

Spinnaker C++

3.0.0.118

Generated by Doxygen 1.8.17

1 Getting Started	1
2 Programmer's Guide	3
3 Benefits of Spinnaker	5
4 FlyCapture2 Feature Comparison with Spinnaker	7
5 Working with GenICam GenTL Devices	9
5.1 GenTL Overview	9
5.2 Installation	9
5.3 Troubleshooting	10
5.3.1 Enable FLIR GenTL Logging	10
5.3.2 USB3 Device Image Tearing	10
6 Software Licensing Information	11
7 Software Maintenance Policy	13
7.1 GenTL Overview	13
7.2 Platform Support Policy	13
7.2.1 Windows Support	13
7.2.2 Linux Desktop Support	13
7.2.3 Linux Embedded Support	13
7.2.4 MacOS Support	14
7.3 Versioning Policy	14
8 Module Index	15
8.1 Modules	15
9 Namespace Index	19
9.1 Namespace List	19
10 Hierarchical Index	21
10.1 Class Hierarchy	21
11 Class Index	27
11.1 Class List	27
12 File Index	35
12.1 File List	35
13 Module Documentation	41
13.1 Spinnaker Classes	41
13.2 AVI Recorder Class	42
13.3 BasePtr Class	43
13.4 Camera Class	44
13.5 Camera Base Class	45

13.6 CameraDefs Class	46
13.7 Camera List Class	47
13.8 CameraPtr Class	48
13.9 ChunkData Class	49
13.10 Chunk Data Inference Class	50
13.11 Spinnaker EventHandler Classes	51
13.12 DeviceArrivalEventHandler Class	52
13.13 DeviceEventHandler Class	53
13.14 DeviceRemovalEventHandler Class	54
13.15 EventHandler Class	55
13.16 Exception Class	56
13.17 Image Class	57
13.18 ImageEventHandler Class	58
13.19 ImageListEventHandler Class	59
13.20 Image Processor Class	60
13.21 ImagePtr Class	61
13.22 ImageStatistics Class	62
13.23 Image Utility Class	63
13.24 Image Utility CCM Class	64
13.25 Image Utility Heatmap Class	65
13.26 Image Utility Polarization Class	66
13.27 Interface Class	67
13.28 InterfaceArrivalEventHandler Class	68
13.29 InterfaceEventHandler Class	69
13.30 InterfaceList Class	70
13.31 InterfacePtr Class	71
13.32 InterfaceRemovalEventHandler Class	72
13.33 Logging EventHandler Class	73
13.34 LoggingEventDataPtr Class	74
13.35 LoggingEventHandler Class	75
13.36 Spinnaker Headers	76
13.37 Spinnaker.h	77
13.38 Spinnaker Definitions	78
13.39 SpinnakerDirectShow.h	79
13.40 Spinnaker Platform	80
13.41 Spinnaker Video Class	81
13.42 Spinnaker Video Definitions	82
13.43 System Class	83
13.44 SystemEventHandler Class	84
13.45 SystemPtr Class	85
13.46 Spinnaker QuickSpin Classes	86
13.47 TransportLayerDefs Class	87

13.48 TransportLayerDevice Class	88
13.49 TransportLayerInterface Class	89
13.50 TransportLayerStream Class	90
13.51 TransportLayerSystem Class	91
13.52 Camera Base Interface Class	92
13.53 IChunkData Class	93
13.54 IImage Class	94
13.55 Image List Class	95
13.56 IImageStatistics Class	96
13.57 IInterface Class	97
13.58 IInterfaceList Class	98
13.59 ISystem Class	99
13.60 Spinnaker GenApi Classes	100
13.61 AutoVector Class	101
13.62 Spinnaker GenApi Interfaces	102
13.63 IBase Interface	103
13.64 BooleanNode Class	104
13.65 CategoryNode Class	105
13.66 ChunkAdapter Class	106
13.67 ChunkAdapterDcam Class	107
13.68 ChunkAdapterGeneric Class	108
13.69 ChunkAdapterGEV Class	109
13.70 ChunkPort Class	110
13.71 CommandNode Class	111
13.72 Container Class	112
13.73 Counter Class	113
13.74 EnumClasses Class	114
13.75 EnumEntryNode Class	115
13.76 EnumNode Class	116
13.77 EnumNodeT Class	117
13.78 EventAdapter Class	118
13.79 EventAdapter1394 Class	119
13.80 EventAdapterGeneric Class	120
13.81 EventAdapterGEV Class	121
13.82 EventAdapterU3V Class	122
13.83 EventPort Class	123
13.84 Filestream Class	124
13.85 FloatNode Class	125
13.86 FloatRegNode Class	126
13.87 GCString Class	127
13.88 GCSynch Class	128
13.89 GCTypes Class	129

13.89.1 Detailed Description	129
13.89.2 Typedef Documentation	129
13.89.2.1 float32_t	129
13.89.2.2 float64_t	129
13.90 Spinnaker GenApi Utilities	130
13.91 GCUtilities Utility	131
13.92 IBoolean Interface	132
13.93 ICategory Interfaces	133
13.94 IChunkPort Interface	134
13.95 ICommand Interface	135
13.96 IDestroy Interface	136
13.97 IDeviceInfo Interface	137
13.98 IEnumEntry Interface	138
13.99 IEnumeration Interface	139
13.100 IEnumerationT Interface	140
13.101 IFloat Interface	141
13.102 IInteger Interface	142
13.103 INode Interface	143
13.103.1 Detailed Description	143
13.103.2 Function Documentation	143
13.103.2.1 Combine() [1/3]	144
13.103.2.2 Combine() [2/3]	144
13.103.2.3 Combine() [3/3]	144
13.103.2.4 IsAvailable() [1/3]	144
13.103.2.5 IsAvailable() [2/3]	144
13.103.2.6 IsAvailable() [3/3]	145
13.103.2.7 IsCacheable()	145
13.103.2.8 IsImplemented() [1/3]	145
13.103.2.9 IsImplemented() [2/3]	145
13.103.2.10 IsImplemented() [3/3]	145
13.103.2.11 IsReadable() [1/3]	145
13.103.2.12 IsReadable() [2/3]	146
13.103.2.13 IsReadable() [3/3]	146
13.103.2.14 IsVisible()	146
13.103.2.15 IsWritable() [1/3]	146
13.103.2.16 IsWritable() [2/3]	146
13.103.2.17 IsWritable() [3/3]	146
13.104 INodeMap Interface	147
13.105 INodeMapDyn Interface	148
13.106 IntegerNode Class	149
13.107 IntRegNode Class	150
13.108 IPort Interface	151

13.109 IPortConstruct Interface	152
13.110 IPortRecorder Interface	153
13.111 IRegister Interfaces	154
13.112 ISelector Interface	155
13.113 ISelectorDigit Interface	156
13.114 IString Class	157
13.115 IValue Class	158
13.116 Node Class	159
13.117 NodeCallback Class	160
13.118 NodeMap Class	161
13.119 NodeMapFactory Class	162
13.120 NodeMapRef Class	163
13.121 Persistence Class	164
13.122 Pointer Class	165
13.122.1 Detailed Description	166
13.122.2 Typedef Documentation	166
13.122.2.1 CBasePtr	166
13.122.2.2 CBooleanPtr	167
13.122.2.3 CCategoryPtr	167
13.122.2.4 CChunkPortPtr	167
13.122.2.5 CCommandPtr	167
13.122.2.6 CDeviceInfoPtr	167
13.122.2.7 CEnumEntryPtr	167
13.122.2.8 CEnumerationPtr	168
13.122.2.9 CIntegerPtr	168
13.122.2.10 CNodeMapDynPtr	168
13.122.2.11 CNodeMapPtr	168
13.122.2.12 CNodePtr	168
13.122.2.13 CPortConstructPtr	168
13.122.2.14 CPortPtr	169
13.122.2.15 CPortRecorderPtr	169
13.122.2.16 CPortReplayPtr	169
13.122.2.17 CPortWriteListPtr	169
13.122.2.18 CRegisterPtr	169
13.122.2.19 CSelectorPtr	169
13.122.2.20 CStringPtr	170
13.122.2.21 CValuePtr	170
13.122.3 Function Documentation	170
13.122.3.1 GetInterfaceName()	170
13.122.3.2 IsAvailable()	170
13.122.3.3 IsImplemented()	170
13.122.3.4 IsReadable()	170

13.122.3.5 IsWritable()	170
13.123 PortImpl Class	171
13.124 PortNode Class	172
13.125 PortRecorder Class	173
13.126 PortReplay Class	174
13.127 PortWriteList Class	175
13.128 Reference Interfaces	176
13.129 RegisterNode Class	177
13.130 RegisterPortImpl Class	178
13.131 SelectorSet Class	179
13.132 SpinTestCamera Class	180
13.133 StringNode Class	181
13.134 StringRegNode Class	182
13.135 StructPort Class	183
13.136 Synch Class	184
13.137 Spinnaker GenApi Enums	185
13.138 Types Enums	186
13.139 ValueNode Class	187
13.140 ChunkAdapterU3V Class	188
14 Namespace Documentation	189
14.1 Conversion Namespace Reference	189
14.1.1 Function Documentation	189
14.1.1.1 NumToCString() [1/3]	189
14.1.1.2 NumToCString() [2/3]	189
14.1.1.3 NumToCString() [3/3]	189
14.2 CpuUtil Namespace Reference	190
14.2.1 Function Documentation	190
14.2.1.1 GetCpuStats()	190
14.2.1.2 StartCpuTracing()	190
14.2.1.3 StopCpuTracing()	190
14.3 PerformanceCounter Namespace Reference	190
14.3.1 Function Documentation	191
14.3.1.1 GetPerformanceCounter()	191
14.3.1.2 StartPerformanceCounter()	191
14.3.2 Variable Documentation	191
14.3.2.1 CounterStart	191
14.3.2.2 PCFreq	191
14.4 SecondsCounter Namespace Reference	191
14.4.1 Function Documentation	192
14.4.1.1 GetSecondsCounter()	192
14.4.1.2 StartSecondsCounter()	192

14.4.2 Variable Documentation	192
14.4.2.1 endTime	192
14.4.2.2 startTime	192
14.4.2.3 timeDiff	192
14.5 Spinnaker Namespace Reference	192
14.5.1 Enumeration Type Documentation	233
14.5.1.1 AcquisitionModeEnums	234
14.5.1.2 AcquisitionStatusSelectorEnums	234
14.5.1.3 ActionCommandStatus	234
14.5.1.4 ActionUnconditionalModeEnums	235
14.5.1.5 AdcBitDepthEnums	235
14.5.1.6 AutoAlgorithmSelectorEnums	235
14.5.1.7 AutoExposureControlPriorityEnums	236
14.5.1.8 AutoExposureLightingModeEnums	236
14.5.1.9 AutoExposureMeteringModeEnums	236
14.5.1.10 AutoExposureTargetGreyValueAutoEnums	237
14.5.1.11 BalanceRatioSelectorEnums	237
14.5.1.12 BalanceWhiteAutoEnums	237
14.5.1.13 BalanceWhiteAutoProfileEnums	238
14.5.1.14 BinningHorizontalModeEnums	238
14.5.1.15 BinningSelectorEnums	239
14.5.1.16 BinningVerticalModeEnums	239
14.5.1.17 BlackLevelAutoBalanceEnums	239
14.5.1.18 BlackLevelAutoEnums	240
14.5.1.19 BlackLevelSelectorEnums	240
14.5.1.20 BufferOwnership	240
14.5.1.21 CCMApplcation	240
14.5.1.22 CCMColorSpace	241
14.5.1.23 CCMColorTemperature	241
14.5.1.24 CCMSensor	241
14.5.1.25 CCMTpe	242
14.5.1.26 ChunkBlackLevelSelectorEnums	242
14.5.1.27 ChunkCounterSelectorEnums	242
14.5.1.28 ChunkEncoderSelectorEnums	242
14.5.1.29 ChunkEncoderStatusEnums	243
14.5.1.30 ChunkExposureTimeSelectorEnums	243
14.5.1.31 ChunkGainSelectorEnums	244
14.5.1.32 ChunkImageComponentEnums	244
14.5.1.33 ChunkPixelFormatEnums	244
14.5.1.34 ChunkRegionIDEnums	245
14.5.1.35 ChunkScan3dCoordinateReferenceSelectorEnums	245
14.5.1.36 ChunkScan3dCoordinateSelectorEnums	246

14.5.1.37 ChunkScan3dCoordinateSystemEnums	246
14.5.1.38 ChunkScan3dCoordinateSystemReferenceEnums	246
14.5.1.39 ChunkScan3dCoordinateTransformSelectorEnums	247
14.5.1.40 ChunkScan3dDistanceUnitEnums	247
14.5.1.41 ChunkScan3dOutputModeEnums	247
14.5.1.42 ChunkSelectorEnums	248
14.5.1.43 ChunkSourceIDEnums	249
14.5.1.44 ChunkTimerSelectorEnums	249
14.5.1.45 ChunkTransferStreamIDEnums	249
14.5.1.46 CIConfigurationEnums	250
14.5.1.47 CITimeSlotsCountEnums	250
14.5.1.48 ColorProcessingAlgorithm	251
14.5.1.49 ColorTransformationSelectorEnums	251
14.5.1.50 ColorTransformationValueSelectorEnums	252
14.5.1.51 CompressionSaturationPriorityEnums	252
14.5.1.52 CounterEventActivationEnums	252
14.5.1.53 CounterEventSourceEnums	253
14.5.1.54 CounterResetActivationEnums	253
14.5.1.55 CounterResetSourceEnums	254
14.5.1.56 CounterSelectorEnums	254
14.5.1.57 CounterStatusEnums	255
14.5.1.58 CounterTriggerActivationEnums	255
14.5.1.59 CounterTriggerSourceEnums	255
14.5.1.60 CxpConnectionTestModeEnums	256
14.5.1.61 CxpLinkConfigurationEnums	256
14.5.1.62 CxpLinkConfigurationPreferredEnums	257
14.5.1.63 CxpLinkConfigurationStatusEnums	258
14.5.1.64 CxpPoCxpStatusEnums	259
14.5.1.65 DecimationHorizontalModeEnums	260
14.5.1.66 DecimationSelectorEnums	260
14.5.1.67 DecimationVerticalModeEnums	260
14.5.1.68 DefectCorrectionModeEnums	260
14.5.1.69 DeinterlacingEnums	261
14.5.1.70 DeviceAccessStatusEnum	261
14.5.1.71 DeviceCharacterSetEnums	262
14.5.1.72 DeviceClockSelectorEnums	262
14.5.1.73 DeviceConnectionStatusEnums	262
14.5.1.74 DeviceCurrentSpeedEnum	262
14.5.1.75 DeviceEndianessMechanismEnum	263
14.5.1.76 DeviceIndicatorModeEnums	263
14.5.1.77 DeviceLinkHeartbeatModeEnums	263
14.5.1.78 DeviceLinkThroughputLimitModeEnums	264

14.5.1.79 DevicePowerSupplySelectorEnums	264
14.5.1.80 DeviceRegistersEndiannessEnums	264
14.5.1.81 DeviceScanTypeEnums	265
14.5.1.82 DeviceSerialPortBaudRateEnums	265
14.5.1.83 DeviceSerialPortSelectorEnums	265
14.5.1.84 DeviceStreamChannelEndiannessEnums	266
14.5.1.85 DeviceStreamChannelTypeEnums	266
14.5.1.86 DeviceTapGeometryEnums	266
14.5.1.87 DeviceTemperatureSelectorEnums	267
14.5.1.88 DeviceTLTypeEnums	268
14.5.1.89 DeviceTypeEnum	268
14.5.1.90 DeviceTypeEnums	268
14.5.1.91 EncoderModeEnums	270
14.5.1.92 EncoderOutputModeEnums	270
14.5.1.93 EncoderResetActivationEnums	271
14.5.1.94 EncoderResetSourceEnums	271
14.5.1.95 EncoderSelectorEnums	272
14.5.1.96 EncoderSourceAEnums	272
14.5.1.97 EncoderSourceBEnums	273
14.5.1.98 EncoderStatusEnums	273
14.5.1.99 Error	273
14.5.1.100 EventNotificationEnums	275
14.5.1.101 EventSelectorEnums	275
14.5.1.102 EventType	275
14.5.1.103 ExposureActiveModeEnums	276
14.5.1.104 ExposureAutoEnums	276
14.5.1.105 ExposureModeEnums	276
14.5.1.106 ExposureTimeModeEnums	277
14.5.1.107 ExposureTimeSelectorEnums	277
14.5.1.108 FileOpenModeEnums	277
14.5.1.109 FileOperationSelectorEnums	279
14.5.1.110 FileOperationStatusEnums	279
14.5.1.111 FileSelectorEnums	279
14.5.1.112 FilterDriverStatusEnum	280
14.5.1.113 GainAutoBalanceEnums	280
14.5.1.114 GainAutoEnums	280
14.5.1.115 GainSelectorEnums	281
14.5.1.116 GenICamXMLLocationEnum	281
14.5.1.117 GevCCPEnum	281
14.5.1.118 GevCCPEnums	282
14.5.1.119 GevCurrentPhysicalLinkConfigurationEnums	282
14.5.1.120 GevGVCPExtendedStatusCodesSelectorEnums	282

14.5.1.121	GevGVSPExtendedIDModeEnums	283
14.5.1.122	GevIEEE1588ClockAccuracyEnums	283
14.5.1.123	GevIEEE1588ModeEnums	283
14.5.1.124	GevIEEE1588StatusEnums	284
14.5.1.125	GevIPConfigurationStatusEnums	284
14.5.1.126	GevPhysicalLinkConfigurationEnums	284
14.5.1.127	GevSupportedOptionSelectorEnums	285
14.5.1.128	GUIXMLLocationEnum	285
14.5.1.129	HeatmapColor	287
14.5.1.130	ImageComponentSelectorEnums	287
14.5.1.131	ImageCompressionJPEGFormatOptionEnums	288
14.5.1.132	ImageCompressionModeEnums	288
14.5.1.133	ImageCompressionRateOptionEnums	289
14.5.1.134	ImageFileFormat	289
14.5.1.135	ImagePayloadType	289
14.5.1.136	ImageScalingAlgorithm	290
14.5.1.137	ImageStatus	290
14.5.1.138	InferenceBoxType	291
14.5.1.139	InterfaceTypeEnum	291
14.5.1.140	LineFormatEnums	292
14.5.1.141	LineInputFilterSelectorEnums	292
14.5.1.142	LineModeEnums	292
14.5.1.143	LineSelectorEnums	293
14.5.1.144	LineSourceEnums	293
14.5.1.145	LogicBlockLUTInputActivationEnums	294
14.5.1.146	LogicBlockLUTInputSelectorEnums	294
14.5.1.147	LogicBlockLUTInputSourceEnums	294
14.5.1.148	LogicBlockLUTSelectorEnums	295
14.5.1.149	LogicBlockSelectorEnums	295
14.5.1.150	LUTSelectorEnums	296
14.5.1.151	PixelColorFilterEnums	296
14.5.1.152	PixelFormatEnums	296
14.5.1.153	PixelFormatInfoSelectorEnums	302
14.5.1.154	PixelFormatIntType	308
14.5.1.155	PixelSizeEnums	308
14.5.1.156	POEStatusEnum	309
14.5.1.157	PolarizationQuadrant	309
14.5.1.158	RegionDestinationEnums	309
14.5.1.159	RegionModeEnums	310
14.5.1.160	RegionSelectorEnums	310
14.5.1.161	RgbTransformLightSourceEnums	310
14.5.1.162	Scan3dCoordinateReferenceSelectorEnums	311

14.5.1.163 Scan3dCoordinateSelectorEnums	311
14.5.1.164 Scan3dCoordinateSystemEnums	312
14.5.1.165 Scan3dCoordinateSystemReferenceEnums	312
14.5.1.166 Scan3dCoordinateTransformSelectorEnums	312
14.5.1.167 Scan3dDistanceUnitEnums	313
14.5.1.168 Scan3dOutputModeEnums	313
14.5.1.169 SensorDigitizationTapsEnums	314
14.5.1.170 SensorShutterModeEnums	314
14.5.1.171 SensorTapsEnums	315
14.5.1.172 SequencerConfigurationModeEnums	315
14.5.1.173 SequencerConfigurationValidEnums	315
14.5.1.174 SequencerModeEnums	316
14.5.1.175 SequencerSetValidEnums	316
14.5.1.176 SequencerTriggerActivationEnums	316
14.5.1.177 SequencerTriggerSourceEnums	316
14.5.1.178 SerialPortBaudRateEnums	317
14.5.1.179 SerialPortParityEnums	317
14.5.1.180 SerialPortSelectorEnums	318
14.5.1.181 SerialPortSourceEnums	318
14.5.1.182 SerialPortStopBitsEnums	318
14.5.1.183 SoftwareSignalSelectorEnums	319
14.5.1.184 SourceDataRange	319
14.5.1.185 SourceSelectorEnums	319
14.5.1.186 SpinnakerLogLevel	320
14.5.1.187 StatisticsChannel	320
14.5.1.188 StreamBufferCountModeEnum	321
14.5.1.189 StreamBufferHandlingModeEnum	321
14.5.1.190 StreamModeEnum	322
14.5.1.191 StreamTypeEnum	322
14.5.1.192 TestPatternEnums	322
14.5.1.193 TestPatternGeneratorSelectorEnums	323
14.5.1.194 TIFFCompressionMethod	323
14.5.1.195 TimerSelectorEnums	323
14.5.1.196 TimerStatusEnums	325
14.5.1.197 TimerTriggerActivationEnums	325
14.5.1.198 TimerTriggerSourceEnums	325
14.5.1.199 TLPayloadType	327
14.5.1.200 TLPixelFormatNamespace	327
14.5.1.201 TLTypeEnum	328
14.5.1.202 TransferComponentSelectorEnums	328
14.5.1.203 TransferControlModeEnums	329
14.5.1.204 TransferOperationModeEnums	329

14.5.1.205 TransferQueueModeEnums	329
14.5.1.206 TransferSelectorEnums	330
14.5.1.207 TransferStatusSelectorEnums	330
14.5.1.208 TransferTriggerActivationEnums	330
14.5.1.209 TransferTriggerModeEnums	331
14.5.1.210 TransferTriggerSelectorEnums	331
14.5.1.211 TransferTriggerSourceEnums	332
14.5.1.212 TriggerActivationEnums	333
14.5.1.213 TriggerModeEnums	333
14.5.1.214 TriggerOverlapEnums	333
14.5.1.215 TriggerSelectorEnums	334
14.5.1.216 TriggerSourceEnums	334
14.5.1.217 UserOutputSelectorEnums	334
14.5.1.218 UserSetDefaultEnums	335
14.5.1.219 UserSetSelectorEnums	335
14.5.1.220 WhiteClipSelectorEnums	335
14.5.2 Function Documentation	336
14.5.2.1 DEPRECATED_CLASS()	336
14.5.2.2 operator==()	337
14.5.3 Variable Documentation	338
14.5.3.1 EVENT_TIMEOUT_INFINITE	338
14.5.3.2 EVENT_TIMEOUT_NONE	338
14.6 Spinnaker::GenApi Namespace Reference	338
14.6.1 Typedef Documentation	353
14.6.1.1 CallbackHandleType	353
14.6.1.2 CBooleanRef	353
14.6.1.3 CCategoryRef	354
14.6.1.4 CCommandRef	354
14.6.1.5 CEnumEntryRef	354
14.6.1.6 CEnumerationRef	354
14.6.1.7 CFloatRef	354
14.6.1.8 CIntegerRef	354
14.6.1.9 CNodeMapRef	354
14.6.1.10 CNodeRef	354
14.6.1.11 CPortRecorderRef	355
14.6.1.12 CPortRef	355
14.6.1.13 CRegisterRef	355
14.6.1.14 CSelectorRef	355
14.6.1.15 CStringRef	355
14.6.1.16 CValueRef	355
14.6.1.17 IDevFileStream	355
14.6.1.18 NodeList_t	356

