- VM1\_32
  - gdcm::VM, [862](#)
- VM1\_4
  - gdcm::VM, [862](#)
- VM1\_5
  - gdcm::VM, [862](#)
- VM1\_8
  - gdcm::VM, [862](#)
- VM1\_99
  - gdcm::VM, [862](#)
- VM1\_n
  - gdcm::VM, [862](#)
- VM2
  - gdcm::VM, [861](#)
- VM24
  - gdcm::VM, [861](#)
- VM256
  - gdcm::VM, [862](#)
- VM28
  - gdcm::VM, [861](#)
- VM2\_2n
  - gdcm::VM, [862](#)
- VM2\_n
  - gdcm::VM, [862](#)
- VM3
  - gdcm::VM, [861](#)
- VM30\_30n
  - gdcm::VM, [862](#)
- VM32
  - gdcm::VM, [861](#)
- VM35
  - gdcm::VM, [861](#)
- VM3\_3n
  - gdcm::VM, [862](#)
- VM3\_4
  - gdcm::VM, [862](#)
- VM3\_n
  - gdcm::VM, [862](#)
- VM4
  - gdcm::VM, [861](#)
- VM47\_47n
  - gdcm::VM, [862](#)
- VM4\_4n
  - gdcm::VM, [862](#)
- VM5
  - gdcm::VM, [861](#)
- VM6
  - gdcm::VM, [861](#)
- VM6\_6n

- gdcM::VM, [862](#)
- VM7\_7n
  - gdcM::VM, [862](#)
- VM8
  - gdcM::VM, [861](#)
- VM9
  - gdcM::VM, [861](#)
- VM99
  - gdcM::VM, [862](#)
- VM\_END
  - gdcM::VM, [862](#)
- VMType
  - gdcM::Attribute, [173](#)
  - gdcM::Attribute< Group, Element, TVR, VM::VM1 >, [180](#)
- VOILUTBoxSOPClass
  - gdcM::UIDs, [782](#)
- VR\_END
  - gdcM::VR, [866](#)
- VR\_VM1
  - gdcM::VR, [866](#)
- VRALL
  - gdcM::VR, [866](#)
- VRASCI
  - gdcM::VR, [866](#)
- VRBINARY
  - gdcM::VR, [866](#)
- VT100
  - gdcM::terminal, [140](#)
- value
  - gdcM::STATIC\_ASSERTION\_FAILURE< true >, [706](#)
- VerificationSOPClass
  - gdcM::UIDs, [779](#)
- Video
  - gdcM::MediaStorage, [518](#)
- VideoEndoscopicImageStorage
  - gdcM::MediaStorage, [517](#)
  - gdcM::UIDs, [783](#)
- VideoMicroscopicImageStorage
  - gdcM::UIDs, [783](#)
- VideoPhotographicImageStorage
  - gdcM::UIDs, [783](#)
- vtkGDCMImageWriter
  - JPEG2000\_COMPRESSION, [886](#)
  - JPEG\_COMPRESSION, [886](#)
  - JPEGLS\_COMPRESSION, [886](#)
  - NO\_COMPRESSION, [886](#)
  - RLE\_COMPRESSION, [886](#)
- vtkImageColorViewer
  - SLICE\_ORIENTATION\_XY, [907](#)
  - SLICE\_ORIENTATION\_XZ, [907](#)
  - SLICE\_ORIENTATION\_YZ, [907](#)
- WIREFRAME
  - gdcM::Surface, [726](#)
- Waveform
  - gdcM::MediaStorage, [518](#)
- WaveformStorageTrialRetired
  - gdcM::UIDs, [782](#)
- white
  - gdcM::terminal, [140](#)
- XML
  - gdcM::Printer, [611](#)
- XMLEncoding
  - gdcM::UIDs, [780](#)
- XRay3DAngiographicImageStorage
  - gdcM::MediaStorage, [518](#)
  - gdcM::UIDs, [783](#)
- XRay3DCraniofacialImageStorage
  - gdcM::UIDs, [783](#)
- XRayAngiographicBiPlaneImageStorageRetired
  - gdcM::MediaStorage, [517](#)
  - gdcM::UIDs, [783](#)
- XRayAngiographicImageStorage
  - gdcM::MediaStorage, [517](#)
  - gdcM::UIDs, [783](#)
- XRayRadiationDoseSR
  - gdcM::MediaStorage, [518](#)
- XRayRadiationDoseSRStorage
  - gdcM::UIDs, [784](#)
- XRayRadiofluoroscopicImageStorage
  - gdcM::UIDs, [783](#)
- XRayRadiofluoroscopicImageStorage
  - gdcM::MediaStorage, [517](#)
- YBR\_FULL
  - gdcM::PhotometricInterpretation, [578](#)
- YBR\_FULL\_422
  - gdcM::PhotometricInterpretation, [578](#)
- YBR\_ICT
  - gdcM::PhotometricInterpretation, [578](#)
- YBR\_PARTIAL\_420
  - gdcM::PhotometricInterpretation, [578](#)
- YBR\_PARTIAL\_422
  - gdcM::PhotometricInterpretation, [578](#)
- YBR\_RCT
  - gdcM::PhotometricInterpretation, [578](#)
- YES
  - gdcM::Surface, [726](#)
- yellow
  - gdcM::terminal, [139](#)
- ZEROED\_OUT
  - gdcM::CSAHeader, [273](#)