14.6.1.19 ODevFileStream	356
14.6.1.20 StringList_t	356
14.6.2 Enumeration Type Documentation	356
14.6.2.1 _EAccessMode	356
14.6.2.2 _ECachingMode	356
14.6.2.3 _ECallbackType	357
14.6.2.4 _EDisplayNotation	357
14.6.2.5 _EEndianness	357
14.6.2.6 _EGenApiSchemaVersion	358
14.6.2.7 _EIncMode	358
14.6.2.8 _EInputDirection	358
14.6.2.9 _EInterfaceType	359
14.6.2.10 _ELinkType	359
14.6.2.11 _ENamespace	359
14.6.2.12 _ERepresentation	360
14.6.2.13 _ESign	360
14.6.2.14 _ESlope	360
14.6.2.15 _EStandardNamespace	361
14.6.2.16 _EVisibility	361
14.6.2.17 _EXMLValidation	361
14.6.2.18 _EYesNo	362
14.6.2.19 ECacheUsage_t	362
14.6.2.20 EContentType_t	363
14.6.2.21 GVCP_MESSAGE_TAGS	363
14.6.3 Function Documentation	363
14.6.3.1 _ClearXMLCache()	363
14.6.3.2 _Connect() [1/2]	363
14.6.3.3 _Connect() [2/2]	363
14.6.3.4 _GetDeviceName()	364
14.6.3.5 _GetNode()	364
14.6.3.6 _GetNodes()	364
14.6.3.7 _GetSupportedSchemaVersions()	364
14.6.3.8 _InvalidateNodes()	364
14.6.3.9 _LoadXMLFromFile()	364
14.6.3.10 _LoadXMLFromFileInject()	364
14.6.3.11 _LoadXMLFromString()	365
14.6.3.12 _LoadXMLFromStringInject()	365
14.6.3.13 _LoadXMLFromZIPData()	365
14.6.3.14 _LoadXMLFromZIPFile()	365
14.6.3.15 _Poll()	365
14.6.3.16 CacheChunkData()	365
14.6.3.17 CastToIDestroy()	366

14.6.3.18 Connect() [1/2]	366
14.6.3.19 Connect() [2/2]	366
14.6.3.20 Deregister()	366
14.6.3.21 DeregisterCallback()	366
14.6.3.22 EatComments()	367
14.6.3.23 ExtractIndependentSubtree()	367
14.6.3.24 FromString()	367
14.6.3.25 Get()	367
14.6.3.26 GetAddress()	368
14.6.3.27 GetAlias()	368
14.6.3.28 GetCachingMode()	368
14.6.3.29 GetCastAlias()	368
14.6.3.30 GetChildren()	368
14.6.3.31 GetCookie()	369
14.6.3.32 GetCurrentEntry()	369
14.6.3.33 GetDescription()	369
14.6.3.34 GetDeviceName()	369
14.6.3.35 GetDeviceVersion()	369
14.6.3.36 GetDisplayName()	370
14.6.3.37 GetDisplayNotation()	370
14.6.3.38 GetDisplayPrecision()	370
14.6.3.39 GetDocuURL()	370
14.6.3.40 GetEntries()	370
14.6.3.41 GetEntry() [1/2]	370
14.6.3.42 GetEntry() [2/2]	371
14.6.3.43 GetEntryByName()	371
14.6.3.44 GetEventID()	371
14.6.3.45 GetGenApiVersion()	371
14.6.3.46 GetInc()	371
14.6.3.47 GetIncMode()	371
14.6.3.48 GetIntValue()	371
14.6.3.49 GetLength()	372
14.6.3.50 GetListOfValidValues()	372
14.6.3.51 GetLock()	372
14.6.3.52 GetMax()	372
14.6.3.53 GetMaxLength()	373
14.6.3.54 GetMin()	373
14.6.3.55 GetNameSpace()	373
14.6.3.56 GetNode()	373
14.6.3.57 GetNodeMap()	373
14.6.3.58 GetNumericValue()	373
14.6.3.59 GetNumNodes()	373

14.6.3.60	GetParents()	373
14.6.3.61	GetPollingTime()	374
14.6.3.62	GetPrincipalInterfaceType()	374
14.6.3.63	GetProductGuid()	374
14.6.3.64	GetProperty()	374
14.6.3.65	GetPropertyNames()	374
14.6.3.66	GetRepresentation()	375
14.6.3.67	GetSchemaVersion()	375
14.6.3.68	GetSelectedFeatures()	375
14.6.3.69	GetSelectingFeatures()	375
14.6.3.70	GetSelectorList()	375
14.6.3.71	GetStandardNameSpace()	376
14.6.3.72	GetSupportedSchemaVersions()	376
14.6.3.73	GetSwapEndianness()	376
14.6.3.74	GetSymbolic()	376
14.6.3.75	GetToolTip()	376
14.6.3.76	GetUnit()	376
14.6.3.77	GetValue()	376
14.6.3.78	GetVendorName()	377
14.6.3.79	GetVersionGuid()	377
14.6.3.80	GetVisibility()	377
14.6.3.81	HasInc()	377
14.6.3.82	ImposeAccessMode()	377
14.6.3.83	ImposeMax() [1/2]	378
14.6.3.84	ImposeMax() [2/2]	378
14.6.3.85	ImposeMin() [1/2]	378
14.6.3.86	ImposeMin() [2/2]	378
14.6.3.87	ImposeVisibility()	378
14.6.3.88	InvalidateNode()	378
14.6.3.89	InvalidateNodes()	379
14.6.3.90	IsAccessModeCacheable()	379
14.6.3.91	IsCachable()	379
14.6.3.92	IsDeprecated()	379
14.6.3.93	IsDone()	379
14.6.3.94	IsFeature()	380
14.6.3.95	IsSelfClearing()	380
14.6.3.96	IsStreamable()	380
14.6.3.97	IsValueCacheValid()	380
14.6.3.98	LoadXMLFromFile()	380
14.6.3.99	LoadXMLFromFileInject()	380
14.6.3.100	LoadXMLFromString()	381
14.6.3.101	LoadXMLFromStringInject()	381

14.6.3.102 LoadXMLFromZIPData()	381
14.6.3.103 LoadXMLFromZIPFile()	381
14.6.3.104 make_NodeCallback() [1/2]	381
14.6.3.105 make_NodeCallback() [2/2]	382
14.6.3.106 MergeXMLFiles()	382
14.6.3.107 operator!=(())	382
14.6.3.108 operator()(())	382
14.6.3.109 operator*()	382
14.6.3.110 operator<<()	383
14.6.3.111 operator=() [1/5]	383
14.6.3.112 operator=() [2/5]	383
14.6.3.113 operator=() [3/5]	383
14.6.3.114 operator=() [4/5]	383
14.6.3.115 operator=() [5/5]	384
14.6.3.116 operator==(())	384
14.6.3.117 operator>>()	384
14.6.3.118 PersistFeature()	384
14.6.3.119 Poll()	384
14.6.3.120 PreprocessXMLFromFile()	384
14.6.3.121 PreprocessXMLFromZIPFile()	385
14.6.3.122 Register() [1/2]	385
14.6.3.123 Register() [2/2]	385
14.6.3.124 RegisterCallback()	386
14.6.3.125 Replay()	386
14.6.3.126 Restore()	386
14.6.3.127 SET_GUID()	386
14.6.3.128 SetCookie()	386
14.6.3.129 SetIntValue()	386
14.6.3.130 SetNext()	387
14.6.3.131 SetNumEnums()	387
14.6.3.132 StopRecording()	387
14.6.3.133 ToString() [1/2]	387
14.6.3.134 ToString() [2/2]	387
14.6.3.135 Write()	388
14.6.4 Variable Documentation	388
14.6.4.1 Address	388
14.6.4.2 COMMAND_MAGIC	388
14.6.4.3 GENCP_COMMAND_HEADER_SIZE	388
14.6.4.4 GENCP_EVENT_BASIC_SIZE	388
14.6.4.5 GENCP_EVENT_CMD_ID	389
14.6.4.6 IBase	389
14.6.4.7 IBoolean	389

14.6.4.8 ICategory	389
14.6.4.9 IChunkPort	389
14.6.4.10 ICommand	389
14.6.4.11 IDestroy	390
14.6.4.12 IDeviceInfo	390
14.6.4.13 IEnumEntry	390
14.6.4.14 IEnumeration	390
14.6.4.15 IEnumerationT	390
14.6.4.16 IEnumReference	391
14.6.4.17 IFloat	391
14.6.4.18 IInteger	391
14.6.4.19 INode	391
14.6.4.20 INodeMap	391
14.6.4.21 INodeMapDyn	392
14.6.4.22 Invalidate	392
14.6.4.23 IPersistScript	392
14.6.4.24 IPort	392
14.6.4.25 IPortConstruct	392
14.6.4.26 IPortRecorder	392
14.6.4.27 IPortReplay	393
14.6.4.28 IPortWriteList	393
14.6.4.29 IReference	393
14.6.4.30 IRegister	393
14.6.4.31 ISelector	393
14.6.4.32 ISelectorDigit	394
14.6.4.33 IString	394
14.6.4.34 IValue	394
14.6.4.35 Length	394
14.6.4.36 U3V_EVENT_PREFIX	394
14.6.4.37 Verify	394
14.7 Spinnaker::GenICam Namespace Reference	395
14.7.1 Function Documentation	396
14.7.1.1 DoesEnvironmentVariableExist()	396
14.7.1.2 GetFiles()	396
14.7.1.3 GetGenICamCacheFolder()	397
14.7.1.4 GetGenICamCLProtocolFolder()	397
14.7.1.5 GetGenICamLogConfig()	397
14.7.1.6 getline() [1/2]	397
14.7.1.7 getline() [2/2]	397
14.7.1.8 GetModulePathFromFunction()	398
14.7.1.9 GetValueOfEnvironmentVariable() [1/2]	398
14.7.1.10 GetValueOfEnvironmentVariable() [2/2]	398

14.7.1.11 INTEGRAL_CAST()	398
14.7.1.12 INTEGRAL_CAST2()	399
14.7.1.13 ReplaceEnvironmentVariables()	399
14.7.1.14 SetGenICamCacheFolder()	399
14.7.1.15 SetGenICamCLProtocolFolder()	399
14.7.1.16 SetGenICamLogConfig()	399
14.7.1.17 ThrowBadAlloc()	399
14.7.1.18 Tokenize()	399
14.7.1.19 UriDecode()	400
14.7.1.20 UriEncode()	400
14.8 Spinnaker::Video Namespace Reference	400
15 Class Documentation	401
15.1 ActionCommandResult Struct Reference	401
15.1.1 Detailed Description	401
15.1.2 Member Data Documentation	401
15.1.2.1 DeviceAddress	401
15.1.2.2 Status	401
15.2 AttachStatistics_t Struct Reference	402
15.2.1 Detailed Description	402
15.2.2 Member Data Documentation	402
15.2.2.1 NumAttachedChunks	402
15.2.2.2 NumChunkPorts	402
15.2.2.3 NumChunks	402
15.3 AutoLock Class Reference	403
15.3.1 Constructor & Destructor Documentation	403
15.3.1.1 AutoLock()	403
15.3.1.2 ~AutoLock()	403
15.4 AutoLock Class Reference	403
15.4.1 Constructor & Destructor Documentation	403
15.4.1.1 AutoLock()	403
15.4.1.2 ~AutoLock()	404
15.5 AVIOption Struct Reference	404
15.5.1 Detailed Description	404
15.5.2 Constructor & Destructor Documentation	404
15.5.2.1 AVIOption()	404
15.5.3 Member Data Documentation	404
15.5.3.1 frameRate	405
15.5.3.2 height	405
15.5.3.3 reserved	405
15.5.3.4 width	405
15.6 BasePtr< T, B > Class Template Reference	405

15.6.1 Detailed Description	407
15.6.2 Constructor & Destructor Documentation	407
15.6.2.1 BasePtr() [1/2]	407
15.6.2.2 ~BasePtr()	407
15.6.2.3 BasePtr() [2/2]	407
15.6.3 Member Function Documentation	407
15.6.3.1 get()	407
15.6.3.2 IsValid()	408
15.6.3.3 operator bool()	408
15.6.3.4 operator T*()	408
15.6.3.5 operator!=() [1/4]	408
15.6.3.6 operator!=() [2/4]	408
15.6.3.7 operator!=() [3/4]	408
15.6.3.8 operator!=() [4/4]	409
15.6.3.9 operator->()	409
15.6.3.10 operator=() [1/4]	409
15.6.3.11 operator=() [2/4]	409
15.6.3.12 operator=() [3/4]	409
15.6.3.13 operator=() [4/4]	409
15.6.3.14 operator==() [1/4]	410
15.6.3.15 operator==() [2/4]	410
15.6.3.16 operator==() [3/4]	410
15.6.3.17 operator==() [4/4]	410
15.6.4 Friends And Related Function Documentation	410
15.6.4.1 CameraListImpl	410
15.6.4.2 InterfaceListImpl	410
15.6.5 Member Data Documentation	411
15.6.5.1 m_pT	411
15.7 BMPOption Struct Reference	411
15.7.1 Detailed Description	411
15.7.2 Constructor & Destructor Documentation	411
15.7.2.1 BMPOption()	411
15.7.3 Member Data Documentation	411
15.7.3.1 indexedColor_8bit	412
15.7.3.2 reserved	412
15.8 BooleanNode Class Reference	412
15.8.1 Detailed Description	413
15.8.2 Constructor & Destructor Documentation	413
15.8.2.1 BooleanNode() [1/2]	414
15.8.2.2 BooleanNode() [2/2]	414
15.8.2.3 ~BooleanNode()	414
15.8.3 Member Function Documentation	414

15.8.3.1 GetValue()	414
15.8.3.2 operator=()	414
15.8.3.3 SetReference()	415
15.8.3.4 SetValue()	415
15.9 Camera Class Reference	415
15.9.1 Detailed Description	441
15.9.2 Constructor & Destructor Documentation	441
15.9.2.1 ~Camera()	442
15.9.2.2 Camera()	442
15.9.3 Member Function Documentation	442
15.9.3.1 Init()	442
15.9.4 Member Data Documentation	442
15.9.4.1 AasRoiEnable	442
15.9.4.2 AasRoiHeight	442
15.9.4.3 AasRoiOffsetX	443
15.9.4.4 AasRoiOffsetY	443
15.9.4.5 AasRoiWidth	443
15.9.4.6 AcquisitionAbort	443
15.9.4.7 AcquisitionArm	444
15.9.4.8 AcquisitionBurstFrameCount	444
15.9.4.9 AcquisitionFrameCount	444
15.9.4.10 AcquisitionFrameRate	444
15.9.4.11 AcquisitionFrameRateEnable	444
15.9.4.12 AcquisitionLineRate	445
15.9.4.13 AcquisitionMode	445
15.9.4.14 AcquisitionResultingFrameRate	445
15.9.4.15 AcquisitionStart	445
15.9.4.16 AcquisitionStatus	445
15.9.4.17 AcquisitionStatusSelector	445
15.9.4.18 AcquisitionStop	446
15.9.4.19 ActionDeviceKey	446
15.9.4.20 ActionGroupKey	446
15.9.4.21 ActionGroupMask	446
15.9.4.22 ActionQueueSize	446
15.9.4.23 ActionSelector	446
15.9.4.24 ActionUnconditionalMode	447
15.9.4.25 AdaptiveCompressionEnable	447
15.9.4.26 AdcBitDepth	447
15.9.4.27 aPAUSEMACCtrlFramesReceived	447
15.9.4.28 aPAUSEMACCtrlFramesTransmitted	447
15.9.4.29 AutoAlgorithmSelector	448
15.9.4.30 AutoExposureControlLoopDamping	448

15.9.4.31 AutoExposureControlPriority	448
15.9.4.32 AutoExposureEVCompensation	448
15.9.4.33 AutoExposureExposureTimeLowerLimit	449
15.9.4.34 AutoExposureExposureTimeUpperLimit	449
15.9.4.35 AutoExposureGainLowerLimit	449
15.9.4.36 AutoExposureGainUpperLimit	449
15.9.4.37 AutoExposureGreyValueLowerLimit	450
15.9.4.38 AutoExposureGreyValueUpperLimit	450
15.9.4.39 AutoExposureLightingMode	450
15.9.4.40 AutoExposureMeteringMode	450
15.9.4.41 AutoExposureTargetGreyValue	451
15.9.4.42 AutoExposureTargetGreyValueAuto	451
15.9.4.43 BalanceRatio	451
15.9.4.44 BalanceRatioSelector	451
15.9.4.45 BalanceWhiteAuto	452
15.9.4.46 BalanceWhiteAutoDamping	452
15.9.4.47 BalanceWhiteAutoLowerLimit	452
15.9.4.48 BalanceWhiteAutoProfile	452
15.9.4.49 BalanceWhiteAutoUpperLimit	453
15.9.4.50 BinningHorizontal	453
15.9.4.51 BinningHorizontalMode	453
15.9.4.52 BinningSelector	453
15.9.4.53 BinningVertical	454
15.9.4.54 BinningVerticalMode	454
15.9.4.55 BlackLevel	454
15.9.4.56 BlackLevelAuto	454
15.9.4.57 BlackLevelAutoBalance	454
15.9.4.58 BlackLevelClampingEnable	455
15.9.4.59 BlackLevelRaw	455
15.9.4.60 BlackLevelSelector	455
15.9.4.61 ChunkBlackLevel	455
15.9.4.62 ChunkBlackLevelSelector	456
15.9.4.63 ChunkCompressionMode	456
15.9.4.64 ChunkCompressionRatio	456
15.9.4.65 ChunkCounterSelector	456
15.9.4.66 ChunkCounterValue	456
15.9.4.67 ChunkCRC	456
15.9.4.68 ChunkEnable	457
15.9.4.69 ChunkEncoderSelector	457
15.9.4.70 ChunkEncoderStatus	457
15.9.4.71 ChunkEncoderValue	457
15.9.4.72 ChunkExposureEndLineStatusAll	457

15.9.4.73 ChunkExposureTime	457
15.9.4.74 ChunkExposureTimeSelector	458
15.9.4.75 ChunkFrameID	458
15.9.4.76 ChunkGain	458
15.9.4.77 ChunkGainSelector	458
15.9.4.78 ChunkHeight	458
15.9.4.79 ChunkImage	458
15.9.4.80 ChunkImageComponent	459
15.9.4.81 ChunkInferenceBoundingBoxResult	459
15.9.4.82 ChunkInferenceConfidence	459
15.9.4.83 ChunkInferenceFrameId	459
15.9.4.84 ChunkInferenceResult	459
15.9.4.85 ChunkLinePitch	459
15.9.4.86 ChunkLineStatusAll	460
15.9.4.87 ChunkModeActive	460
15.9.4.88 ChunkOffsetX	460
15.9.4.89 ChunkOffsetY	460
15.9.4.90 ChunkPartSelector	460
15.9.4.91 ChunkPixelDynamicRangeMax	460
15.9.4.92 ChunkPixelDynamicRangeMin	461
15.9.4.93 ChunkPixelFormat	461
15.9.4.94 ChunkRegionID	461
15.9.4.95 ChunkScan3dAxisMax	461
15.9.4.96 ChunkScan3dAxisMin	461
15.9.4.97 ChunkScan3dCoordinateOffset	461
15.9.4.98 ChunkScan3dCoordinateReferenceSelector	462
15.9.4.99 ChunkScan3dCoordinateReferenceValue	462
15.9.4.100 ChunkScan3dCoordinateScale	462
15.9.4.101 ChunkScan3dCoordinateSelector	462
15.9.4.102 ChunkScan3dCoordinateSystem	462
15.9.4.103 ChunkScan3dCoordinateSystemReference	462
15.9.4.104 ChunkScan3dCoordinateTransformSelector	463
15.9.4.105 ChunkScan3dDistanceUnit	463
15.9.4.106 ChunkScan3dInvalidDataFlag	463
15.9.4.107 ChunkScan3dInvalidDataValue	463
15.9.4.108 ChunkScan3dOutputMode	463
15.9.4.109 ChunkScan3dTransformValue	463
15.9.4.110 ChunkScanLineSelector	464
15.9.4.111 ChunkSelector	464
15.9.4.112 ChunkSequencerSetActive	464
15.9.4.113 ChunkSerialData	464
15.9.4.114 ChunkSerialDataLength	464

15.9.4.115 ChunkSerialReceiveOverflow	464
15.9.4.116 ChunkSourceID	465
15.9.4.117 ChunkStreamChannelID	465
15.9.4.118 ChunkTimerSelector	465
15.9.4.119 ChunkTimerValue	465
15.9.4.120 ChunkTimestamp	465
15.9.4.121 ChunkTimestampLatchValue	465
15.9.4.122 ChunkTransferBlockID	466
15.9.4.123 ChunkTransferQueueCurrentBlockCount	466
15.9.4.124 ChunkTransferStreamID	466
15.9.4.125 ChunkWidth	466
15.9.4.126 CIConfiguration	466
15.9.4.127 CITimeSlotsCount	466
15.9.4.128 ColorTransformationEnable	467
15.9.4.129 ColorTransformationSelector	467
15.9.4.130 ColorTransformationValue	467
15.9.4.131 ColorTransformationValueSelector	467
15.9.4.132 CompressionRatio	468
15.9.4.133 CompressionSaturationPriority	468
15.9.4.134 CounterDelay	468
15.9.4.135 CounterDuration	468
15.9.4.136 CounterEventActivation	468
15.9.4.137 CounterEventSource	468
15.9.4.138 CounterReset	469
15.9.4.139 CounterResetActivation	469
15.9.4.140 CounterResetSource	469
15.9.4.141 CounterSelector	469
15.9.4.142 CounterStatus	469
15.9.4.143 CounterTriggerActivation	469
15.9.4.144 CounterTriggerSource	470
15.9.4.145 CounterValue	470
15.9.4.146 CounterValueAtReset	470
15.9.4.147 CxpConnectionSelector	470
15.9.4.148 CxpConnectionTestErrorCount	470
15.9.4.149 CxpConnectionTestMode	470
15.9.4.150 CxpConnectionTestPacketCount	471
15.9.4.151 CxpLinkConfiguration	471
15.9.4.152 CxpLinkConfigurationPreferred	471
15.9.4.153 CxpLinkConfigurationStatus	471
15.9.4.154 CxpPoCxpAuto	471
15.9.4.155 CxpPoCxpStatus	471
15.9.4.156 CxpPoCxpTripReset	472

15.9.4.157 CxpPoCxpTurnOff	472
15.9.4.158 DecimationHorizontal	472
15.9.4.159 DecimationHorizontalMode	472
15.9.4.160 DecimationSelector	472
15.9.4.161 DecimationVertical	473
15.9.4.162 DecimationVerticalMode	473
15.9.4.163 DefectCorrectionMode	473
15.9.4.164 DefectCorrectStaticEnable	473
15.9.4.165 DefectTableApply	474
15.9.4.166 DefectTableCoordinateX	474
15.9.4.167 DefectTableCoordinateY	474
15.9.4.168 DefectTableFactoryRestore	474
15.9.4.169 DefectTableIndex	475
15.9.4.170 DefectTablePixelCount	475
15.9.4.171 DefectTableSave	475
15.9.4.172 Deinterlacing	475
15.9.4.173 DeviceCharacterSet	476
15.9.4.174 DeviceClockFrequency	476
15.9.4.175 DeviceClockSelector	476
15.9.4.176 DeviceConnectionSelector	476
15.9.4.177 DeviceConnectionSpeed	476
15.9.4.178 DeviceConnectionStatus	477
15.9.4.179 DeviceEventChannelCount	477
15.9.4.180 DeviceFamilyName	477
15.9.4.181 DeviceFeaturePersistenceEnd	477
15.9.4.182 DeviceFeaturePersistenceStart	477
15.9.4.183 DeviceFirmwareVersion	478
15.9.4.184 DeviceGenCPVersionMajor	478
15.9.4.185 DeviceGenCPVersionMinor	478
15.9.4.186 DeviceID	478
15.9.4.187 DeviceIndicatorMode	478
15.9.4.188 DeviceLinkBandwidthReserve	478
15.9.4.189 DeviceLinkCommandTimeout	479
15.9.4.190 DeviceLinkConnectionCount	479
15.9.4.191 DeviceLinkCurrentThroughput	479
15.9.4.192 DeviceLinkHeartbeatMode	479
15.9.4.193 DeviceLinkHeartbeatTimeout	479
15.9.4.194 DeviceLinkSelector	479
15.9.4.195 DeviceLinkSpeed	480
15.9.4.196 DeviceLinkThroughputLimit	480
15.9.4.197 DeviceLinkThroughputLimitMode	480
15.9.4.198 DeviceManifestEntrySelector	480

15.9.4.199 DeviceManifestPrimaryURL	480
15.9.4.200 DeviceManifestSchemaMajorVersion	481
15.9.4.201 DeviceManifestSchemaMinorVersion	481
15.9.4.202 DeviceManifestSecondaryURL	481
15.9.4.203 DeviceManifestXMLMajorVersion	481
15.9.4.204 DeviceManifestXMLMinorVersion	481
15.9.4.205 DeviceManifestXMLSubMinorVersion	481
15.9.4.206 DeviceManufacturerInfo	482
15.9.4.207 DeviceMaxThroughput	482
15.9.4.208 DeviceModelName	482
15.9.4.209 DevicePowerSupplySelector	482
15.9.4.210 DeviceRegistersCheck	482
15.9.4.211 DeviceRegistersEndianness	483
15.9.4.212 DeviceRegistersStreamingEnd	483
15.9.4.213 DeviceRegistersStreamingStart	483
15.9.4.214 DeviceRegistersValid	483
15.9.4.215 DeviceReset	483
15.9.4.216 DeviceScanType	483
15.9.4.217 DeviceSerialNumber	484
15.9.4.218 DeviceSerialPortBaudRate	484
15.9.4.219 DeviceSerialPortSelector	484
15.9.4.220 DeviceSFNCVersionMajor	484
15.9.4.221 DeviceSFNCVersionMinor	484
15.9.4.222 DeviceSFNCVersionSubMinor	485
15.9.4.223 DeviceStreamChannelCount	485
15.9.4.224 DeviceStreamChannelEndianness	485
15.9.4.225 DeviceStreamChannelLink	485
15.9.4.226 DeviceStreamChannelPacketSize	485
15.9.4.227 DeviceStreamChannelSelector	486
15.9.4.228 DeviceStreamChannelType	486
15.9.4.229 DeviceTapGeometry	486
15.9.4.230 DeviceTemperature	486
15.9.4.231 DeviceTemperatureSelector	486
15.9.4.232 DeviceTLType	487
15.9.4.233 DeviceTLVersionMajor	487
15.9.4.234 DeviceTLVersionMinor	487
15.9.4.235 DeviceTLVersionSubMinor	487
15.9.4.236 DeviceType	487
15.9.4.237 DeviceUptime	488
15.9.4.238 DeviceUserID	488
15.9.4.239 DeviceVendorName	488
15.9.4.240 DeviceVersion	488

15.9.4.241 EncoderDivider	488
15.9.4.242 EncoderMode	488
15.9.4.243 EncoderOutputMode	489
15.9.4.244 EncoderReset	489
15.9.4.245 EncoderResetActivation	489
15.9.4.246 EncoderResetSource	489
15.9.4.247 EncoderSelector	489
15.9.4.248 EncoderSourceA	489
15.9.4.249 EncoderSourceB	490
15.9.4.250 EncoderStatus	490
15.9.4.251 EncoderTimeout	490
15.9.4.252 EncoderValue	490
15.9.4.253 EncoderValueAtReset	490
15.9.4.254 EnumerationCount	490
15.9.4.255 EventAcquisitionEnd	491
15.9.4.256 EventAcquisitionEndFrameID	491
15.9.4.257 EventAcquisitionEndTimestamp	491
15.9.4.258 EventAcquisitionError	491
15.9.4.259 EventAcquisitionErrorFrameID	491
15.9.4.260 EventAcquisitionErrorTimestamp	491
15.9.4.261 EventAcquisitionStart	492
15.9.4.262 EventAcquisitionStartFrameID	492
15.9.4.263 EventAcquisitionStartTimestamp	492
15.9.4.264 EventAcquisitionTransferEnd	492
15.9.4.265 EventAcquisitionTransferEndFrameID	492
15.9.4.266 EventAcquisitionTransferEndTimestamp	492
15.9.4.267 EventAcquisitionTransferStart	493
15.9.4.268 EventAcquisitionTransferStartFrameID	493
15.9.4.269 EventAcquisitionTransferStartTimestamp	493
15.9.4.270 EventAcquisitionTrigger	493
15.9.4.271 EventAcquisitionTriggerFrameID	493
15.9.4.272 EventAcquisitionTriggerTimestamp	493
15.9.4.273 EventActionLate	494
15.9.4.274 EventActionLateFrameID	494
15.9.4.275 EventActionLateTimestamp	494
15.9.4.276 EventCounter0End	494
15.9.4.277 EventCounter0EndFrameID	494
15.9.4.278 EventCounter0EndTimestamp	494
15.9.4.279 EventCounter0Start	495
15.9.4.280 EventCounter0StartFrameID	495
15.9.4.281 EventCounter0StartTimestamp	495
15.9.4.282 EventCounter1End	495

15.9.4.283 EventCounter1EndFrameID	495
15.9.4.284 EventCounter1EndTimestamp	495
15.9.4.285 EventCounter1Start	496
15.9.4.286 EventCounter1StartFrameID	496
15.9.4.287 EventCounter1StartTimestamp	496
15.9.4.288 EventEncoder0Restarted	496
15.9.4.289 EventEncoder0RestartedFrameID	496
15.9.4.290 EventEncoder0RestartedTimestamp	496
15.9.4.291 EventEncoder0Stopped	497
15.9.4.292 EventEncoder0StoppedFrameID	497
15.9.4.293 EventEncoder0StoppedTimestamp	497
15.9.4.294 EventEncoder1Restarted	497
15.9.4.295 EventEncoder1RestartedFrameID	497
15.9.4.296 EventEncoder1RestartedTimestamp	497
15.9.4.297 EventEncoder1Stopped	498
15.9.4.298 EventEncoder1StoppedFrameID	498
15.9.4.299 EventEncoder1StoppedTimestamp	498
15.9.4.300 EventError	498
15.9.4.301 EventErrorCode	498
15.9.4.302 EventErrorFrameID	498
15.9.4.303 EventErrorTimestamp	499
15.9.4.304 EventExposureEnd	499
15.9.4.305 EventExposureEndFrameID	499
15.9.4.306 EventExposureEndTimestamp	499
15.9.4.307 EventExposureStart	499
15.9.4.308 EventExposureStartFrameID	499
15.9.4.309 EventExposureStartTimestamp	500
15.9.4.310 EventFrameBurstEnd	500
15.9.4.311 EventFrameBurstEndFrameID	500
15.9.4.312 EventFrameBurstEndTimestamp	500
15.9.4.313 EventFrameBurstStart	500
15.9.4.314 EventFrameBurstStartFrameID	500
15.9.4.315 EventFrameBurstStartTimestamp	501
15.9.4.316 EventFrameEnd	501
15.9.4.317 EventFrameEndFrameID	501
15.9.4.318 EventFrameEndTimestamp	501
15.9.4.319 EventFrameStart	501
15.9.4.320 EventFrameStartFrameID	501
15.9.4.321 EventFrameStartTimestamp	502
15.9.4.322 EventFrameTransferEnd	502
15.9.4.323 EventFrameTransferEndFrameID	502
15.9.4.324 EventFrameTransferEndTimestamp	502

15.9.4.325 EventFrameTransferStart	502
15.9.4.326 EventFrameTransferStartFrameID	502
15.9.4.327 EventFrameTransferStartTimestamp	503
15.9.4.328 EventFrameTrigger	503
15.9.4.329 EventFrameTriggerFrameID	503
15.9.4.330 EventFrameTriggerTimestamp	503
15.9.4.331 EventLine0AnyEdge	503
15.9.4.332 EventLine0AnyEdgeFrameID	503
15.9.4.333 EventLine0AnyEdgeTimestamp	504
15.9.4.334 EventLine0FallingEdge	504
15.9.4.335 EventLine0FallingEdgeFrameID	504
15.9.4.336 EventLine0FallingEdgeTimestamp	504
15.9.4.337 EventLine0RisingEdge	504
15.9.4.338 EventLine0RisingEdgeFrameID	504
15.9.4.339 EventLine0RisingEdgeTimestamp	505
15.9.4.340 EventLine1AnyEdge	505
15.9.4.341 EventLine1AnyEdgeFrameID	505
15.9.4.342 EventLine1AnyEdgeTimestamp	505
15.9.4.343 EventLine1FallingEdge	505
15.9.4.344 EventLine1FallingEdgeFrameID	505
15.9.4.345 EventLine1FallingEdgeTimestamp	506
15.9.4.346 EventLine1RisingEdge	506
15.9.4.347 EventLine1RisingEdgeFrameID	506
15.9.4.348 EventLine1RisingEdgeTimestamp	506
15.9.4.349 EventLinkSpeedChange	506
15.9.4.350 EventLinkSpeedChangeFrameID	506
15.9.4.351 EventLinkSpeedChangeTimestamp	507
15.9.4.352 EventLinkTrigger0	507
15.9.4.353 EventLinkTrigger0FrameID	507
15.9.4.354 EventLinkTrigger0Timestamp	507
15.9.4.355 EventLinkTrigger1	507
15.9.4.356 EventLinkTrigger1FrameID	507
15.9.4.357 EventLinkTrigger1Timestamp	508
15.9.4.358 EventNotification	508
15.9.4.359 EventSelector	508
15.9.4.360 EventSequencerSetChange	508
15.9.4.361 EventSequencerSetChangeFrameID	508
15.9.4.362 EventSequencerSetChangeTimestamp	508
15.9.4.363 EventSerialData	509
15.9.4.364 EventSerialDataLength	509
15.9.4.365 EventSerialPortReceive	509
15.9.4.366 EventSerialPortReceiveTimestamp	509

15.9.4.367 EventSerialReceiveOverflow	509
15.9.4.368 EventStream0TransferBlockEnd	509
15.9.4.369 EventStream0TransferBlockEndFrameID	510
15.9.4.370 EventStream0TransferBlockEndTimestamp	510
15.9.4.371 EventStream0TransferBlockStart	510
15.9.4.372 EventStream0TransferBlockStartFrameID	510
15.9.4.373 EventStream0TransferBlockStartTimestamp	510
15.9.4.374 EventStream0TransferBlockTrigger	510
15.9.4.375 EventStream0TransferBlockTriggerFrameID	511
15.9.4.376 EventStream0TransferBlockTriggerTimestamp	511
15.9.4.377 EventStream0TransferBurstEnd	511
15.9.4.378 EventStream0TransferBurstEndFrameID	511
15.9.4.379 EventStream0TransferBurstEndTimestamp	511
15.9.4.380 EventStream0TransferBurstStart	511
15.9.4.381 EventStream0TransferBurstStartFrameID	512
15.9.4.382 EventStream0TransferBurstStartTimestamp	512
15.9.4.383 EventStream0TransferEnd	512
15.9.4.384 EventStream0TransferEndFrameID	512
15.9.4.385 EventStream0TransferEndTimestamp	512
15.9.4.386 EventStream0TransferOverflow	512
15.9.4.387 EventStream0TransferOverflowFrameID	513
15.9.4.388 EventStream0TransferOverflowTimestamp	513
15.9.4.389 EventStream0TransferPause	513
15.9.4.390 EventStream0TransferPauseFrameID	513
15.9.4.391 EventStream0TransferPauseTimestamp	513
15.9.4.392 EventStream0TransferResume	513
15.9.4.393 EventStream0TransferResumeFrameID	514
15.9.4.394 EventStream0TransferResumeTimestamp	514
15.9.4.395 EventStream0TransferStart	514
15.9.4.396 EventStream0TransferStartFrameID	514
15.9.4.397 EventStream0TransferStartTimestamp	514
15.9.4.398 EventTest	514
15.9.4.399 EventTestTimestamp	515
15.9.4.400 EventTimer0End	515
15.9.4.401 EventTimer0EndFrameID	515
15.9.4.402 EventTimer0EndTimestamp	515
15.9.4.403 EventTimer0Start	515
15.9.4.404 EventTimer0StartFrameID	515
15.9.4.405 EventTimer0StartTimestamp	516
15.9.4.406 EventTimer1End	516
15.9.4.407 EventTimer1EndFrameID	516
15.9.4.408 EventTimer1EndTimestamp	516

15.9.4.409 EventTimer1Start	516
15.9.4.410 EventTimer1StartFrameID	516
15.9.4.411 EventTimer1StartTimestamp	517
15.9.4.412 ExposureActiveMode	517
15.9.4.413 ExposureAuto	517
15.9.4.414 ExposureMode	517
15.9.4.415 ExposureTime	517
15.9.4.416 ExposureTimeMode	518
15.9.4.417 ExposureTimeSelector	518
15.9.4.418 FactoryReset	518
15.9.4.419 FileAccessBuffer	518
15.9.4.420 FileAccessLength	518
15.9.4.421 FileAccessOffset	518
15.9.4.422 FileOpenMode	519
15.9.4.423 FileOperationExecute	519
15.9.4.424 FileOperationResult	519
15.9.4.425 FileOperationSelector	519
15.9.4.426 FileOperationStatus	520
15.9.4.427 FileSelector	520
15.9.4.428 FileSize	520
15.9.4.429 Gain	520
15.9.4.430 GainAuto	520
15.9.4.431 GainAutoBalance	521
15.9.4.432 GainSelector	521
15.9.4.433 Gamma	521
15.9.4.434 GammaEnable	521
15.9.4.435 GevActiveLinkCount	521
15.9.4.436 GevCCP	521
15.9.4.437 GevCurrentDefaultGateway	522
15.9.4.438 GevCurrentIPAddress	522
15.9.4.439 GevCurrentIPConfigurationDHCP	522
15.9.4.440 GevCurrentIPConfigurationLLA	522
15.9.4.441 GevCurrentIPConfigurationPersistentIP	522
15.9.4.442 GevCurrentPhysicalLinkConfiguration	522
15.9.4.443 GevCurrentSubnetMask	523
15.9.4.444 GevDiscoveryAckDelay	523
15.9.4.445 GevFirstURL	523
15.9.4.446 GevGVCPExtendedStatusCodes	523
15.9.4.447 GevGVCPExtendedStatusCodesSelector	523
15.9.4.448 GevGVCPHeartbeatDisable	523
15.9.4.449 GevGVCPPendingAck	524
15.9.4.450 GevGVCPPendingTimeout	524

15.9.4.451	GevGVSPExtendedIDMode	524
15.9.4.452	GevHeartbeatTimeout	524
15.9.4.453	GevIEEE1588	524
15.9.4.454	GevIEEE1588ClockAccuracy	524
15.9.4.455	GevIEEE1588Mode	525
15.9.4.456	GevIEEE1588Status	525
15.9.4.457	GevInterfaceSelector	525
15.9.4.458	GevIPConfigurationStatus	525
15.9.4.459	GevMACAddress	525
15.9.4.460	GevMCDA	525
15.9.4.461	GevMCPHostPort	526
15.9.4.462	GevMCRC	526
15.9.4.463	GevMCSP	526
15.9.4.464	GevMCTT	526
15.9.4.465	GevNumberOfInterfaces	526
15.9.4.466	GevPAUSEFrameReception	526
15.9.4.467	GevPAUSEFrameTransmission	527
15.9.4.468	GevPersistentDefaultGateway	527
15.9.4.469	GevPersistentIPAddress	527
15.9.4.470	GevPersistentSubnetMask	527
15.9.4.471	GevPhysicalLinkConfiguration	527
15.9.4.472	GevPrimaryApplicationIPAddress	527
15.9.4.473	GevPrimaryApplicationSocket	528
15.9.4.474	GevPrimaryApplicationSwitchoverKey	528
15.9.4.475	GevSCCFGAllInTransmission	528
15.9.4.476	GevSCCFGExtendedChunkData	528
15.9.4.477	GevSCCFGPacketResendDestination	528
15.9.4.478	GevSCCFGUnconditionalStreaming	528
15.9.4.479	GevSCDA	529
15.9.4.480	GevSCPD	529
15.9.4.481	GevSCPDDirection	529
15.9.4.482	GevSCPHostPort	529
15.9.4.483	GevSCPInterfaceIndex	529
15.9.4.484	GevSCPSBigEndian	529
15.9.4.485	GevSCPSDoNotFragment	530
15.9.4.486	GevSCPSFireTestPacket	530
15.9.4.487	GevSCPSPacketSize	530
15.9.4.488	GevSCSP	530
15.9.4.489	GevSCZoneConfigurationLock	530
15.9.4.490	GevSCZoneCount	530
15.9.4.491	GevSCZoneDirectionAll	531
15.9.4.492	GevSecondURL	531

15.9.4.493	GevStreamChannelSelector	531
15.9.4.494	GevSupportedOption	531
15.9.4.495	GevSupportedOptionSelector	531
15.9.4.496	GevTimestampTickFrequency	531
15.9.4.497	GuiXmlManifestAddress	532
15.9.4.498	Height	532
15.9.4.499	HeightMax	532
15.9.4.500	ImageComponentEnable	532
15.9.4.501	ImageComponentSelector	532
15.9.4.502	ImageCompressionBitrate	533
15.9.4.503	ImageCompressionJPEGFormatOption	533
15.9.4.504	ImageCompressionMode	533
15.9.4.505	ImageCompressionQuality	533
15.9.4.506	ImageCompressionRateOption	533
15.9.4.507	IspEnable	533
15.9.4.508	LineFilterWidth	534
15.9.4.509	LineFormat	534
15.9.4.510	LineInputFilterSelector	534
15.9.4.511	LineInverter	534
15.9.4.512	LineMode	534
15.9.4.513	LinePitch	534
15.9.4.514	LineSelector	535
15.9.4.515	LineSource	535
15.9.4.516	LineStatus	535
15.9.4.517	LineStatusAll	535
15.9.4.518	LinkErrorCount	535
15.9.4.519	LinkUptime	535
15.9.4.520	LogicBlockLUTInputActivation	536
15.9.4.521	LogicBlockLUTInputSelector	536
15.9.4.522	LogicBlockLUTInputSource	536
15.9.4.523	LogicBlockLUTOutputValue	536
15.9.4.524	LogicBlockLUTOutputValueAll	536
15.9.4.525	LogicBlockLUTRowIndex	536
15.9.4.526	LogicBlockLUTSelector	537
15.9.4.527	LogicBlockSelector	537
15.9.4.528	LUTEnable	537
15.9.4.529	LUTIndex	537
15.9.4.530	LUTSelector	537
15.9.4.531	LUTValue	538
15.9.4.532	LUTValueAll	538
15.9.4.533	MaxDeviceResetTime	538
15.9.4.534	OffsetX	538

15.9.4.535 OffsetY	538
15.9.4.536 PacketResendRequestCount	539
15.9.4.537 PayloadSize	539
15.9.4.538 PixelColorFilter	539
15.9.4.539 PixelDynamicRangeMax	539
15.9.4.540 PixelDynamicRangeMin	539
15.9.4.541 PixelFormat	540
15.9.4.542 PixelFormatInFold	540
15.9.4.543 PixelFormatInfoSelector	540
15.9.4.544 PixelSize	540
15.9.4.545 PowerSupplyCurrent	540
15.9.4.546 PowerSupplyVoltage	541
15.9.4.547 RegionDestination	541
15.9.4.548 RegionMode	541
15.9.4.549 RegionSelector	541
15.9.4.550 ReverseX	541
15.9.4.551 ReverseY	542
15.9.4.552 RgbTransformLightSource	542
15.9.4.553 Saturation	542
15.9.4.554 SaturationEnable	542
15.9.4.555 Scan3dAxisMax	542
15.9.4.556 Scan3dAxisMin	543
15.9.4.557 Scan3dCoordinateOffset	543
15.9.4.558 Scan3dCoordinateReferenceSelector	543
15.9.4.559 Scan3dCoordinateReferenceValue	543
15.9.4.560 Scan3dCoordinateScale	543
15.9.4.561 Scan3dCoordinateSelector	543
15.9.4.562 Scan3dCoordinateSystem	544
15.9.4.563 Scan3dCoordinateSystemReference	544
15.9.4.564 Scan3dCoordinateTransformSelector	544
15.9.4.565 Scan3dDistanceUnit	544
15.9.4.566 Scan3dInvalidDataFlag	544
15.9.4.567 Scan3dInvalidDataValue	544
15.9.4.568 Scan3dOutputMode	545
15.9.4.569 Scan3dTransformValue	545
15.9.4.570 SensorDescription	545
15.9.4.571 SensorDigitizationTaps	545
15.9.4.572 SensorHeight	545
15.9.4.573 SensorShutterMode	545
15.9.4.574 SensorTaps	546
15.9.4.575 SensorWidth	546
15.9.4.576 SequencerConfigurationMode	546

15.9.4.577 SequencerConfigurationValid	546
15.9.4.578 SequencerFeatureEnable	546
15.9.4.579 SequencerMode	547
15.9.4.580 SequencerPathSelector	547
15.9.4.581 SequencerSetActive	547
15.9.4.582 SequencerSetLoad	547
15.9.4.583 SequencerSetNext	547
15.9.4.584 SequencerSetSave	548
15.9.4.585 SequencerSetSelector	548
15.9.4.586 SequencerSetStart	548
15.9.4.587 SequencerSetValid	548
15.9.4.588 SequencerTriggerActivation	549
15.9.4.589 SequencerTriggerSource	549
15.9.4.590 SerialPortBaudRate	549
15.9.4.591 SerialPortDataBits	549
15.9.4.592 SerialPortParity	549
15.9.4.593 SerialPortSelector	550
15.9.4.594 SerialPortSource	550
15.9.4.595 SerialPortStopBits	550
15.9.4.596 SerialReceiveFramingErrorCount	550
15.9.4.597 SerialReceiveParityErrorCount	550
15.9.4.598 SerialReceiveQueueClear	550
15.9.4.599 SerialReceiveQueueCurrentCharacterCount	551
15.9.4.600 SerialReceiveQueueMaxCharacterCount	551
15.9.4.601 SerialTransmitQueueCurrentCharacterCount	551
15.9.4.602 SerialTransmitQueueMaxCharacterCount	551
15.9.4.603 Sharpening	551
15.9.4.604 SharpeningAuto	552
15.9.4.605 SharpeningEnable	552
15.9.4.606 SharpeningThreshold	552
15.9.4.607 SoftwareSignalPulse	552
15.9.4.608 SoftwareSignalSelector	553
15.9.4.609 SourceCount	553
15.9.4.610 SourceSelector	553
15.9.4.611 Test0001	553
15.9.4.612 TestEventGenerate	553
15.9.4.613 TestPattern	553
15.9.4.614 TestPatternGeneratorSelector	554
15.9.4.615 TestPendingAck	554
15.9.4.616 TimerDelay	554
15.9.4.617 TimerDuration	554
15.9.4.618 TimerReset	554

15.9.4.619 TimerSelector	555
15.9.4.620 TimerStatus	555
15.9.4.621 TimerTriggerActivation	555
15.9.4.622 TimerTriggerSource	555
15.9.4.623 TimerValue	555
15.9.4.624 Timestamp	555
15.9.4.625 TimestampLatch	556
15.9.4.626 TimestampLatchValue	556
15.9.4.627 TimestampReset	556
15.9.4.628 TLParamsLocked	556
15.9.4.629 TransferAbort	556
15.9.4.630 TransferBlockCount	556
15.9.4.631 TransferBurstCount	557
15.9.4.632 TransferComponentSelector	557
15.9.4.633 TransferControlMode	557
15.9.4.634 TransferOperationMode	557
15.9.4.635 TransferPause	557
15.9.4.636 TransferQueueCurrentBlockCount	557
15.9.4.637 TransferQueueMaxBlockCount	558
15.9.4.638 TransferQueueMode	558
15.9.4.639 TransferQueueOverflowCount	558
15.9.4.640 TransferResume	558
15.9.4.641 TransferSelector	558
15.9.4.642 TransferStart	558
15.9.4.643 TransferStatus	559
15.9.4.644 TransferStatusSelector	559
15.9.4.645 TransferStop	559
15.9.4.646 TransferStreamChannel	559
15.9.4.647 TransferTriggerActivation	559
15.9.4.648 TransferTriggerMode	559
15.9.4.649 TransferTriggerSelector	560
15.9.4.650 TransferTriggerSource	560
15.9.4.651 TriggerActivation	560
15.9.4.652 TriggerDelay	560
15.9.4.653 TriggerDivider	560
15.9.4.654 TriggerEventTest	561
15.9.4.655 TriggerMode	561
15.9.4.656 TriggerMultiplier	561
15.9.4.657 TriggerOverlap	561
15.9.4.658 TriggerSelector	561
15.9.4.659 TriggerSoftware	562
15.9.4.660 TriggerSource	562

15.9.4.661 UserOutputSelector	562
15.9.4.662 UserOutputValue	562
15.9.4.663 UserOutputValueAll	562
15.9.4.664 UserOutputValueAllMask	563
15.9.4.665 UserSetDefault	563
15.9.4.666 UserSetFeatureEnable	563
15.9.4.667 UserSetLoad	563
15.9.4.668 UserSetSave	564
15.9.4.669 UserSetSelector	564
15.9.4.670 V3_3Enable	564
15.9.4.671 WhiteClip	564
15.9.4.672 WhiteClipSelector	564
15.9.4.673 Width	565
15.9.4.674 WidthMax	565
15.10 CameraBase Class Reference	565
15.10.1 Detailed Description	568
15.10.2 Constructor & Destructor Documentation	568
15.10.2.1 ~CameraBase()	568
15.10.2.2 CameraBase() [1/2]	568
15.10.2.3 CameraBase() [2/2]	568
15.10.3 Member Function Documentation	568
15.10.3.1 BeginAcquisition()	569
15.10.3.2 DelInit()	569
15.10.3.3 DiscoverMaxPacketSize()	569
15.10.3.4 EndAcquisition()	570
15.10.3.5 ForceIP()	570
15.10.3.6 GetAccessMode()	570
15.10.3.7 GetBufferOwnership()	571
15.10.3.8 GetGuiXml()	571
15.10.3.9 GetNextImage()	571
15.10.3.10 GetNextImageSync()	572
15.10.3.11 GetNodeMap()	572
15.10.3.12 GetNumDataStreams()	573
15.10.3.13 GetNumImagesInUse()	573
15.10.3.14 GetTLDeviceNodeMap()	573
15.10.3.15 GetTLStreamNodeMap()	574
15.10.3.16 GetUniqueID()	574
15.10.3.17 GetUserBufferCount()	574
15.10.3.18 GetUserBufferSize()	575
15.10.3.19 GetUserBufferTotalSize()	575
15.10.3.20 Init()	576
15.10.3.21 IsInitialized()	576

15.10.3.22 IsStreaming()	576
15.10.3.23 IsValid()	577
15.10.3.24 operator=()	577
15.10.3.25 ReadPort()	577
15.10.3.26 RegisterEventHandler() [1/3]	577
15.10.3.27 RegisterEventHandler() [2/3]	578
15.10.3.28 RegisterEventHandler() [3/3]	578
15.10.3.29 SetBufferOwnership()	579
15.10.3.30 SetUserBuffers() [1/2]	579
15.10.3.31 SetUserBuffers() [2/2]	580
15.10.3.32 UnregisterEventHandler()	580
15.10.3.33 WritePort()	581
15.10.4 Friends And Related Function Documentation	581
15.10.4.1 InterfacImpl	581
15.11 CameraList Class Reference	581
15.11.1 Detailed Description	583
15.11.2 Constructor & Destructor Documentation	583
15.11.2.1 CameraList() [1/2]	583
15.11.2.2 ~CameraList()	583
15.11.2.3 CameraList() [2/2]	583
15.11.3 Member Function Documentation	583
15.11.3.1 Add()	583
15.11.3.2 Append()	584
15.11.3.3 Clear()	584
15.11.3.4 GetByDeviceID()	584
15.11.3.5 GetByIndex()	585
15.11.3.6 GetBySerial()	585
15.11.3.7 GetSize()	586
15.11.3.8 operator=()	586
15.11.3.9 operator[]()	586
15.11.3.10 Remove()	586
15.11.3.11 RemoveByDeviceID()	587
15.11.3.12 RemoveByIndex()	587
15.11.3.13 RemoveBySerial()	587
15.12 CameraPtr Class Reference	588
15.12.1 Detailed Description	589
15.12.2 Constructor & Destructor Documentation	589
15.12.2.1 CameraPtr() [1/4]	589
15.12.2.2 CameraPtr() [2/4]	589
15.12.2.3 CameraPtr() [3/4]	589
15.12.2.4 CameraPtr() [4/4]	590
15.13 CategoryNode Class Reference	590

15.13.1 Detailed Description	591
15.13.2 Constructor & Destructor Documentation	591
15.13.2.1 CategoryNode() [1/2]	591
15.13.2.2 CategoryNode() [2/2]	592
15.13.2.3 ~CategoryNode()	592
15.13.3 Member Function Documentation	592
15.13.3.1 GetFeatures()	592
15.13.3.2 SetReference()	592
15.14 CChunkAdapter Class Reference	592
15.14.1 Detailed Description	593
15.14.2 Constructor & Destructor Documentation	593
15.14.2.1 ~CChunkAdapter()	593
15.14.2.2 CChunkAdapter()	594
15.14.3 Member Function Documentation	594
15.14.3.1 AttachBuffer()	594
15.14.3.2 AttachNodeMap()	594
15.14.3.3 CheckBufferLayout()	594
15.14.3.4 ClearCaches()	595
15.14.3.5 DetachBuffer()	595
15.14.3.6 DetachNodeMap()	595
15.14.3.7 UpdateBuffer()	595
15.14.4 Member Data Documentation	595
15.14.4.1 m_pChunkAdapter	595
15.15 CChunkAdapterDcam Class Reference	596
15.15.1 Detailed Description	597
15.15.2 Constructor & Destructor Documentation	597
15.15.2.1 CChunkAdapterDcam()	597
15.15.2.2 ~CChunkAdapterDcam()	597
15.15.3 Member Function Documentation	597
15.15.3.1 AttachBuffer()	597
15.15.3.2 CheckBufferLayout()	598
15.15.3.3 CheckCRC()	598
15.15.3.4 HasCRC()	598
15.16 CChunkAdapterGeneric Class Reference	598
15.16.1 Constructor & Destructor Documentation	599
15.16.1.1 CChunkAdapterGeneric()	599
15.16.1.2 ~CChunkAdapterGeneric()	599
15.16.2 Member Function Documentation	600
15.16.2.1 AttachBuffer() [1/3]	600
15.16.2.2 AttachBuffer() [2/3]	600
15.16.2.3 AttachBuffer() [3/3]	600
15.16.2.4 CheckBufferLayout()	600

15.17 CChunkAdapterGEV Class Reference	601
15.17.1 Detailed Description	601
15.17.2 Constructor & Destructor Documentation	602
15.17.2.1 CChunkAdapterGEV()	602
15.17.2.2 ~CChunkAdapterGEV()	602
15.17.3 Member Function Documentation	602
15.17.3.1 AttachBuffer()	602
15.17.3.2 CheckBufferLayout()	602
15.18 CChunkAdapterU3V Class Reference	603
15.18.1 Detailed Description	603
15.18.2 Constructor & Destructor Documentation	604
15.18.2.1 CChunkAdapterU3V()	604
15.18.2.2 ~CChunkAdapterU3V()	604
15.18.3 Member Function Documentation	604
15.18.3.1 AttachBuffer()	604
15.18.3.2 CheckBufferLayout()	604
15.19 CChunkPort Class Reference	605
15.19.1 Detailed Description	606
15.19.2 Constructor & Destructor Documentation	606
15.19.2.1 CChunkPort()	606
15.19.2.2 ~CChunkPort()	607
15.19.3 Member Function Documentation	607
15.19.3.1 AttachChunk()	607
15.19.3.2 AttachPort()	607
15.19.3.3 CheckChunkID() [1/2]	607
15.19.3.4 CheckChunkID() [2/2]	607
15.19.3.5 ClearCache()	608
15.19.3.6 DetachChunk()	608
15.19.3.7 DetachPort()	608
15.19.3.8 GetAccessMode()	608
15.19.3.9 GetChunkIDLength()	608
15.19.3.10 GetPrincipalInterfaceType()	608
15.19.3.11 GetSwapEndianess()	609
15.19.3.12 InvalidateNode()	609
15.19.3.13 Read()	609
15.19.3.14 SetPortImpl()	609
15.19.3.15 UpdateBuffer()	609
15.19.3.16 Write()	609
15.19.4 Member Data Documentation	610
15.19.4.1 m_pChunkPort	610
15.19.4.2 m_pPort	610
15.19.4.3 m_pPortAdapter	610

15.20 CCMSettings Struct Reference	610
15.20.1 Constructor & Destructor Documentation	610
15.20.1.1 CCMSettings()	611
15.20.2 Member Data Documentation	611
15.20.2.1 Application	611
15.20.2.2 ColorSpace	611
15.20.2.3 ColorTemperature	611
15.20.2.4 CustomCCMCode	611
15.20.2.5 Sensor	611
15.20.2.6 Type	611
15.21 CEnumerationTRef< EnumT > Class Template Reference	612
15.21.1 Detailed Description	614
15.21.2 Constructor & Destructor Documentation	614
15.21.2.1 CEnumerationTRef() [1/2]	614
15.21.2.2 CEnumerationTRef() [2/2]	614
15.21.2.3 ~CEnumerationTRef()	614
15.21.3 Member Function Documentation	614
15.21.3.1 GetCurrentEntry()	615
15.21.3.2 GetEntry() [1/2]	615
15.21.3.3 GetEntry() [2/2]	615
15.21.3.4 GetValue()	615
15.21.3.5 operator()()	616
15.21.3.6 operator=() [1/2]	616
15.21.3.7 operator=() [2/2]	616
15.21.3.8 SetEnumReference()	616
15.21.3.9 SetNumEnums()	616
15.21.3.10 SetReference()	617
15.21.3.11 SetValue()	617
15.22 CEventAdapter Class Reference	617
15.22.1 Detailed Description	618
15.22.2 Constructor & Destructor Documentation	618
15.22.2.1 CEventAdapter()	618
15.22.2.2 ~CEventAdapter()	618
15.22.3 Member Function Documentation	618
15.22.3.1 AttachNodeMap()	618
15.22.3.2 DeliverMessage()	619
15.22.3.3 DetachNodeMap()	619
15.22.4 Member Data Documentation	619
15.22.4.1 m_pEventAdapter	619
15.23 CEventAdapter1394 Class Reference	619
15.23.1 Detailed Description	620
15.23.2 Constructor & Destructor Documentation	620

15.23.2.1 CEventAdapter1394()	620
15.23.2.2 ~CEventAdapter1394()	620
15.23.3 Member Function Documentation	621
15.23.3.1 DeliverEventMessage()	621
15.23.3.2 DeliverMessage()	621
15.24 CEventAdapterGeneric Class Reference	621
15.24.1 Detailed Description	622
15.24.2 Constructor & Destructor Documentation	622
15.24.2.1 CEventAdapterGeneric()	622
15.24.2.2 ~CEventAdapterGeneric()	623
15.24.3 Member Function Documentation	623
15.24.3.1 DeliverMessage() [1/3]	623
15.24.3.2 DeliverMessage() [2/3]	623
15.24.3.3 DeliverMessage() [3/3]	623
15.25 CEventAdapterGEV Class Reference	624
15.25.1 Detailed Description	625
15.25.2 Constructor & Destructor Documentation	625
15.25.2.1 CEventAdapterGEV()	625
15.25.2.2 ~CEventAdapterGEV()	625
15.25.3 Member Function Documentation	625
15.25.3.1 DeliverEventMessage() [1/2]	625
15.25.3.2 DeliverEventMessage() [2/2]	625
15.25.3.3 DeliverMessage()	626
15.26 CEventAdapterU3V Class Reference	626
15.26.1 Detailed Description	627
15.26.2 Constructor & Destructor Documentation	627
15.26.2.1 CEventAdapterU3V()	627
15.26.2.2 ~CEventAdapterU3V()	627
15.26.3 Member Function Documentation	627
15.26.3.1 DeliverEventMessage()	627
15.26.3.2 DeliverMessage()	628
15.27 CEventPort Class Reference	628
15.27.1 Detailed Description	629
15.27.2 Constructor & Destructor Documentation	629
15.27.2.1 CEventPort()	630
15.27.2.2 ~CEventPort()	630
15.27.3 Member Function Documentation	630
15.27.3.1 AttachEvent()	630
15.27.3.2 AttachNode()	630
15.27.3.3 CheckEventID() [1/2]	630
15.27.3.4 CheckEventID() [2/2]	631
15.27.3.5 DetachEvent()	631

15.27.3.6 DetachNode()	631
15.27.3.7 GetAccessMode()	631
15.27.3.8 GetEventIDLength()	631
15.27.3.9 GetPrincipalInterfaceType()	631
15.27.3.10 GetSwapEndianness()	632
15.27.3.11 InvalidateNode()	632
15.27.3.12 Read()	632
15.27.3.13 SetPortImpl()	632
15.27.3.14 Write()	632
15.27.4 Member Data Documentation	632
15.27.4.1 m_pEventPort	633
15.27.4.2 m_pNode	633
15.27.4.3 m_pPortAdapter	633
15.28 CFeatureBag Class Reference	633
15.28.1 Detailed Description	634
15.28.2 Constructor & Destructor Documentation	634
15.28.2.1 CFeatureBag()	634
15.28.2.2 ~CFeatureBag()	634
15.28.3 Member Function Documentation	634
15.28.3.1 GetFeatureBagHandle()	634
15.28.3.2 LoadFromBag()	634
15.28.3.3 operator==()	635
15.28.3.4 PersistFeature()	635
15.28.3.5 SetInfo()	635
15.28.3.6 StoreToBag()	635
15.29 CFloatPtr Class Reference	636
15.29.1 Detailed Description	637
15.29.2 Constructor & Destructor Documentation	637
15.29.2.1 CFloatPtr() [1/2]	637
15.29.2.2 CFloatPtr() [2/2]	637
15.29.3 Member Function Documentation	637
15.29.3.1 GetEnumAlias()	637
15.29.3.2 GetIntAlias()	637
15.29.3.3 operator=()	638
15.30 CGeneric_XMLLoaderParams Class Reference	638
15.30.1 Detailed Description	638
15.30.2 Member Function Documentation	638
15.30.2.1 _Initialize()	639
15.31 CGlobalLock Class Reference	639
15.31.1 Detailed Description	639
15.31.2 Constructor & Destructor Documentation	639
15.31.2.1 CGlobalLock() [1/2]	640

15.31.2.2 CGlobalLock() [2/2]	640
15.31.2.3 ~CGlobalLock()	640
15.31.3 Member Function Documentation	640
15.31.3.1 IsValid()	640
15.31.3.2 Lock()	640
15.31.3.3 TryLock()	641
15.31.3.4 Unlock()	641
15.31.4 Member Data Documentation	641
15.31.4.1 m_DebugCount	641
15.32 CGlobalLockUnlocker Class Reference	641
15.32.1 Detailed Description	642
15.32.2 Constructor & Destructor Documentation	642
15.32.2.1 CGlobalLockUnlocker()	642
15.32.2.2 ~CGlobalLockUnlocker()	642
15.32.3 Member Function Documentation	642
15.32.3.1 UnlockEarly()	642
15.32.4 Member Data Documentation	643
15.32.4.1 m_enabled	643
15.32.4.2 m_Lock	643
15.33 ChunkData Class Reference	643
15.33.1 Detailed Description	645
15.33.2 Constructor & Destructor Documentation	646
15.33.2.1 ChunkData() [1/2]	646
15.33.2.2 ChunkData() [2/2]	646
15.33.2.3 ~ChunkData()	646
15.33.3 Member Function Documentation	646
15.33.3.1 GetBlackLevel()	646
15.33.3.2 GetCompressionMode()	646
15.33.3.3 GetCompressionRatio()	647
15.33.3.4 GetCounterValue()	647
15.33.3.5 GetCRC()	647
15.33.3.6 GetEncoderValue()	647
15.33.3.7 GetExposureEndLineStatusAll()	648
15.33.3.8 GetExposureTime()	648
15.33.3.9 GetFrameID()	648
15.33.3.10 GetGain()	648
15.33.3.11 GetHeight()	649
15.33.3.12 GetImage()	649
15.33.3.13 GetInferenceBoundingBoxResult()	649
15.33.3.14 GetInferenceConfidence()	649
15.33.3.15 GetInferenceFrameId()	650
15.33.3.16 GetInferenceResult()	650

15.33.3.17 GetLinePitch()	650
15.33.3.18 GetLineStatusAll()	650
15.33.3.19 GetOffsetX()	651
15.33.3.20 GetOffsetY()	651
15.33.3.21 GetPartSelector()	651
15.33.3.22 GetPixelDynamicRangeMax()	651
15.33.3.23 GetPixelDynamicRangeMin()	652
15.33.3.24 GetScan3dAxisMax()	652
15.33.3.25 GetScan3dAxisMin()	652
15.33.3.26 GetScan3dCoordinateOffset()	652
15.33.3.27 GetScan3dCoordinateReferenceValue()	653
15.33.3.28 GetScan3dCoordinateScale()	653
15.33.3.29 GetScan3dInvalidDataValue()	653
15.33.3.30 GetScan3dTransformValue()	653
15.33.3.31 GetScanLineSelector()	654
15.33.3.32 GetSequencerSetActive()	654
15.33.3.33 GetSerialDataLength()	654
15.33.3.34 GetStreamChannelID()	654
15.33.3.35 GetTimerValue()	655
15.33.3.36 GetTimestamp()	655
15.33.3.37 GetTimestampLatchValue()	655
15.33.3.38 GetTransferBlockID()	655
15.33.3.39 GetTransferQueueCurrentBlockCount()	656
15.33.3.40 GetWidth()	656
15.33.3.41 SetChunks()	656
15.33.4 Friends And Related Function Documentation	656
15.33.4.1 ChunkDataImpl	656
15.34 CLock Class Reference	657
15.34.1 Detailed Description	657
15.34.2 Constructor & Destructor Documentation	657
15.34.2.1 CLock()	657
15.34.2.2 ~CLock()	658
15.34.3 Member Function Documentation	658
15.34.3.1 Lock()	658
15.34.3.2 TryLock()	658
15.34.3.3 Unlock()	658
15.35 CLock Class Reference	658
15.35.1 Detailed Description	659
15.35.2 Constructor & Destructor Documentation	659
15.35.2.1 CLock() [1/2]	659
15.35.2.2 CLock() [2/2]	659
15.35.2.3 ~CLock()	660

15.35.3 Member Function Documentation	660
15.35.3.1 Lock()	660
15.35.3.2 TryLock()	660
15.35.3.3 Unlock()	660
15.35.4 Friends And Related Function Documentation	660
15.35.4.1 NodeMap	660
15.35.5 Member Data Documentation	660
15.35.5.1 m_bOwnLock	661
15.35.5.2 m_lock	661
15.36 CLockEx Class Reference	661
15.36.1 Detailed Description	662
15.37 CLockEx Class Reference	662
15.37.1 Detailed Description	663
15.37.2 Member Data Documentation	663
15.37.2.1 m_lockEx	663
15.38 CNodeCallback Class Reference	663
15.38.1 Detailed Description	664
15.38.2 Constructor & Destructor Documentation	664
15.38.2.1 CNodeCallback()	664
15.38.2.2 ~CNodeCallback()	664
15.38.3 Member Function Documentation	664
15.38.3.1 Destroy()	665
15.38.3.2 GetCallbackType()	665
15.38.3.3 GetNode()	665
15.38.3.4 operator()()	665
15.38.4 Member Data Documentation	665
15.38.4.1 m_CallbackType	665
15.38.4.2 m_pNode	666
15.39 CNodeMapFactory Class Reference	666
15.39.1 Detailed Description	667
15.39.2 Constructor & Destructor Documentation	668
15.39.2.1 CNodeMapFactory() [1/5]	668
15.39.2.2 ~CNodeMapFactory()	668
15.39.2.3 CNodeMapFactory() [2/5]	668
15.39.2.4 CNodeMapFactory() [3/5]	668
15.39.2.5 CNodeMapFactory() [4/5]	669
15.39.2.6 CNodeMapFactory() [5/5]	670
15.39.3 Member Function Documentation	670
15.39.3.1 AddInjectionData()	670
15.39.3.2 ApplyStyleSheet()	670
15.39.3.3 ClearCache()	671
15.39.3.4 CreateEmptyNodeMap()	671

15.39.3.5 CreateNodeDataFromNodeMap()	671
15.39.3.6 CreateNodeMap() [1/2]	671
15.39.3.7 CreateNodeMap() [2/2]	671
15.39.3.8 ExtractSubtree()	672
15.39.3.9 GetNodeStatistics()	672
15.39.3.10 GetSupportedSchemaVersions()	672
15.39.3.11 IsCameraDescriptionFileDataReleased()	672
15.39.3.12 IsEmpty()	672
15.39.3.13 IsLoaded()	673
15.39.3.14 IsPreprocessed()	673
15.39.3.15 LoadAndInject()	673
15.39.3.16 operator=()	673
15.39.3.17 Preprocess()	673
15.39.3.18 ReleaseCameraDescriptionFileData()	674
15.39.3.19 ToString()	674
15.39.3.20 ToXml()	674
15.40 CNodeMapRef Class Reference	674
15.40.1 Detailed Description	675
15.40.2 Constructor & Destructor Documentation	675
15.40.2.1 CNodeMapRef() [1/3]	676
15.40.2.2 CNodeMapRef() [2/3]	676
15.40.2.3 CNodeMapRef() [3/3]	676
15.40.3 Member Function Documentation	676
15.40.3.1 operator=() [1/2]	676
15.40.3.2 operator=() [2/2]	676
15.41 CNodeMapRefT< TCameraParams > Class Template Reference	677
15.41.1 Detailed Description	678
15.41.2 Constructor & Destructor Documentation	679
15.41.2.1 CNodeMapRefT() [1/3]	679
15.41.2.2 CNodeMapRefT() [2/3]	679
15.41.2.3 CNodeMapRefT() [3/3]	679
15.41.2.4 ~CNodeMapRefT()	679
15.41.3 Member Function Documentation	679
15.41.3.1 _ClearXMLCache()	680
15.41.3.2 _Connect() [1/2]	680
15.41.3.3 _Connect() [2/2]	680
15.41.3.4 _Destroy()	680
15.41.3.5 _GetDeviceName()	680
15.41.3.6 _GetNode()	680
15.41.3.7 _GetNodes()	681
15.41.3.8 _GetSupportedSchemaVersions()	681
15.41.3.9 _InvalidateNodes()	681

15.41.3.10	_LoadXMLFromFile()	681
15.41.3.11	_LoadXMLFromFileInject()	681
15.41.3.12	_LoadXMLFromString()	682
15.41.3.13	_LoadXMLFromStringInject()	682
15.41.3.14	_LoadXMLFromZIPData()	682
15.41.3.15	_LoadXMLFromZIPFile()	682
15.41.3.16	_Poll()	682
15.41.3.17	operator=() [1/2]	683
15.41.3.18	operator=() [2/2]	683
15.41.4	Member Data Documentation	683
15.41.4.1	_Ptr	683
15.42	CommandNode Class Reference	683
15.42.1	Detailed Description	684
15.42.2	Constructor & Destructor Documentation	684
15.42.2.1	CommandNode() [1/2]	685
15.42.2.2	CommandNode() [2/2]	685
15.42.2.3	~CommandNode()	685
15.42.3	Member Function Documentation	685
15.42.3.1	Execute()	685
15.42.3.2	IsDone()	685
15.42.3.3	operator()()	686
15.42.3.4	SetReference()	686
15.43	CompressedImageInfo Struct Reference	686
15.43.1	Constructor & Destructor Documentation	686
15.43.1.1	CompressedImageInfo()	687
15.43.2	Member Data Documentation	687
15.43.2.1	compressedImageSize	687
15.43.2.2	fileName	687
15.43.2.3	imageHeight	687
15.43.2.4	imagePixelFormat	687
15.43.2.5	imageWidth	687
15.43.2.6	imageXOffset	688
15.43.2.7	imageYOffset	688
15.44	Counter Class Reference	688
15.44.1	Detailed Description	688
15.44.2	Constructor & Destructor Documentation	688
15.44.2.1	Counter()	688
15.44.3	Member Function Documentation	688
15.44.3.1	GetValue()	689
15.44.3.2	IsZero()	689
15.44.3.3	operator unsigned int()	689
15.44.3.4	operator++() [1/2]	689

15.44.3.5 operator++() [2/2]	689
15.44.3.6 operator--() [1/2]	689
15.44.3.7 operator--() [2/2]	689
15.45 CPointer< T, B > Class Template Reference	690
15.45.1 Detailed Description	691
15.45.2 Constructor & Destructor Documentation	691
15.45.2.1 CPointer() [1/2]	691
15.45.2.2 CPointer() [2/2]	691
15.45.2.3 ~CPointer()	691
15.45.3 Member Function Documentation	691
15.45.3.1 IsValid()	691
15.45.3.2 operator bool()	692
15.45.3.3 operator T*()	692
15.45.3.4 operator"!="" [1/5]	692
15.45.3.5 operator"!="" [2/5]	692
15.45.3.6 operator"!="" [3/5]	692
15.45.3.7 operator"!="" [4/5]	692
15.45.3.8 operator"!="" [5/5]	693
15.45.3.9 operator()()	693
15.45.3.10 operator*()	693
15.45.3.11 operator->()	693
15.45.3.12 operator=()	693
15.45.3.13 operator==(1/3)	693
15.45.3.14 operator==(2/3)	694
15.45.3.15 operator==(3/3)	694
15.45.4 Member Data Documentation	694
15.45.4.1 m_pT	694
15.46 CPortImpl Class Reference	694
15.46.1 Detailed Description	695
15.46.2 Constructor & Destructor Documentation	695
15.46.2.1 CPortImpl()	696
15.46.2.2 ~CPortImpl()	696
15.46.3 Member Function Documentation	696
15.46.3.1 GetAccessMode()	696
15.46.3.2 GetSwapEndianness()	696
15.46.3.3 InvalidateNode()	696
15.46.3.4 Read()	697
15.46.3.5 Replay()	697
15.46.3.6 SetPortImpl()	697
15.46.3.7 Write()	697
15.46.4 Member Data Documentation	697
15.46.4.1 m_ptrPort	698

15.47 CPortWriteList Class Reference	698
15.47.1 Detailed Description	699
15.47.2 Constructor & Destructor Documentation	699
15.47.2.1 CPortWriteList()	699
15.47.2.2 ~CPortWriteList()	699
15.47.3 Member Function Documentation	699
15.47.3.1 GetCookie()	700
15.47.3.2 GetPortWriteListHandle()	700
15.47.3.3 Replay()	700
15.47.3.4 SetCookie()	700
15.47.3.5 Write()	700
15.47.4 Member Data Documentation	700
15.47.4.1 m_pWriteList	701
15.48 CpuUsageInfo Struct Reference	701
15.48.1 Member Data Documentation	701
15.48.1.1 dummy	701
15.49 CRegisterPortImpl Class Reference	701
15.49.1 Detailed Description	702
15.49.2 Constructor & Destructor Documentation	702
15.49.2.1 CRegisterPortImpl()	703
15.49.2.2 ~CRegisterPortImpl()	703
15.49.3 Member Function Documentation	703
15.49.3.1 GetAccessMode()	703
15.49.3.2 Read()	703
15.49.3.3 ReadRegister()	704
15.49.3.4 SetPortImpl()	704
15.49.3.5 Write()	704
15.49.3.6 WriteRegister()	704
15.50 CSelectorSet Class Reference	705
15.50.1 Detailed Description	706
15.50.2 Constructor & Destructor Documentation	706
15.50.2.1 CSelectorSet()	706
15.50.2.2 ~CSelectorSet()	706
15.50.3 Member Function Documentation	706
15.50.3.1 GetSelectorList()	706
15.50.3.2 IsEmpty()	707
15.50.3.3 Restore()	707
15.50.3.4 SetFirst()	707
15.50.3.5 SetNext()	707
15.50.3.6 ToString()	707
15.51 CTestPortStruct< CDataStruct > Class Template Reference	708
15.51.1 Detailed Description	709

15.51.2 Constructor & Destructor Documentation	709
15.51.2.1 CTestPortStruct()	709
15.51.3 Member Function Documentation	709
15.51.3.1 GetAccessMode()	709
15.51.3.2 GetNumReads()	710
15.51.3.3 GetNumWrites()	710
15.51.3.4 GetPrincipalInterfaceType()	710
15.51.3.5 MemSet()	710
15.51.3.6 Read()	710
15.51.3.7 ResetStatistics()	710
15.51.3.8 Write()	711
15.51.4 Member Data Documentation	711
15.51.4.1 m_BaseAddress	711
15.51.4.2 m_NumReads	711
15.51.4.3 m_NumWrites	711
15.52 DCAM_CHECKSUM Struct Reference	711
15.52.1 Member Data Documentation	712
15.52.1.1 CRCChecksum	712
15.53 DCAM_CHUNK_TRAILER Struct Reference	712
15.53.1 Member Data Documentation	712
15.53.1.1 ChunkID	712
15.53.1.2 ChunkLength	712
15.53.1.3 InverseChunkLength	712
15.54 DeviceArrivalEventHandler Class Reference	713
15.54.1 Detailed Description	714
15.54.2 Constructor & Destructor Documentation	714
15.54.2.1 DeviceArrivalEventHandler()	714
15.54.2.2 ~DeviceArrivalEventHandler()	714
15.54.3 Member Function Documentation	714
15.54.3.1 OnDeviceArrival()	714
15.54.3.2 operator=()	715
15.55 DeviceEventExposureEndData Struct Reference	715
15.55.1 Detailed Description	715
15.55.2 Member Data Documentation	715
15.55.2.1 frameID	715
15.56 DeviceEventHandler Class Reference	716
15.56.1 Detailed Description	717
15.56.2 Constructor & Destructor Documentation	717
15.56.2.1 DeviceEventHandler()	717
15.56.2.2 ~DeviceEventHandler()	717
15.56.3 Member Function Documentation	717
15.56.3.1 GetDeviceEventId()	718

15.56.3.2 GetDeviceEventName()	718
15.56.3.3 OnDeviceEvent()	718
15.56.3.4 operator=()	718
15.57 DeviceEventHandlerImpl Class Reference	719
15.57.1 Constructor & Destructor Documentation	720
15.57.1.1 DeviceEventHandlerImpl()	720
15.57.1.2 ~DeviceEventHandlerImpl()	720
15.57.2 Member Function Documentation	721
15.57.2.1 OnDeviceEvent()	721
15.58 DeviceEventInferenceData Struct Reference	721
15.58.1 Detailed Description	721
15.58.2 Member Data Documentation	721
15.58.2.1 confidence	721
15.58.2.2 frameID	722
15.58.2.3 result	722
15.59 DeviceEventUtility Class Reference	722
15.59.1 Member Function Documentation	722
15.59.1.1 ParseDeviceEventExposureEnd()	722
15.59.1.2 ParseDeviceEventInference()	723
15.60 DeviceRemovalEventHandler Class Reference	723
15.60.1 Detailed Description	724
15.60.2 Constructor & Destructor Documentation	724
15.60.2.1 DeviceRemovalEventHandler()	725
15.60.2.2 ~DeviceRemovalEventHandler()	725
15.60.3 Member Function Documentation	725
15.60.3.1 OnDeviceRemoval()	725
15.60.3.2 operator=()	725
15.61 double_autovector_t Class Reference	725
15.61.1 Detailed Description	726
15.61.2 Constructor & Destructor Documentation	726
15.61.2.1 double_autovector_t() [1/3]	726
15.61.2.2 double_autovector_t() [2/3]	726
15.61.2.3 double_autovector_t() [3/3]	726
15.61.2.4 ~double_autovector_t()	727
15.61.3 Member Function Documentation	727
15.61.3.1 operator delete()	727
15.61.3.2 operator new()	727
15.61.3.3 operator=()	727
15.61.3.4 operator[]() [1/2]	727
15.61.3.5 operator[]() [2/2]	727
15.61.3.6 size()	728
15.61.4 Member Data Documentation	728

15.61.4.1 _pCount	728
15.61.4.2 _pv	728
15.62 EAccessModeClass Class Reference	728
15.62.1 Detailed Description	728
15.62.2 Member Function Documentation	728
15.62.2.1 FromString()	729
15.62.2.2 ToString() [1/2]	729
15.62.2.3 ToString() [2/2]	729
15.63 ECachingModeClass Class Reference	729
15.63.1 Detailed Description	729
15.63.2 Member Function Documentation	730
15.63.2.1 FromString()	730
15.63.2.2 ToString() [1/2]	730
15.63.2.3 ToString() [2/2]	730
15.64 EDisplayNotationClass Class Reference	730
15.64.1 Detailed Description	731
15.64.2 Member Function Documentation	731
15.64.2.1 FromString()	731
15.64.2.2 ToString() [1/2]	731
15.64.2.3 ToString() [2/2]	731
15.65 EEndianessClass Class Reference	731
15.65.1 Detailed Description	732
15.65.2 Member Function Documentation	732
15.65.2.1 FromString()	732
15.65.2.2 ToString() [1/2]	732
15.65.2.3 ToString() [2/2]	732
15.66 EGenApiSchemaVersionClass Class Reference	733
15.66.1 Detailed Description	733
15.66.2 Member Function Documentation	733
15.66.2.1 FromString()	733
15.66.2.2 ToString() [1/2]	733
15.66.2.3 ToString() [2/2]	733
15.67 EInputDirectionClass Class Reference	734
15.67.1 Detailed Description	734
15.67.2 Member Function Documentation	734
15.67.2.1 FromString()	734
15.67.2.2 ToString() [1/2]	734
15.67.2.3 ToString() [2/2]	734
15.68 ENamespaceClass Class Reference	735
15.68.1 Detailed Description	735
15.68.2 Member Function Documentation	735
15.68.2.1 FromString()	735

15.68.2.2 ToString() [1/2]	735
15.68.2.3 ToString() [2/2]	735
15.69 EnumEntryNode Class Reference	736
15.69.1 Detailed Description	737
15.69.2 Constructor & Destructor Documentation	737
15.69.2.1 EnumEntryNode() [1/2]	737
15.69.2.2 EnumEntryNode() [2/2]	737
15.69.2.3 ~EnumEntryNode()	737
15.69.3 Member Function Documentation	737
15.69.3.1 GetNumericValue()	738
15.69.3.2 GetSymbolic()	738
15.69.3.3 GetValue()	738
15.69.3.4 IsSelfClearing()	738
15.69.3.5 SetReference()	738
15.70 EnumNode Class Reference	739
15.70.1 Detailed Description	741
15.70.2 Constructor & Destructor Documentation	741
15.70.2.1 EnumNode() [1/2]	741
15.70.2.2 EnumNode() [2/2]	741
15.70.2.3 ~EnumNode()	741
15.70.3 Member Function Documentation	741
15.70.3.1 GetCurrentEntry()	741
15.70.3.2 GetEntries()	742
15.70.3.3 GetEntry()	742
15.70.3.4 GetEntryByName()	742
15.70.3.5 GetIntValue()	742
15.70.3.6 GetSymbolics()	743
15.70.3.7 operator*()	743
15.70.3.8 operator=()	743
15.70.3.9 SetIntValue()	743
15.70.3.10 SetReference()	743
15.70.4 Member Data Documentation	744
15.70.4.1 m_pEnumeration	744
15.71 ERepresentationClass Class Reference	744
15.71.1 Detailed Description	744
15.71.2 Member Function Documentation	744
15.71.2.1 FromString()	745
15.71.2.2 ToString() [1/2]	745
15.71.2.3 ToString() [2/2]	745
15.72 ESignClass Class Reference	745
15.72.1 Detailed Description	745
15.72.2 Member Function Documentation	746

15.72.2.1 FromString()	746
15.72.2.2 ToString() [1/2]	746
15.72.2.3 ToString() [2/2]	746
15.73 ESlopeClass Class Reference	746
15.73.1 Detailed Description	747
15.73.2 Member Function Documentation	747
15.73.2.1 FromString()	747
15.73.2.2 ToString() [1/2]	747
15.73.2.3 ToString() [2/2]	747
15.74 EStandardNameSpaceClass Class Reference	747
15.74.1 Detailed Description	748
15.74.2 Member Function Documentation	748
15.74.2.1 FromString()	748
15.74.2.2 ToString() [1/2]	748
15.74.2.3 ToString() [2/2]	748
15.75 EventHandler Class Reference	749
15.75.1 Detailed Description	750
15.75.2 Constructor & Destructor Documentation	750
15.75.2.1 ~EventHandler()	750
15.75.2.2 EventHandler()	750
15.75.3 Member Function Documentation	750
15.75.3.1 GetEventPayloadData()	750
15.75.3.2 GetEventPayloadDataSize()	750
15.75.3.3 GetEventType()	751
15.75.3.4 operator=()	751
15.75.3.5 SetEventPayload()	751
15.75.3.6 SetEventType()	751
15.75.4 Friends And Related Function Documentation	751
15.75.4.1 CallbackHandlerInternal	751
15.75.4.2 EventProcessor	752
15.75.4.3 IDataStream	752
15.75.4.4 ImageSetInspector	752
15.75.4.5 Stream	752
15.75.5 Member Data Documentation	752
15.75.5.1 m_pEventData	752
15.76 EVisibilityClass Class Reference	752
15.76.1 Detailed Description	753
15.76.2 Member Function Documentation	753
15.76.2.1 FromString()	753
15.76.2.2 ToString() [1/2]	753
15.76.2.3 ToString() [2/2]	753
15.77 Exception Class Reference	754

15.77.1 Detailed Description	755
15.77.2 Constructor & Destructor Documentation	755
15.77.2.1 Exception() [1/4]	755
15.77.2.2 Exception() [2/4]	755
15.77.2.3 Exception() [3/4]	756
15.77.2.4 Exception() [4/4]	756
15.77.2.5 ~Exception()	756
15.77.3 Member Function Documentation	757
15.77.3.1 GetBuildDate()	757
15.77.3.2 GetBuildTime()	757
15.77.3.3 GetError()	757
15.77.3.4 GetErrorMessage()	757
15.77.3.5 GetFileName()	757
15.77.3.6 GetFullErrorMessage()	757
15.77.3.7 GetFunctionName()	758
15.77.3.8 GetLineNumber()	758
15.77.3.9 operator!=()	758
15.77.3.10 operator=()	758
15.77.3.11 operator==(())	758
15.77.3.12 what()	758
15.78 EYesNoClass Class Reference	759
15.78.1 Detailed Description	759
15.78.2 Member Function Documentation	759
15.78.2.1 FromString()	759
15.78.2.2 ToString() [1/2]	759
15.78.2.3 ToString() [2/2]	759
15.79 FileProtocolAdapter Class Reference	760
15.79.1 Detailed Description	760
15.79.2 Constructor & Destructor Documentation	760
15.79.2.1 FileProtocolAdapter()	760
15.79.2.2 ~FileProtocolAdapter()	760
15.79.3 Member Function Documentation	761
15.79.3.1 attach()	761
15.79.3.2 closeFile()	762
15.79.3.3 deleteFile()	762
15.79.3.4 getBufSize()	762
15.79.3.5 openFile()	763
15.79.3.6 read()	763
15.79.3.7 write()	764
15.80 FloatNode Class Reference	764
15.80.1 Detailed Description	766
15.80.2 Constructor & Destructor Documentation	767

15.80.2.1 FloatNode() [1/2]	767
15.80.2.2 FloatNode() [2/2]	767
15.80.2.3 ~FloatNode()	767
15.80.3 Member Function Documentation	767
15.80.3.1 GetDisplayNotation()	767
15.80.3.2 GetDisplayPrecision()	767
15.80.3.3 GetEnumAlias()	767
15.80.3.4 GetInc()	768
15.80.3.5 GetIncMode()	768
15.80.3.6 GetIntAlias()	768
15.80.3.7 GetListOfValidValues()	768
15.80.3.8 GetMax()	768
15.80.3.9 GetMin()	768
15.80.3.10 GetRepresentation()	769
15.80.3.11 GetUnit()	769
15.80.3.12 GetValue()	769
15.80.3.13 HasInc()	769
15.80.3.14 ImposeMax()	769
15.80.3.15 ImposeMin()	770
15.80.3.16 operator>()	770
15.80.3.17 operator*()	770
15.80.3.18 operator=()	770
15.80.3.19 SetReference()	770
15.80.3.20 SetValue()	770
15.81 FloatRegNode Class Reference	771
15.81.1 Detailed Description	772
15.81.2 Constructor & Destructor Documentation	772
15.81.2.1 FloatRegNode() [1/2]	773
15.81.2.2 FloatRegNode() [2/2]	773
15.81.2.3 ~FloatRegNode()	773
15.81.3 Member Function Documentation	773
15.81.3.1 SetReference()	773
15.82 Function_NodeCallback< Function > Class Template Reference	774
15.82.1 Detailed Description	774
15.82.2 Constructor & Destructor Documentation	775
15.82.2.1 Function_NodeCallback()	775
15.82.3 Member Function Documentation	775
15.82.3.1 Destroy()	775
15.82.3.2 operator>()	775
15.83 gcstring Class Reference	776
15.83.1 Constructor & Destructor Documentation	777
15.83.1.1 gcstring() [1/5]	777

15.83.1.2 gcstring() [2/5]	777
15.83.1.3 gcstring() [3/5]	777
15.83.1.4 gcstring() [4/5]	777
15.83.1.5 gcstring() [5/5]	778
15.83.1.6 ~gcstring()	778
15.83.2 Member Function Documentation	778
15.83.2.1 _npos()	778
15.83.2.2 append() [1/2]	778
15.83.2.3 append() [2/2]	778
15.83.2.4 assign() [1/4]	778
15.83.2.5 assign() [2/4]	779
15.83.2.6 assign() [3/4]	779
15.83.2.7 assign() [4/4]	779
15.83.2.8 c_str()	779
15.83.2.9 compare()	779
15.83.2.10 empty()	779
15.83.2.11 find() [1/5]	780
15.83.2.12 find() [2/5]	780
15.83.2.13 find() [3/5]	780
15.83.2.14 find() [4/5]	780
15.83.2.15 find() [5/5]	780
15.83.2.16 find_first_not_of()	780
15.83.2.17 find_first_of()	781
15.83.2.18 length()	781
15.83.2.19 max_size()	781
15.83.2.20 operator const char *()	781
15.83.2.21 operator delete() [1/2]	781
15.83.2.22 operator delete() [2/2]	781
15.83.2.23 operator new() [1/2]	781
15.83.2.24 operator new() [2/2]	782
15.83.2.25 operator"!=(	782
15.83.2.26 operator"!=(	782
15.83.2.27 operator+=(	782
15.83.2.28 operator+=(	782
15.83.2.29 operator+=(	782
15.83.2.30 operator+=(	782
15.83.2.31 operator+=(	783
15.83.2.32 operator<()	783
15.83.2.33 operator=()	783
15.83.2.34 operator==(	783
15.83.2.35 operator==(	783
15.83.2.36 operator>()	783

15.83.2.37	resize()	783
15.83.2.38	size()	784
15.83.2.39	substr()	784
15.83.2.40	swap()	784
15.83.3	Friends And Related Function Documentation	784
15.83.3.1	operator+ [1/3]	784
15.83.3.2	operator+ [2/3]	784
15.83.3.3	operator+ [3/3]	784
15.83.4	Member Data Documentation	785
15.83.4.1	npos	785
15.84	GrabInfo Struct Reference	785
15.84.1	Constructor & Destructor Documentation	785
15.84.1.1	GrabInfo()	785
15.84.2	Member Data Documentation	785
15.84.2.1	imageEventHandler	785
15.84.2.2	numImagesGrabbed	786
15.84.2.3	numIncompleteImages	786
15.84.2.4	numRemovals	786
15.85	GVCP_CHUNK_TRAILER Struct Reference	786
15.85.1	Detailed Description	786
15.85.2	Member Data Documentation	786
15.85.2.1	ChunkID	786
15.85.2.2	ChunkLength	787
15.86	GVCP_EVENT_ITEM Struct Reference	787
15.86.1	Detailed Description	787
15.86.2	Member Data Documentation	787
15.86.2.1	BlockId	787
15.86.2.2	EventId	787
15.86.2.3	ReservedOrEventSize	788
15.86.2.4	StreamChannelId	788
15.86.2.5	TimestampHigh	788
15.86.2.6	TimestampLow	788
15.87	GVCP_EVENT_ITEM_BASIC Struct Reference	788
15.87.1	Detailed Description	788
15.87.2	Member Data Documentation	788
15.87.2.1	EventId	789
15.87.2.2	ReservedOrEventSize	789
15.88	GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference	789
15.88.1	Detailed Description	789
15.88.2	Member Data Documentation	789
15.88.2.1	BlockId	789
15.88.2.2	BlockId64High	790

15.88.2.3 BlockId64Low	790
15.88.2.4 EventId	790
15.88.2.5 ReservedOrEventSize	790
15.88.2.6 StreamChannelId	790
15.88.2.7 TimestampHigh	790
15.88.2.8 TimestampLow	790
15.89 GVCP_EVENT_REQUEST Struct Reference	791
15.89.1 Detailed Description	791
15.89.2 Member Data Documentation	791
15.89.2.1 Header	791
15.89.2.2 Items	791
15.90 GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference	792
15.90.1 Detailed Description	792
15.90.2 Member Data Documentation	792
15.90.2.1 Header	792
15.90.2.2 Items	792
15.91 GVCP_EVENTDATA_REQUEST Struct Reference	793
15.91.1 Detailed Description	793
15.91.2 Member Data Documentation	793
15.91.2.1 Data	793
15.91.2.2 Event	793
15.91.2.3 Header	794
15.92 GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference	794
15.92.1 Detailed Description	794
15.92.2 Member Data Documentation	794
15.92.2.1 Data	794
15.92.2.2 Event	795
15.92.2.3 Header	795
15.93 GVCP_REQUEST_HEADER Struct Reference	795
15.93.1 Detailed Description	795
15.93.2 Member Data Documentation	795
15.93.2.1 Command	795
15.93.2.2 Flags	795
15.93.2.3 Length	796
15.93.2.4 Magic	796
15.93.2.5 ReqId	796
15.94 H264Option Struct Reference	796
15.94.1 Detailed Description	796
15.94.2 Constructor & Destructor Documentation	797
15.94.2.1 H264Option()	797
15.94.3 Member Data Documentation	797
15.94.3.1 bitrate	797

15.94.3.2 frameRate	797
15.94.3.3 height	797
15.94.3.4 reserved	797
15.94.3.5 width	798
15.95 ICameraBase Class Reference	798
15.95.1 Detailed Description	800
15.95.2 Constructor & Destructor Documentation	800
15.95.2.1 ~ICameraBase()	800
15.95.2.2 ICameraBase() [1/2]	800
15.95.2.3 ICameraBase() [2/2]	800
15.95.3 Member Function Documentation	800
15.95.3.1 BeginAcquisition()	800
15.95.3.2 DelInit()	801
15.95.3.3 DiscoverMaxPacketSize()	801
15.95.3.4 EndAcquisition()	801
15.95.3.5 ForceIP()	801
15.95.3.6 GetAccessMode()	801
15.95.3.7 GetBufferOwnership()	801
15.95.3.8 GetGuiXml()	802
15.95.3.9 GetNextImage()	802
15.95.3.10 GetNodeMap()	802
15.95.3.11 GetNumDataStreams()	802
15.95.3.12 GetNumImagesInUse()	802
15.95.3.13 GetTLDeviceNodeMap()	802
15.95.3.14 GetTLStreamNodeMap()	803
15.95.3.15 GetUniqueID()	803
15.95.3.16 GetUserBufferCount()	803
15.95.3.17 GetUserBufferSize()	803
15.95.3.18 GetUserBufferTotalSize()	803
15.95.3.19 Init()	803
15.95.3.20 IsInitialized()	804
15.95.3.21 IsStreaming()	804
15.95.3.22 IsValid()	804
15.95.3.23 operator=()	804
15.95.3.24 ReadPort()	804
15.95.3.25 RegisterEventHandler() [1/3]	804
15.95.3.26 RegisterEventHandler() [2/3]	805
15.95.3.27 RegisterEventHandler() [3/3]	805
15.95.3.28 SetBufferOwnership()	805
15.95.3.29 SetUserBuffers() [1/2]	805
15.95.3.30 SetUserBuffers() [2/2]	805
15.95.3.31 UnregisterEventHandler()	806

15.95.3.32 WritePort()	806
15.95.4 Friends And Related Function Documentation	806
15.95.4.1 CameraInternal	806
15.95.4.2 InterfaceImpl	806
15.95.5 Member Data Documentation	806
15.95.5.1 m_pCameraBaseData	806
15.95.5.2 TLDevice	807
15.95.5.3 TLStream	807
15.96 ICameraList Class Reference	807
15.96.1 Detailed Description	808
15.96.2 Constructor & Destructor Documentation	808
15.96.2.1 ~ICameraList()	808
15.96.2.2 ICameraList() [1/2]	808
15.96.2.3 ICameraList() [2/2]	808
15.96.3 Member Function Documentation	808
15.96.3.1 Add()	809
15.96.3.2 Append()	809
15.96.3.3 Clear()	809
15.96.3.4 GetByDeviceID()	809
15.96.3.5 GetByIndex()	809
15.96.3.6 GetBySerial()	809
15.96.3.7 GetSize()	810
15.96.3.8 operator=()	810
15.96.3.9 operator[]()	810
15.96.3.10 Remove()	810
15.96.3.11 RemoveByDeviceID()	810
15.96.3.12 RemoveByIndex()	810
15.96.3.13 RemoveBySerial()	811
15.96.4 Friends And Related Function Documentation	811
15.96.4.1 CameraListImpl	811
15.96.4.2 InterfaceImpl	811
15.96.5 Member Data Documentation	811
15.96.5.1 m_pCameraListData	811
15.97 IChunkData Class Reference	811
15.97.1 Detailed Description	813
15.97.2 Constructor & Destructor Documentation	813
15.97.2.1 ~IChunkData()	813
15.97.2.2 IChunkData()	813
15.97.3 Member Function Documentation	813
15.97.3.1 GetBlackLevel()	813
15.97.3.2 GetCompressionMode()	813
15.97.3.3 GetCompressionRatio()	814

15.97.3.4 GetCounterValue()	814
15.97.3.5 GetCRC()	814
15.97.3.6 GetEncoderValue()	814
15.97.3.7 GetExposureEndLineStatusAll()	814
15.97.3.8 GetExposureTime()	814
15.97.3.9 GetFrameID()	815
15.97.3.10 GetGain()	815
15.97.3.11 GetHeight()	815
15.97.3.12 GetImage()	815
15.97.3.13 GetInferenceBoundingBoxResult()	815
15.97.3.14 GetInferenceConfidence()	815
15.97.3.15 GetInferenceFrameId()	816
15.97.3.16 GetInferenceResult()	816
15.97.3.17 GetLinePitch()	816
15.97.3.18 GetLineStatusAll()	816
15.97.3.19 GetOffsetX()	816
15.97.3.20 GetOffsetY()	816
15.97.3.21 GetPartSelector()	817
15.97.3.22 GetPixelDynamicRangeMax()	817
15.97.3.23 GetPixelDynamicRangeMin()	817
15.97.3.24 GetScan3dAxisMax()	817
15.97.3.25 GetScan3dAxisMin()	817
15.97.3.26 GetScan3dCoordinateOffset()	817
15.97.3.27 GetScan3dCoordinateReferenceValue()	818
15.97.3.28 GetScan3dCoordinateScale()	818
15.97.3.29 GetScan3dInvalidDataValue()	818
15.97.3.30 GetScan3dTransformValue()	818
15.97.3.31 GetScanLineSelector()	818
15.97.3.32 GetSequencerSetActive()	818
15.97.3.33 GetSerialDataLength()	819
15.97.3.34 GetStreamChannelID()	819
15.97.3.35 GetTimerValue()	819
15.97.3.36 GetTimestamp()	819
15.97.3.37 GetTimestampLatchValue()	819
15.97.3.38 GetTransferBlockID()	819
15.97.3.39 GetTransferQueueCurrentBlockCount()	820
15.97.3.40 GetWidth()	820
15.97.3.41 SetChunks()	820
15.97.4 Friends And Related Function Documentation	820
15.97.4.1 ChunkDataImpl	820
15.97.4.2 ImageImpl	820
15.97.5 Member Data Documentation	820

15.97.5.1 m_pChunkDataData	820
15.98 IDataStream Class Reference	821
15.98.1 Constructor & Destructor Documentation	822
15.98.1.1 ~IDataStream()	822
15.98.1.2 IDataStream()	822
15.98.2 Member Function Documentation	822
15.98.2.1 AnnouncelImage() [1/3]	822
15.98.2.2 AnnouncelImage() [2/3]	822
15.98.2.3 AnnouncelImage() [3/3]	822
15.98.2.4 AttachBuffer()	823
15.98.2.5 CleanupChunkAdapter()	823
15.98.2.6 FlushQueueAllDiscard()	823
15.98.2.7 GetBufferChunkData()	823
15.98.2.8 GetBufferInfoBool8Type()	823
15.98.2.9 GetBufferInfoPtrType()	823
15.98.2.10 GetBufferInfoSizeType()	824
15.98.2.11 GetBufferInfoUInt64Type()	824
15.98.2.12 GetDeviceNodeMap()	824
15.98.2.13 GetNextImage()	824
15.98.2.14 GetNextImageInternal()	824
15.98.2.15 GetNodeMap()	824
15.98.2.16 GetNumImagesInUse()	824
15.98.2.17 GetPort()	825
15.98.2.18 GetStreamID()	825
15.98.2.19 GetStreamIndex()	825
15.98.2.20 GetStreamInfoBool8Type()	825
15.98.2.21 GetStreamInfoSizeType()	825
15.98.2.22 GetStreamType()	825
15.98.2.23 GetSystemBuffer()	825
15.98.2.24 InitChunkAdapter()	826
15.98.2.25 IsAutoPollingEventHandlerRegistered()	826
15.98.2.26 IsCRCCheckEnabled()	826
15.98.2.27 IsEventHandlerRegistered()	826
15.98.2.28 IsImageBufferInUse()	826
15.98.2.29 IsStreaming()	826
15.98.2.30 KillBufferEvent()	826
15.98.2.31 RegisterImageEventHandler()	827
15.98.2.32 ReleaselImageBuffer()	827
15.98.2.33 ResetStreamHandle()	827
15.98.2.34 RevokelImages()	827
15.98.2.35 StartStream()	827
15.98.2.36 StopStream()	827

15.98.2.37 TransportLayerStreamInfo()	827
15.98.2.38 UnregisterImageEventHandler()	828
15.98.2.39 WaitOnImageEvent()	828
15.99 IDevFileStreamBase< CharType, Traits > Class Template Reference	828
15.99.1 Member Typedef Documentation	829
15.99.1.1 filebuf_type	829
15.99.1.2 ios_type	829
15.99.1.3 istream_type	829
15.99.2 Member Function Documentation	829
15.99.2.1 close()	829
15.99.2.2 is_open()	830
15.99.2.3 open()	830
15.99.2.4 rdbuf()	830
15.100 IDevFileStreamBuf< CharType, Traits > Class Template Reference	830
15.100.1 Constructor & Destructor Documentation	831
15.100.1.1 IDevFileStreamBuf()	831
15.100.1.2 ~IDevFileStreamBuf()	831
15.100.2 Member Function Documentation	832
15.100.2.1 close()	832
15.100.2.2 is_open()	832
15.100.2.3 open()	832
15.100.2.4 pbackfail()	832
15.100.2.5 underflow()	832
15.101 IDeviceArrivalEventHandler Class Reference	833
15.101.1 Constructor & Destructor Documentation	834
15.101.1.1 ~IDeviceArrivalEventHandler()	834
15.101.1.2 IDeviceArrivalEventHandler() [1/2]	834
15.101.1.3 IDeviceArrivalEventHandler() [2/2]	834
15.101.2 Member Function Documentation	834
15.101.2.1 OnDeviceArrival()	834
15.101.2.2 operator=()	835
15.102 IDeviceEventHandler Class Reference	835
15.102.1 Constructor & Destructor Documentation	836
15.102.1.1 ~IDeviceEventHandler()	836
15.102.1.2 IDeviceEventHandler() [1/2]	836
15.102.1.3 IDeviceEventHandler() [2/2]	836
15.102.2 Member Function Documentation	836
15.102.2.1 GetDeviceEventId()	836
15.102.2.2 GetDeviceEventName()	837
15.102.2.3 OnDeviceEvent()	837
15.102.2.4 operator=()	837
15.103 IDeviceRemovalEventHandler Class Reference	837

15.103.1 Constructor & Destructor Documentation	838
15.103.1.1 ~IDeviceRemovalEventHandler()	838
15.103.1.2 IDeviceRemovalEventHandler() [1/2]	838
15.103.1.3 IDeviceRemovalEventHandler() [2/2]	839
15.103.2 Member Function Documentation	839
15.103.2.1 OnDeviceRemoval()	839
15.103.2.2 operator=()	839
15.104 IImage Class Reference	839
15.104.1 Detailed Description	841
15.104.2 Constructor & Destructor Documentation	841
15.104.2.1 ~IImage()	841
15.104.2.2 IImage()	841
15.104.3 Member Function Documentation	841
15.104.3.1 CalculateStatistics()	841
15.104.3.2 CheckCRC()	842
15.104.3.3 DeepCopy()	842
15.104.3.4 GetBitsPerPixel()	842
15.104.3.5 GetBufferSize()	842
15.104.3.6 GetChunkData()	842
15.104.3.7 GetChunkLayoutId()	842
15.104.3.8 GetColorProcessing()	843
15.104.3.9 GetData()	843
15.104.3.10 GetDataAbsoluteMax()	843
15.104.3.11 GetDataAbsoluteMin()	843
15.104.3.12 GetFrameID()	843
15.104.3.13 GetHeight()	843
15.104.3.14 GetID()	844
15.104.3.15 GetImageData()	844
15.104.3.16 GetImageSize()	844
15.104.3.17 GetImageStatus()	844
15.104.3.18 GetNumChannels()	844
15.104.3.19 GetPayloadType()	844
15.104.3.20 GetPixelFormat()	845
15.104.3.21 GetPixelFormatIntType()	845
15.104.3.22 GetPixelFormatName()	845
15.104.3.23 GetPrivateData()	845
15.104.3.24 GetStreamIndex()	845
15.104.3.25 GetStride()	845
15.104.3.26 GetTimeStamp()	846
15.104.3.27 GetTLPayloadType()	846
15.104.3.28 GetTLPixelFormat()	846
15.104.3.29 GetTLPixelFormatNamespace()	846

15.104.3.30	IsValidPayloadSize()	846
15.104.3.31	GetWidth()	846
15.104.3.32	GetXOffset()	847
15.104.3.33	GetXPadding()	847
15.104.3.34	GetYOffset()	847
15.104.3.35	GetYPadding()	847
15.104.3.36	HasChunkData()	847
15.104.3.37	HasCRC()	847
15.104.3.38	IsCompressed()	848
15.104.3.39	IsIncomplete()	848
15.104.3.40	IsInUse()	848
15.104.3.41	Release()	848
15.104.3.42	ResetImage() [1/3]	848
15.104.3.43	ResetImage() [2/3]	849
15.104.3.44	ResetImage() [3/3]	849
15.104.3.45	Save() [1/9]	849
15.104.3.46	Save() [2/9]	849
15.104.3.47	Save() [3/9]	850
15.104.3.48	Save() [4/9]	850
15.104.3.49	Save() [5/9]	850
15.104.3.50	Save() [6/9]	850
15.104.3.51	Save() [7/9]	850
15.104.3.52	Save() [8/9]	851
15.104.3.53	Save() [9/9]	851
15.104.4	Friends And Related Function Documentation	851
15.104.4.1	ImageImpl	851
15.104.4.2	ImageListImpl	851
15.104.4.3	Stream	851
15.105	ImageEventHandler Class Reference	852
15.105.1	Constructor & Destructor Documentation	853
15.105.1.1	~ImageEventHandler()	853
15.105.1.2	ImageEventHandler() [1/2]	853
15.105.1.3	ImageEventHandler() [2/2]	853
15.105.2	Member Function Documentation	853
15.105.2.1	OnImageEvent()	853
15.105.2.2	operator=()	853
15.105.3	Friends And Related Function Documentation	854
15.105.3.1	Stream	854
15.106	ImageList Class Reference	854
15.106.1	Detailed Description	855
15.106.2	Constructor & Destructor Documentation	855
15.106.2.1	~ImageList()	855

15.106.2.2 IImageList() [1/2]	855
15.106.2.3 IImageList() [2/2]	855
15.106.3 Member Function Documentation	855
15.106.3.1 Add()	856
15.106.3.2 Append()	856
15.106.3.3 Clear()	856
15.106.3.4 GetByIndex()	856
15.106.3.5 GetByPixelFormat()	856
15.106.3.6 GetSize()	856
15.106.3.7 operator=()	857
15.106.3.8 operator[]()	857
15.106.3.9 Release()	857
15.106.3.10 RemoveByIndex()	857
15.106.3.11 RemoveByPixelFormat()	857
15.106.3.12 Save()	857
15.106.4 Friends And Related Function Documentation	858
15.106.4.1 ImageListImpl	858
15.106.5 Member Data Documentation	858
15.106.5.1 m_pImageListData	858
15.107 IImageListEventHandler Class Reference	858
15.107.1 Constructor & Destructor Documentation	859
15.107.1.1 ~IImageListEventHandler()	859
15.107.1.2 IImageListEventHandler() [1/2]	859
15.107.1.3 IImageListEventHandler() [2/2]	860
15.107.2 Member Function Documentation	860
15.107.2.1 OnImageListEvent()	860
15.107.2.2 operator=()	860
15.107.3 Friends And Related Function Documentation	860
15.107.3.1 EventCallbackInfo	860
15.108 ImageProcessor Class Reference	860
15.108.1 Detailed Description	861
15.108.2 Constructor & Destructor Documentation	861
15.108.2.1 ~ImageProcessor()	861
15.108.2.2 ImageProcessor() [1/2]	861
15.108.2.3 ImageProcessor() [2/2]	862
15.108.3 Member Function Documentation	862
15.108.3.1 ApplyGamma() [1/2]	862
15.108.3.2 ApplyGamma() [2/2]	862
15.108.3.3 Convert() [1/4]	862
15.108.3.4 Convert() [2/4]	862
15.108.3.5 Convert() [3/4]	863
15.108.3.6 Convert() [4/4]	863

15.108.3.7 GetColorProcessing()	863
15.108.3.8 GetNumDecompressionThreads()	863
15.108.3.9 operator=()	863
15.108.3.10 SetColorProcessing()	863
15.108.3.11 SetNumDecompressionThreads()	864
15.108.4 Member Data Documentation	864
15.108.4.1 m_pImageProcessorData	864
15.109 IImageStatistics Class Reference	864
15.109.1 Detailed Description	865
15.109.2 Constructor & Destructor Documentation	865
15.109.2.1 ~IImageStatistics()	865
15.109.2.2 IImageStatistics() [1/2]	865
15.109.2.3 IImageStatistics() [2/2]	866
15.109.3 Member Function Documentation	866
15.109.3.1 DisableAll()	866
15.109.3.2 EnableAll()	866
15.109.3.3 EnableGreyOnly()	866
15.109.3.4 EnableHSLOnly()	866
15.109.3.5 EnableRGBOnly()	866
15.109.3.6 GetChannelStatus()	867
15.109.3.7 GetHistogram()	867
15.109.3.8 GetMean()	867
15.109.3.9 GetNumPixelValues()	867
15.109.3.10 GetPixelValueRange()	867
15.109.3.11 GetRange()	868
15.109.3.12 GetStatistics()	868
15.109.3.13 SetChannelStatus()	868
15.110 IInterface Class Reference	868
15.110.1 Detailed Description	870
15.110.2 Constructor & Destructor Documentation	870
15.110.2.1 ~IInterface()	870
15.110.2.2 IInterface() [1/2]	870
15.110.2.3 IInterface() [2/2]	870
15.110.3 Member Function Documentation	870
15.110.3.1 GetCameras()	871
15.110.3.2 GetTLNodeMap()	871
15.110.3.3 IsInUse()	871
15.110.3.4 IsValid()	871
15.110.3.5 operator=()	871
15.110.3.6 RegisterEventHandler()	871
15.110.3.7 SendActionCommand()	872
15.110.3.8 UnregisterEventHandler()	872

15.110.3.9 UpdateCameras()	872
15.110.4 Friends And Related Function Documentation	872
15.110.4.1 InterfaceInternal	872
15.110.4.2 ProducerImpl	872
15.110.5 Member Data Documentation	872
15.110.5.1 m_InterfaceData	873
15.110.5.2 TLInterface	873
15.111 IInterfaceArrivalEventHandler Class Reference	873
15.111.1 Constructor & Destructor Documentation	874
15.111.1.1 ~IInterfaceArrivalEventHandler()	874
15.111.1.2 IInterfaceArrivalEventHandler() [1/2]	874
15.111.1.3 IInterfaceArrivalEventHandler() [2/2]	875
15.111.2 Member Function Documentation	875
15.111.2.1 OnInterfaceArrival()	875
15.111.2.2 operator=()	875
15.112 IInterfaceEventHandler Class Reference	875
15.112.1 Constructor & Destructor Documentation	876
15.112.1.1 ~IInterfaceEventHandler()	876
15.112.1.2 IInterfaceEventHandler() [1/2]	876
15.112.1.3 IInterfaceEventHandler() [2/2]	877
15.112.2 Member Function Documentation	877
15.112.2.1 OnDeviceArrival()	877
15.112.2.2 OnDeviceRemoval()	877
15.112.2.3 operator=()	877
15.113 IInterfaceList Class Reference	878
15.113.1 Detailed Description	878
15.113.2 Constructor & Destructor Documentation	879
15.113.2.1 ~IInterfaceList()	879
15.113.2.2 IInterfaceList() [1/2]	879
15.113.2.3 IInterfaceList() [2/2]	879
15.113.3 Member Function Documentation	879
15.113.3.1 Add()	879
15.113.3.2 Append()	879
15.113.3.3 Clear()	880
15.113.3.4 GetByIndex()	880
15.113.3.5 GetSize()	880
15.113.3.6 operator=()	880
15.113.3.7 operator[]()	880
15.113.3.8 Remove()	880
15.113.4 Friends And Related Function Documentation	881
15.113.4.1 InterfaceListImpl	881
15.113.5 Member Data Documentation	881

15.113.5.1 m_InterfaceListData	881
15.114 IInterfaceRemovalEventHandler Class Reference	881
15.114.1 Constructor & Destructor Documentation	882
15.114.1.1 ~IInterfaceRemovalEventHandler()	882
15.114.1.2 IInterfaceRemovalEventHandler() [1/2]	882
15.114.1.3 IInterfaceRemovalEventHandler() [2/2]	883
15.114.2 Member Function Documentation	883
15.114.2.1 OnInterfaceRemoval()	883
15.114.2.2 operator=()	883
15.115 ILoggingEventHandler Class Reference	883
15.115.1 Constructor & Destructor Documentation	884
15.115.1.1 ~ILoggingEventHandler()	884
15.115.1.2 ILoggingEventHandler() [1/2]	884
15.115.1.3 ILoggingEventHandler() [2/2]	885
15.115.2 Member Function Documentation	885
15.115.2.1 OnLogEvent()	885
15.115.2.2 operator=()	885
15.116 Image Class Reference	885
15.116.1 Detailed Description	889
15.116.2 Constructor & Destructor Documentation	889
15.116.2.1 ~Image()	889
15.116.2.2 Image() [1/4]	889
15.116.2.3 Image() [2/4]	889
15.116.2.4 Image() [3/4]	890
15.116.2.5 Image() [4/4]	890
15.116.3 Member Function Documentation	890
15.116.3.1 CalculateStatistics()	890
15.116.3.2 CheckCRC()	890
15.116.3.3 Create() [1/4]	891
15.116.3.4 Create() [2/4]	891
15.116.3.5 Create() [3/4]	891
15.116.3.6 Create() [4/4]	892
15.116.3.7 CreateShared()	892
15.116.3.8 DeepCopy() [1/2]	893
15.116.3.9 DeepCopy() [2/2]	893
15.116.3.10 GetBitsPerPixel()	893
15.116.3.11 GetBufferSize()	893
15.116.3.12 GetChunkData()	894
15.116.3.13 GetChunkLayoutId()	894
15.116.3.14 GetColorProcessing()	894
15.116.3.15 GetData()	895
15.116.3.16 GetDataAbsoluteMax()	895

15.116.3.17 GetDataAbsoluteMin()	895
15.116.3.18 GetFrameID()	896
15.116.3.19 GetHeight()	896
15.116.3.20 GetID()	896
15.116.3.21 GetImageData()	897
15.116.3.22 GetImageSize()	897
15.116.3.23 GetImageStatus()	897
15.116.3.24 GetImageStatusDescription()	898
15.116.3.25 GetNumChannels()	898
15.116.3.26 GetPayloadType()	898
15.116.3.27 GetPixelFormat()	899
15.116.3.28 GetPixelFormatIntType()	899
15.116.3.29 GetPixelFormatName()	899
15.116.3.30 GetPrivateData()	900
15.116.3.31 GetStreamIndex()	900
15.116.3.32 GetStride()	900
15.116.3.33 GetTimeStamp()	901
15.116.3.34 GetTLPayloadType()	901
15.116.3.35 GetTLPixelFormat()	901
15.116.3.36 GetTLPixelFormatNamespace()	902
15.116.3.37 GetValidPayloadSize()	902
15.116.3.38 GetWidth()	902
15.116.3.39 GetXOffset()	903
15.116.3.40 GetXPadding()	903
15.116.3.41 GetYOffset()	903
15.116.3.42 GetYPadding()	904
15.116.3.43 HasChunkData()	904
15.116.3.44 HasCRC()	904
15.116.3.45 IsCompressed()	905
15.116.3.46 IsIncomplete()	905
15.116.3.47 IsInUse()	905
15.116.3.48 Load()	905
15.116.3.49 Release()	906
15.116.3.50 ResetImage() [1/3]	906
15.116.3.51 ResetImage() [2/3]	907
15.116.3.52 ResetImage() [3/3]	907
15.116.3.53 Save() [1/9]	907
15.116.3.54 Save() [2/9]	908
15.116.3.55 Save() [3/9]	908
15.116.3.56 Save() [4/9]	909
15.116.3.57 Save() [5/9]	909
15.116.3.58 Save() [6/9]	909

15.116.3.59 Save() [7/9]	910
15.116.3.60 Save() [8/9]	910
15.116.3.61 Save() [9/9]	910
15.116.4 Friends And Related Function Documentation	911
15.116.4.1 IDataStream	911
15.116.4.2 ImageConverter	911
15.116.4.3 ImageConverterIpp	911
15.116.4.4 ImageFiler	911
15.116.4.5 ImageImpl	911
15.116.4.6 ImageListImpl	911
15.116.4.7 ImageProcessorImpl	911
15.116.4.8 ImageStatsCalculator	912
15.116.4.9 ImageUtilityImpl	912
15.116.4.10 ImageUtilityPlanarImpl	912
15.116.4.11 ImageUtilityPolarizationImpl	912
15.116.4.12 Stream	912
15.117 ImageEventHandler Class Reference	913
15.117.1 Detailed Description	914
15.117.2 Constructor & Destructor Documentation	914
15.117.2.1 ImageEventHandler()	914
15.117.2.2 ~ImageEventHandler()	914
15.117.3 Member Function Documentation	914
15.117.3.1 OnImageEvent()	914
15.117.3.2 operator=()	915
15.118 ImageEventHandlerImpl Class Reference	915
15.118.1 Constructor & Destructor Documentation	916
15.118.1.1 ImageEventHandlerImpl() [1/2]	916
15.118.1.2 ~ImageEventHandlerImpl() [1/2]	917
15.118.1.3 ImageEventHandlerImpl() [2/2]	917
15.118.1.4 ~ImageEventHandlerImpl() [2/2]	917
15.118.2 Member Function Documentation	917
15.118.2.1 getImageCount()	917
15.118.2.2 getMaxImages()	917
15.118.2.3 OnImageEvent() [1/2]	917
15.118.2.4 OnImageEvent() [2/2]	918
15.119 ImageInfo Struct Reference	918
15.119.1 Constructor & Destructor Documentation	918
15.119.1.1 ImageInfo()	918
15.119.2 Member Data Documentation	918
15.119.2.1 imageFile	919
15.119.2.2 imageFileName	919
15.119.2.3 imageHeight	919

15.119.2.4	imageWidth	919
15.119.2.5	pixelFormat	919
15.120	ImageList Class Reference	920
15.120.1	Constructor & Destructor Documentation	921
15.120.1.1	ImageList() [1/2]	921
15.120.1.2	~ImageList()	921
15.120.1.3	ImageList() [2/2]	922
15.120.2	Member Function Documentation	922
15.120.2.1	Add()	922
15.120.2.2	Append()	922
15.120.2.3	Clear()	922
15.120.2.4	GetByIndex()	923
15.120.2.5	GetByPixelFormat()	923
15.120.2.6	GetSize()	924
15.120.2.7	Load()	924
15.120.2.8	operator=()	924
15.120.2.9	operator[]() [1/2]	924
15.120.2.10	operator[]() [2/2]	925
15.120.2.11	Release()	925
15.120.2.12	RemoveByIndex()	925
15.120.2.13	RemoveByPixelFormat()	925
15.120.2.14	Save()	926
15.121	ImageListEventHandler Class Reference	926
15.121.1	Detailed Description	927
15.121.2	Constructor & Destructor Documentation	927
15.121.2.1	ImageListEventHandler()	928
15.121.2.2	~ImageListEventHandler()	928
15.121.3	Member Function Documentation	928
15.121.3.1	OnImageListEvent()	928
15.121.3.2	operator=()	928
15.122	ImageProcessor Class Reference	929
15.122.1	Detailed Description	930
15.122.2	Constructor & Destructor Documentation	933
15.122.2.1	ImageProcessor() [1/2]	933
15.122.2.2	~ImageProcessor()	933
15.122.2.3	ImageProcessor() [2/2]	934
15.122.3	Member Function Documentation	934
15.122.3.1	ApplyGamma() [1/2]	934
15.122.3.2	ApplyGamma() [2/2]	934
15.122.3.3	Convert() [1/5]	935
15.122.3.4	Convert() [2/5]	935
15.122.3.5	Convert() [3/5]	936

15.122.3.6 Convert() [4 / 5]	937
15.122.3.7 Convert() [5 / 5]	937
15.122.3.8 GetColorProcessing()	938
15.122.3.9 GetNumDecompressionThreads()	939
15.122.3.10 operator=()	939
15.122.3.11 SetColorProcessing()	939
15.122.3.12 SetNumDecompressionThreads()	940
15.122.4 Friends And Related Function Documentation	940
15.122.4.1 ImageFiler	940
15.123 ImagePtr Class Reference	940
15.123.1 Detailed Description	941
15.123.2 Constructor & Destructor Documentation	941
15.123.2.1 ImagePtr() [1 / 4]	941
15.123.2.2 ImagePtr() [2 / 4]	942
15.123.2.3 ImagePtr() [3 / 4]	942
15.123.2.4 ImagePtr() [4 / 4]	942
15.123.2.5 ~ImagePtr()	942
15.123.3 Member Function Documentation	942
15.123.3.1 operator=()	942
15.124 ImageStatistics Class Reference	943
15.124.1 Detailed Description	944
15.124.2 Constructor & Destructor Documentation	944
15.124.2.1 ImageStatistics() [1 / 2]	944
15.124.2.2 ~ImageStatistics()	945
15.124.2.3 ImageStatistics() [2 / 2]	945
15.124.3 Member Function Documentation	945
15.124.3.1 DisableAll()	945
15.124.3.2 EnableAll()	945
15.124.3.3 EnableGreyOnly()	945
15.124.3.4 EnableHSLOnly()	946
15.124.3.5 EnableRGBOnly()	946
15.124.3.6 GetChannelStatus()	946
15.124.3.7 GetHistogram()	946
15.124.3.8 GetMean()	947
15.124.3.9 GetNumPixelValues()	947
15.124.3.10 GetPixelValueRange()	947
15.124.3.11 GetRange()	948
15.124.3.12 GetStatistics()	948
15.124.3.13 operator=()	949
15.124.3.14 SetChannelStatus()	949
15.124.4 Friends And Related Function Documentation	950
15.124.4.1 ImageStatsCalculator	950

15.125 ImageUtility Class Reference	950
15.125.1 Detailed Description	950
15.125.2 Member Function Documentation	950
15.125.2.1 CreateNormalized() [1/5]	951
15.125.2.2 CreateNormalized() [2/5]	951
15.125.2.3 CreateNormalized() [3/5]	952
15.125.2.4 CreateNormalized() [4/5]	952
15.125.2.5 CreateNormalized() [5/5]	953
15.125.2.6 CreateScaled() [1/2]	953
15.125.2.7 CreateScaled() [2/2]	953
15.126 ImageUtilityCCM Class Reference	954
15.126.1 Detailed Description	954
15.126.2 Member Function Documentation	954
15.126.2.1 CreateColorCorrected() [1/2]	955
15.126.2.2 CreateColorCorrected() [2/2]	955
15.127 ImageUtilityHeatmap Class Reference	956
15.127.1 Detailed Description	957
15.127.2 Member Function Documentation	957
15.127.2.1 CreateHeatmap() [1/2]	957
15.127.2.2 CreateHeatmap() [2/2]	957
15.127.2.3 GetHeatmapColorGradient()	958
15.127.2.4 GetHeatmapRange()	958
15.127.2.5 SetHeatmapColorGradient()	959
15.127.2.6 SetHeatmapRange()	959
15.128 ImageUtilityPolarization Class Reference	959
15.128.1 Detailed Description	960
15.128.2 Member Function Documentation	960
15.128.2.1 CreateAolp() [1/2]	961
15.128.2.2 CreateAolp() [2/2]	961
15.128.2.3 CreateDolp() [1/2]	961
15.128.2.4 CreateDolp() [2/2]	962
15.128.2.5 CreateGlareReduced() [1/2]	962
15.128.2.6 CreateGlareReduced() [2/2]	963
15.128.2.7 CreateStokesS0() [1/2]	963
15.128.2.8 CreateStokesS0() [2/2]	964
15.128.2.9 CreateStokesS1() [1/2]	964
15.128.2.10 CreateStokesS1() [2/2]	964
15.128.2.11 CreateStokesS2() [1/2]	965
15.128.2.12 CreateStokesS2() [2/2]	965
15.128.2.13 ExtractPolarQuadrant() [1/2]	967
15.128.2.14 ExtractPolarQuadrant() [2/2]	967
15.129 InferenceBoundingBox Struct Reference	968

15.129.1 Detailed Description	968
15.129.2 Member Data Documentation	968
15.129.2.1 boxType	968
15.129.2.2 circle	968
15.129.2.3 classId	969
15.129.2.4 confidence	969
15.129.2.5 rect	969
15.129.2.6 rotatedRect	969
15.130 InferenceBoundingBoxResult Class Reference	969
15.130.1 Detailed Description	970
15.130.2 Constructor & Destructor Documentation	970
15.130.2.1 InferenceBoundingBoxResult() [1/3]	970
15.130.2.2 ~InferenceBoundingBoxResult()	970
15.130.2.3 InferenceBoundingBoxResult() [2/3]	970
15.130.2.4 InferenceBoundingBoxResult() [3/3]	970
15.130.3 Member Function Documentation	971
15.130.3.1 GetBoxAt()	971
15.130.3.2 GetBoxCount()	971
15.130.3.3 GetBoxSize()	971
15.130.3.4 GetVersion()	971
15.130.3.5 operator=()	972
15.130.4 Friends And Related Function Documentation	972
15.130.4.1 ChunkDataImpl	972
15.131 InferenceBoxCircle Struct Reference	972
15.131.1 Member Data Documentation	972
15.131.1.1 centerXCoord	972
15.131.1.2 centerYCoord	972
15.131.1.3 radius	973
15.132 InferenceBoxRect Struct Reference	973
15.132.1 Detailed Description	973
15.132.2 Member Data Documentation	973
15.132.2.1 bottomRightXCoord	973
15.132.2.2 bottomRightYCoord	973
15.132.2.3 topLeftXCoord	973
15.132.2.4 topLeftYCoord	974
15.133 InferenceBoxRotatedRect Struct Reference	974
15.133.1 Member Data Documentation	974
15.133.1.1 bottomRightXCoord	974
15.133.1.2 bottomRightYCoord	974
15.133.1.3 rotationAngle	974
15.133.1.4 topLeftXCoord	974
15.133.1.5 topLeftYCoord	975

15.134 <code>int64_autovector_t</code> Class Reference	975
15.134.1 Detailed Description	975
15.134.2 Constructor & Destructor Documentation	975
15.134.2.1 <code>int64_autovector_t()</code> [1/3]	975
15.134.2.2 <code>int64_autovector_t()</code> [2/3]	976
15.134.2.3 <code>int64_autovector_t()</code> [3/3]	976
15.134.2.4 <code>~int64_autovector_t()</code>	976
15.134.3 Member Function Documentation	976
15.134.3.1 <code>operator delete()</code>	976
15.134.3.2 <code>operator new()</code>	976
15.134.3.3 <code>operator=()</code>	976
15.134.3.4 <code>operator[]()</code> [1/2]	977
15.134.3.5 <code>operator[]()</code> [2/2]	977
15.134.3.6 <code>size()</code>	977
15.134.4 Member Data Documentation	977
15.134.4.1 <code>_pCount</code>	977
15.134.4.2 <code>_pv</code>	977
15.135 <code>IntegerNode</code> Class Reference	978
15.135.1 Detailed Description	980
15.135.2 Constructor & Destructor Documentation	980
15.135.2.1 <code>IntegerNode()</code> [1/2]	980
15.135.2.2 <code>IntegerNode()</code> [2/2]	980
15.135.2.3 <code>~IntegerNode()</code>	980
15.135.3 Member Function Documentation	980
15.135.3.1 <code>GetFloatAlias()</code>	981
15.135.3.2 <code>GetInc()</code>	981
15.135.3.3 <code>GetIncMode()</code>	981
15.135.3.4 <code>GetListOfValidValues()</code>	981
15.135.3.5 <code>GetMax()</code>	981
15.135.3.6 <code>GetMin()</code>	981
15.135.3.7 <code>GetRepresentation()</code>	982
15.135.3.8 <code>GetUnit()</code>	982
15.135.3.9 <code>GetValue()</code>	982
15.135.3.10 <code>ImposeMax()</code>	982
15.135.3.11 <code>ImposeMin()</code>	982
15.135.3.12 <code>operator()()</code>	983
15.135.3.13 <code>operator*()</code>	983
15.135.3.14 <code>operator=()</code>	983
15.135.3.15 <code>SetReference()</code>	983
15.135.3.16 <code>SetValue()</code>	983
15.136 Interface Class Reference	984
15.136.1 Detailed Description	985

15.136.2 Constructor & Destructor Documentation	985
15.136.2.1 ~Interface()	985
15.136.3 Member Function Documentation	985
15.136.3.1 GetCameras()	986
15.136.3.2 GetTLNodeMap()	986
15.136.3.3 IsInUse()	986
15.136.3.4 IsValid()	987
15.136.3.5 RegisterEventHandler()	987
15.136.3.6 SendActionCommand()	987
15.136.3.7 UnregisterEventHandler()	988
15.136.3.8 UpdateCameras()	988
15.136.4 Friends And Related Function Documentation	989
15.136.4.1 InterfaceInternal	989
15.137 InterfaceArrivalEventHandler Class Reference	989
15.137.1 Detailed Description	990
15.137.2 Constructor & Destructor Documentation	990
15.137.2.1 InterfaceArrivalEventHandler()	991
15.137.2.2 ~InterfaceArrivalEventHandler()	991
15.137.3 Member Function Documentation	991
15.137.3.1 OnInterfaceArrival()	991
15.137.3.2 operator=()	991
15.138 InterfaceEventHandler Class Reference	992
15.138.1 Detailed Description	993
15.138.2 Constructor & Destructor Documentation	993
15.138.2.1 InterfaceEventHandler()	993
15.138.2.2 ~InterfaceEventHandler()	993
15.138.3 Member Function Documentation	993
15.138.3.1 OnDeviceArrival()	993
15.138.3.2 OnDeviceRemoval()	994
15.138.3.3 operator=()	994
15.139 InterfaceEventHandlerImpl Class Reference	995
15.139.1 Constructor & Destructor Documentation	996
15.139.1.1 InterfaceEventHandlerImpl() [1/3]	997
15.139.1.2 ~InterfaceEventHandlerImpl() [1/2]	997
15.139.1.3 InterfaceEventHandlerImpl() [2/3]	997
15.139.1.4 InterfaceEventHandlerImpl() [3/3]	997
15.139.1.5 ~InterfaceEventHandlerImpl() [2/2]	997
15.139.2 Member Function Documentation	997
15.139.2.1 GetInterfaceId()	997
15.139.2.2 OnDeviceArrival() [1/2]	997
15.139.2.3 OnDeviceArrival() [2/2]	998
15.139.2.4 OnDeviceRemoval() [1/2]	998

15.139.2.5 OnDeviceRemoval() [2/2]	998
15.139.2.6 PrintGenericHandlerMessage()	999
15.140 InterfaceList Class Reference	999
15.140.1 Detailed Description	1000
15.140.2 Constructor & Destructor Documentation	1000
15.140.2.1 InterfaceList() [1/2]	1000
15.140.2.2 ~InterfaceList()	1001
15.140.2.3 InterfaceList() [2/2]	1001
15.140.3 Member Function Documentation	1001
15.140.3.1 Add()	1001
15.140.3.2 Append()	1001
15.140.3.3 Clear()	1002
15.140.3.4 GetByIndex()	1002
15.140.3.5 GetByInterfaceID()	1002
15.140.3.6 GetSize()	1003
15.140.3.7 operator=()	1003
15.140.3.8 operator[]()	1003
15.140.3.9 Remove()	1003
15.140.4 Friends And Related Function Documentation	1004
15.140.4.1 ProducerImpl	1004
15.140.4.2 SystemImpl	1004
15.141 InterfacePtr Class Reference	1004
15.141.1 Detailed Description	1005
15.141.2 Constructor & Destructor Documentation	1005
15.141.2.1 InterfacePtr() [1/4]	1005
15.141.2.2 InterfacePtr() [2/4]	1005
15.141.2.3 InterfacePtr() [3/4]	1005
15.141.2.4 InterfacePtr() [4/4]	1006
15.142 InterfaceRemovalEventHandler Class Reference	1006
15.142.1 Detailed Description	1007
15.142.2 Constructor & Destructor Documentation	1007
15.142.2.1 InterfaceRemovalEventHandler()	1007
15.142.2.2 ~InterfaceRemovalEventHandler()	1007
15.142.3 Member Function Documentation	1007
15.142.3.1 OnInterfaceRemoval()	1007
15.142.3.2 operator=()	1008
15.143 IntRegNode Class Reference	1008
15.143.1 Detailed Description	1009
15.143.2 Constructor & Destructor Documentation	1009
15.143.2.1 IntRegNode() [1/2]	1010
15.143.2.2 IntRegNode() [2/2]	1010
15.143.2.3 ~IntRegNode()	1010

15.143.3 Member Function Documentation	1010
15.143.3.1 SetReference()	1010
15.144 ISystem Class Reference	1011
15.144.1 Detailed Description	1012
15.144.2 Constructor & Destructor Documentation	1012
15.144.2.1 ~ISystem()	1012
15.144.2.2 ISystem() [1/2]	1012
15.144.2.3 ISystem() [2/2]	1012
15.144.3 Member Function Documentation	1012
15.144.3.1 GetCameras()	1013
15.144.3.2 GetInterfaces()	1013
15.144.3.3 GetLibraryVersion()	1013
15.144.3.4 GetLoggingEventPriorityLevel()	1013
15.144.3.5 GetTLNodeMap()	1013
15.144.3.6 IsInUse()	1013
15.144.3.7 operator=()	1014
15.144.3.8 RegisterEventHandler()	1014
15.144.3.9 RegisterLoggingEventHandler()	1014
15.144.3.10 ReleaseInstance()	1014
15.144.3.11 SendActionCommand()	1014
15.144.3.12 SetLoggingEventPriorityLevel()	1015
15.144.3.13 UnregisterAllLoggingEventHandlers()	1015
15.144.3.14 UnregisterEventHandler()	1015
15.144.3.15 UnregisterLoggingEventHandler()	1015
15.144.3.16 UpdateCameras()	1015
15.144.3.17 UpdateInterfaceList()	1015
15.144.4 Friends And Related Function Documentation	1016
15.144.4.1 SystemPtrInternal	1016
15.144.5 Member Data Documentation	1016
15.144.5.1 TLSystem	1016
15.145 ISystemEventHandler Class Reference	1016
15.145.1 Constructor & Destructor Documentation	1017
15.145.1.1 ~ISystemEventHandler()	1017
15.145.1.2 ISystemEventHandler() [1/2]	1017
15.145.1.3 ISystemEventHandler() [2/2]	1018
15.145.2 Member Function Documentation	1018
15.145.2.1 OnInterfaceArrival()	1018
15.145.2.2 OnInterfaceRemoval()	1018
15.145.2.3 operator=()	1018
15.146 JPEGOption Struct Reference	1018
15.146.1 Detailed Description	1019
15.146.2 Constructor & Destructor Documentation	1019

15.146.2.1 JPEGOption()	1019
15.146.3 Member Data Documentation	1019
15.146.3.1 progressive	1019
15.146.3.2 quality	1019
15.146.3.3 reserved	1020
15.147 JPG2Option Struct Reference	1020
15.147.1 Detailed Description	1020
15.147.2 Constructor & Destructor Documentation	1020
15.147.2.1 JPG2Option()	1020
15.147.3 Member Data Documentation	1020
15.147.3.1 quality	1021
15.147.3.2 reserved	1021
15.148 LibraryVersion Struct Reference	1021
15.148.1 Detailed Description	1021
15.148.2 Member Data Documentation	1021
15.148.2.1 build	1022
15.148.2.2 major	1022
15.148.2.3 minor	1022
15.148.2.4 type	1022
15.149 LockableObject< Object >::Lock Class Reference	1022
15.149.1 Detailed Description	1023
15.149.2 Constructor & Destructor Documentation	1023
15.149.2.1 Lock()	1023
15.149.2.2 ~Lock()	1023
15.150 LockableObject< Object > Class Template Reference	1023
15.150.1 Detailed Description	1024
15.150.2 Member Function Documentation	1024
15.150.2.1 GetLock()	1024
15.150.3 Friends And Related Function Documentation	1024
15.150.3.1 Lock	1024
15.150.4 Member Data Documentation	1025
15.150.4.1 m_Lock	1025
15.151 LoggingEventData Class Reference	1025
15.151.1 Detailed Description	1026
15.151.2 Constructor & Destructor Documentation	1026
15.151.2.1 ~LoggingEventData()	1026
15.151.2.2 LoggingEventData()	1026
15.151.3 Member Function Documentation	1026
15.151.3.1 GetCategoryName()	1026
15.151.3.2 GetLogMessage()	1026
15.151.3.3 GetNDC()	1027
15.151.3.4 GetPriority()	1027

15.151.3.5 GetPriorityName()	1027
15.151.3.6 GetThreadName()	1027
15.151.3.7 GetTimestamp()	1028
15.151.4 Friends And Related Function Documentation	1028
15.151.4.1 SystemImpl	1028
15.152 LoggingEventDataPtr Class Reference	1028
15.152.1 Detailed Description	1029
15.152.2 Constructor & Destructor Documentation	1029
15.152.2.1 LoggingEventDataPtr() [1/4]	1029
15.152.2.2 LoggingEventDataPtr() [2/4]	1029
15.152.2.3 LoggingEventDataPtr() [3/4]	1029
15.152.2.4 LoggingEventDataPtr() [4/4]	1030
15.153 LoggingEventHandler Class Reference	1030
15.153.1 Detailed Description	1031
15.153.2 Constructor & Destructor Documentation	1031
15.153.2.1 LoggingEventHandler()	1032
15.153.2.2 ~LoggingEventHandler()	1032
15.153.3 Member Function Documentation	1032
15.153.3.1 OnLogEvent()	1032
15.153.3.2 operator=()	1032
15.154 LoggingEventHandlerImpl Class Reference	1033
15.155 Member_NodeCallback< Client, Member > Class Template Reference	1034
15.155.1 Detailed Description	1035
15.155.2 Member Typedef Documentation	1035
15.155.2.1 PMEMBERFUNC	1035
15.155.3 Constructor & Destructor Documentation	1035
15.155.3.1 Member_NodeCallback()	1035
15.155.4 Member Function Documentation	1035
15.155.4.1 Destroy()	1036
15.155.4.2 operator>()()	1036
15.156 MJPGOption Struct Reference	1036
15.156.1 Detailed Description	1036
15.156.2 Constructor & Destructor Documentation	1037
15.156.2.1 MJPGOption()	1037
15.156.3 Member Data Documentation	1037
15.156.3.1 frameRate	1037
15.156.3.2 height	1037
15.156.3.3 quality	1037
15.156.3.4 reserved	1037
15.156.3.5 width	1038
15.157 Node Class Reference	1038
15.157.1 Detailed Description	1040

15.157.2 Constructor & Destructor Documentation	1041
15.157.2.1 Node() [1/2]	1041
15.157.2.2 Node() [2/2]	1041
15.157.2.3 ~Node()	1041
15.157.3 Member Function Documentation	1041
15.157.3.1 DeregisterCallback()	1041
15.157.3.2 GetAccessMode()	1042
15.157.3.3 GetAlias()	1042
15.157.3.4 GetCachingMode()	1042
15.157.3.5 GetCastAlias()	1042
15.157.3.6 GetChildren()	1042
15.157.3.7 GetDescription()	1043
15.157.3.8 GetDeviceName()	1043
15.157.3.9 GetDisplayName()	1043
15.157.3.10 GetDocuURL()	1043
15.157.3.11 GetEventID()	1043
15.157.3.12 GetName()	1043
15.157.3.13 GetNameSpace()	1044
15.157.3.14 GetNodeHandle()	1044
15.157.3.15 GetNodeMap()	1044
15.157.3.16 GetParents()	1044
15.157.3.17 GetPollingTime()	1044
15.157.3.18 GetPrincipalInterfaceType()	1045
15.157.3.19 GetProperty()	1045
15.157.3.20 GetPropertyNames()	1045
15.157.3.21 GetSelectedFeatures()	1045
15.157.3.22 GetSelectingFeatures()	1045
15.157.3.23 GetToolTip()	1046
15.157.3.24 GetVisibility()	1046
15.157.3.25 ImposeAccessMode()	1046
15.157.3.26 ImposeVisibility()	1046
15.157.3.27 InvalidateNode()	1046
15.157.3.28 IsAccessModeCacheable()	1046
15.157.3.29 IsCachable()	1047
15.157.3.30 IsDeprecated()	1047
15.157.3.31 IsFeature()	1047
15.157.3.32 IsSelector()	1047
15.157.3.33 IsStreamable()	1047
15.157.3.34 operator!=(())	1047
15.157.3.35 operator==(())	1048
15.157.3.36 RegisterCallback()	1048
15.157.3.37 SetNodeHandle()	1048

15.157.3.38 SetNodeMap()	1048
15.157.3.39 SetReference() [1/2]	1048
15.157.3.40 SetReference() [2/2]	1048
15.157.4 Member Data Documentation	1049
15.157.4.1 m_Callbacks	1049
15.157.4.2 m_pNodeData	1049
15.157.4.3 m_pNodeMap	1049
15.158 NodeMap Class Reference	1049
15.158.1 Detailed Description	1051
15.158.2 Constructor & Destructor Documentation	1051
15.158.2.1 NodeMap()	1052
15.158.2.2 ~NodeMap()	1052
15.158.3 Member Function Documentation	1052
15.158.3.1 ClearXMLCache()	1052
15.158.3.2 Connect() [1/2]	1052
15.158.3.3 Connect() [2/2]	1052
15.158.3.4 Destroy()	1053
15.158.3.5 GetDeviceName()	1053
15.158.3.6 GetDeviceVersion()	1053
15.158.3.7 GetGenApiVersion()	1053
15.158.3.8 GetLock()	1053
15.158.3.9 GetModelName()	1053
15.158.3.10 GetNode()	1054
15.158.3.11 GetNodeMapHandle()	1054
15.158.3.12 GetNodes()	1054
15.158.3.13 GetNumNodes()	1054
15.158.3.14 GetProductGuid()	1054
15.158.3.15 GetSchemaVersion()	1054
15.158.3.16 GetStandardNameSpace()	1055
15.158.3.17 GetSupportedSchemaVersions()	1055
15.158.3.18 GetToolTip()	1055
15.158.3.19 GetVendorName()	1055
15.158.3.20 GetVersionGuid()	1056
15.158.3.21 InvalidateNodes()	1056
15.158.3.22 LoadXMLFromFile()	1056
15.158.3.23 LoadXMLFromFileInject()	1056
15.158.3.24 LoadXMLFromString()	1056
15.158.3.25 LoadXMLFromStringInject()	1057
15.158.3.26 LoadXMLFromZIPData()	1057
15.158.3.27 LoadXMLFromZIPFile()	1057
15.158.3.28 Poll()	1057
15.158.4 Member Data Documentation	1057

15.158.4.1 _Ptr	1057
15.159 CNodeMapFactory::NodeStatistics_t Struct Reference	1058
15.159.1 Member Data Documentation	1058
15.159.1.1 NumLinks	1058
15.159.1.2 NumNodes	1058
15.159.1.3 NumProperties	1058
15.159.1.4 NumStrings	1058
15.160 ODevFileStreamBase< CharType, Traits > Class Template Reference	1059
15.160.1 Member Typedef Documentation	1060
15.160.1.1 filebuf_type	1060
15.160.1.2 ios_type	1060
15.160.1.3 ostream_type	1060
15.160.2 Member Function Documentation	1060
15.160.2.1 close()	1060
15.160.2.2 is_open()	1060
15.160.2.3 open()	1060
15.160.2.4 rdbuf()	1061
15.161 ODevFileStreamBuf< CharType, Traits > Class Template Reference	1061
15.161.1 Constructor & Destructor Documentation	1062
15.161.1.1 ODevFileStreamBuf()	1062
15.161.1.2 ~ODevFileStreamBuf()	1062
15.161.2 Member Function Documentation	1062
15.161.2.1 close()	1062
15.161.2.2 is_open()	1062
15.161.2.3 open()	1063
15.161.2.4 overflow()	1063
15.161.2.5 sync()	1063
15.161.2.6 xspn()	1063
15.162 PGMOption Struct Reference	1063
15.162.1 Detailed Description	1064
15.162.2 Constructor & Destructor Documentation	1064
15.162.2.1 PGMOption()	1064
15.162.3 Member Data Documentation	1064
15.162.3.1 binaryFile	1064
15.162.3.2 reserved	1064
15.163 PNGOption Struct Reference	1064
15.163.1 Detailed Description	1065
15.163.2 Constructor & Destructor Documentation	1065
15.163.2.1 PNGOption()	1065
15.163.3 Member Data Documentation	1065
15.163.3.1 compressionLevel	1065
15.163.3.2 interlaced	1065

15.163.3.3 reserved	1066
15.164 PortNode Class Reference	1066
15.164.1 Detailed Description	1068
15.164.2 Constructor & Destructor Documentation	1068
15.164.2.1 PortNode() [1/2]	1068
15.164.2.2 PortNode() [2/2]	1068
15.164.2.3 ~PortNode()	1068
15.164.3 Member Function Documentation	1068
15.164.3.1 CacheChunkData()	1068
15.164.3.2 GetChunkID()	1069
15.164.3.3 GetPortHandle()	1069
15.164.3.4 GetSwapEndianness()	1069
15.164.3.5 Read()	1069
15.164.3.6 Replay()	1069
15.164.3.7 SetPortImpl()	1070
15.164.3.8 SetReference() [1/5]	1070
15.164.3.9 SetReference() [2/5]	1070
15.164.3.10 SetReference() [3/5]	1070
15.164.3.11 SetReference() [4/5]	1070
15.164.3.12 SetReference() [5/5]	1071
15.164.3.13 StartRecording()	1071
15.164.3.14 StopRecording()	1071
15.164.3.15 Write()	1071
15.165 PortRecorder Class Reference	1072
15.165.1 Detailed Description	1073
15.165.2 Constructor & Destructor Documentation	1073
15.165.2.1 PortRecorder()	1073
15.165.2.2 ~PortRecorder()	1073
15.165.3 Member Function Documentation	1073
15.165.3.1 GetAccessMode()	1074
15.165.3.2 Read()	1074
15.165.3.3 Replay()	1074
15.165.3.4 SetReference()	1074
15.165.3.5 StartRecording()	1075
15.165.3.6 StopRecording()	1075
15.165.3.7 Write()	1075
15.166 PortReplay Class Reference	1076
15.166.1 Detailed Description	1077
15.166.2 Constructor & Destructor Documentation	1077
15.166.2.1 PortReplay()	1077
15.166.2.2 ~PortReplay()	1077
15.166.3 Member Function Documentation	1077

15.166.3.1 GetAccessMode()	1078
15.166.3.2 GetPortReplayHandle()	1078
15.166.3.3 Read()	1078
15.166.3.4 Replay()	1078
15.166.3.5 SetReference()	1079
15.166.3.6 Write()	1079
15.167 PPMOption Struct Reference	1079
15.167.1 Detailed Description	1079
15.167.2 Constructor & Destructor Documentation	1080
15.167.2.1 PPMOption()	1080
15.167.3 Member Data Documentation	1080
15.167.3.1 binaryFile	1080
15.167.3.2 reserved	1080
15.168 RegisterNode Class Reference	1080
15.168.1 Detailed Description	1082
15.168.2 Constructor & Destructor Documentation	1082
15.168.2.1 RegisterNode() [1/2]	1082
15.168.2.2 RegisterNode() [2/2]	1082
15.168.2.3 ~RegisterNode()	1082
15.168.3 Member Function Documentation	1082
15.168.3.1 Get()	1083
15.168.3.2 GetAddress()	1084
15.168.3.3 GetLength()	1084
15.168.3.4 Set()	1084
15.168.3.5 SetReference()	1085
15.169 SingleChunkData_t Struct Reference	1085
15.169.1 Member Data Documentation	1085
15.169.1.1 ChunkID	1085
15.169.1.2 ChunkLength	1085
15.169.1.3 ChunkOffset	1085
15.170 SingleChunkDataStr_t Struct Reference	1086
15.170.1 Member Data Documentation	1086
15.170.1.1 ChunkID	1086
15.170.1.2 ChunkLength	1086
15.170.1.3 ChunkOffset	1086
15.171 SIOption Struct Reference	1087
15.171.1 Detailed Description	1087
15.171.2 Constructor & Destructor Documentation	1087
15.171.2.1 SIOption()	1087
15.171.3 Member Data Documentation	1087
15.171.3.1 reserved	1087
15.172 SpinTestCamera Class Reference	1088

15.173 SpinVideo Class Reference	1088
15.173.1 Detailed Description	1089
15.173.2 Constructor & Destructor Documentation	1089
15.173.2.1 SpinVideo()	1089
15.173.2.2 ~SpinVideo()	1089
15.173.3 Member Function Documentation	1089
15.173.3.1 Append()	1089
15.173.3.2 Close()	1090
15.173.3.3 Open() [1/3]	1090
15.173.3.4 Open() [2/3]	1090
15.173.3.5 Open() [3/3]	1092
15.173.3.6 SetMaximumFileSize()	1092
15.174 StringNode Class Reference	1093
15.174.1 Detailed Description	1094
15.174.2 Constructor & Destructor Documentation	1095
15.174.2.1 StringNode() [1/2]	1095
15.174.2.2 StringNode() [2/2]	1095
15.174.2.3 ~StringNode()	1095
15.174.3 Member Function Documentation	1095
15.174.3.1 GetMaxLength()	1095
15.174.3.2 GetValue()	1095
15.174.3.3 operator>()	1096
15.174.3.4 operator*()	1096
15.174.3.5 operator=()	1096
15.174.3.6 SetReference()	1096
15.174.3.7 SetValue()	1096
15.175 StringRegNode Class Reference	1097
15.175.1 Detailed Description	1098
15.175.2 Constructor & Destructor Documentation	1098
15.175.2.1 StringRegNode() [1/2]	1099
15.175.2.2 StringRegNode() [2/2]	1099
15.175.2.3 ~StringRegNode()	1099
15.175.3 Member Function Documentation	1099
15.175.3.1 SetReference()	1099
15.176 System Class Reference	1100
15.176.1 Detailed Description	1101
15.176.2 Constructor & Destructor Documentation	1101
15.176.2.1 ~System()	1102
15.176.2.2 System()	1102
15.176.3 Member Function Documentation	1102
15.176.3.1 GetCameras()	1102
15.176.3.2 GetInstance()	1103

15.176.3.3 GetInterfaces()	1103
15.176.3.4 GetLibraryVersion()	1103
15.176.3.5 GetLoggingEventPriorityLevel()	1104
15.176.3.6 GetTLNodeMap()	1104
15.176.3.7 IsInUse()	1105
15.176.3.8 RegisterEventHandler()	1105
15.176.3.9 RegisterLoggingEventHandler()	1105
15.176.3.10 ReleaseInstance()	1106
15.176.3.11 SendActionCommand()	1106
15.176.3.12 SetLoggingEventPriorityLevel()	1107
15.176.3.13 UnregisterAllLoggingEventHandlers()	1107
15.176.3.14 UnregisterEventHandler()	1107
15.176.3.15 UnregisterLoggingEventHandler()	1108
15.176.3.16 UpdateCameras()	1108
15.176.3.17 UpdateInterfaceList()	1109
15.177 SystemEventHandler Class Reference	1109
15.177.1 Detailed Description	1110
15.177.2 Constructor & Destructor Documentation	1110
15.177.2.1 SystemEventHandler()	1111
15.177.2.2 ~SystemEventHandler()	1111
15.177.3 Member Function Documentation	1111
15.177.3.1 OnInterfaceArrival()	1111
15.177.3.2 OnInterfaceRemoval()	1111
15.177.3.3 operator=()	1112
15.178 SystemEventHandlerImpl Class Reference	1112
15.178.1 Constructor & Destructor Documentation	1113
15.178.1.1 SystemEventHandlerImpl()	1113
15.178.1.2 ~SystemEventHandlerImpl()	1114
15.178.2 Member Function Documentation	1114
15.178.2.1 LockEventHandlerMutex()	1114
15.178.2.2 OnInterfaceArrival()	1114
15.178.2.3 OnInterfaceRemoval()	1114
15.178.2.4 RegisterAllInterfaceEvents()	1115
15.178.2.5 RegisterInterfaceEventToSystem()	1115
15.178.2.6 UnlockEventHandlerMutex()	1115
15.178.2.7 UnregisterAllInterfaceEvents()	1115
15.178.2.8 UnregisterInterfaceEventFromSystem()	1115
15.179 SystemPtr Class Reference	1116
15.179.1 Detailed Description	1117
15.179.2 Constructor & Destructor Documentation	1117
15.179.2.1 SystemPtr() [1/4]	1117
15.179.2.2 SystemPtr() [2/4]	1117

15.179.2.3 SystemPtr() [3/4]	1117
15.179.2.4 SystemPtr() [4/4]	1117
15.179.2.5 ~SystemPtr()	1118
15.180 TIFFOption Struct Reference	1118
15.180.1 Detailed Description	1118
15.180.2 Constructor & Destructor Documentation	1118
15.180.2.1 TIFFOption()	1118
15.180.3 Member Data Documentation	1118
15.180.3.1 compression	1119
15.180.3.2 reserved	1119
15.181 TransportLayerDevice Class Reference	1119
15.181.1 Detailed Description	1121
15.181.2 Constructor & Destructor Documentation	1121
15.181.2.1 TransportLayerDevice() [1/2]	1121
15.181.2.2 ~TransportLayerDevice()	1122
15.181.2.3 TransportLayerDevice() [2/2]	1122
15.181.3 Friends And Related Function Documentation	1122
15.181.3.1 CameraBase	1122
15.181.3.2 CameraInternal	1122
15.181.3.3 ICameraBase	1122
15.181.4 Member Data Documentation	1122
15.181.4.1 DeviceAccessStatus	1122
15.181.4.2 DeviceBootloaderVersion	1123
15.181.4.3 DeviceCurrentSpeed	1123
15.181.4.4 DeviceDisplayName	1123
15.181.4.5 DeviceDriverVersion	1123
15.181.4.6 DeviceEndiannessMechanism	1123
15.181.4.7 DeviceID	1123
15.181.4.8 DeviceInstanceId	1124
15.181.4.9 DevicesUpdater	1124
15.181.4.10 DeviceLinkSpeed	1124
15.181.4.11 DeviceLocation	1124
15.181.4.12 DeviceModelName	1124
15.181.4.13 DeviceMulticastMonitorMode	1124
15.181.4.14 DevicePortId	1125
15.181.4.15 DeviceReset	1125
15.181.4.16 DeviceSerialNumber	1125
15.181.4.17 DeviceType	1125
15.181.4.18 DeviceU3VProtocol	1125
15.181.4.19 DeviceUserID	1125
15.181.4.20 DeviceVendorName	1126
15.181.4.21 DeviceVersion	1126

15.181.4.22 GenICamXMLLocation	1126
15.181.4.23 GenICamXMLPath	1126
15.181.4.24 GevCCP	1126
15.181.4.25 GevDeviceAutoForceIP	1126
15.181.4.26 GevDeviceDiscoverMaximumPacketSize	1127
15.181.4.27 GevDeviceForceGateway	1127
15.181.4.28 GevDeviceForceIP	1127
15.181.4.29 GevDeviceForceIPAddress	1127
15.181.4.30 GevDeviceForceSubnetMask	1127
15.181.4.31 GevDeviceGateway	1127
15.181.4.32 GevDeviceIPAddress	1128
15.181.4.33 GevDeviceIsWrongSubnet	1128
15.181.4.34 GevDeviceMACAddress	1128
15.181.4.35 GevDeviceMaximumPacketSize	1128
15.181.4.36 GevDeviceMaximumRetryCount	1128
15.181.4.37 GevDeviceModelsBigEndian	1128
15.181.4.38 GevDevicePort	1129
15.181.4.39 GevDeviceReadAndWriteTimeout	1129
15.181.4.40 GevDeviceSubnetMask	1129
15.181.4.41 GevVersionMajor	1129
15.181.4.42 GevVersionMinor	1129
15.181.4.43 GUIXMLLocation	1129
15.181.4.44 GUIXMLPath	1130
15.182 TransportLayerInterface Class Reference	1130
15.182.1 Detailed Description	1132
15.182.2 Constructor & Destructor Documentation	1132
15.182.2.1 TransportLayerInterface() [1/2]	1132
15.182.2.2 ~TransportLayerInterface()	1133
15.182.2.3 TransportLayerInterface() [2/2]	1133
15.182.3 Friends And Related Function Documentation	1133
15.182.3.1 IInterface	1133
15.182.3.2 Interface	1133
15.182.3.3 InterfaceInternal	1133
15.182.4 Member Data Documentation	1133
15.182.4.1 ActionCommand	1133
15.182.4.2 DeviceAccessStatus	1134
15.182.4.3 DeviceCount	1134
15.182.4.4 DeviceID	1134
15.182.4.5 DeviceModelName	1134
15.182.4.6 DeviceSelector	1134
15.182.4.7 DeviceSerialNumber	1134
15.182.4.8 DeviceUnlock	1135

15.182.4.9 DeviceUpdateList	1135
15.182.4.10 DeviceVendorName	1135
15.182.4.11 FilterDriverStatus	1135
15.182.4.12 GevActionDeviceKey	1135
15.182.4.13 GevActionGroupKey	1135
15.182.4.14 GevActionGroupMask	1136
15.182.4.15 GevActionTime	1136
15.182.4.16 GevDeviceAutoForceIP	1136
15.182.4.17 GevDeviceForceGateway	1136
15.182.4.18 GevDeviceForceIP	1136
15.182.4.19 GevDeviceForceIPAddress	1136
15.182.4.20 GevDeviceForceSubnetMask	1137
15.182.4.21 GevDeviceGateway	1137
15.182.4.22 GevDeviceIPAddress	1137
15.182.4.23 GevDeviceMACAddress	1137
15.182.4.24 GevDeviceSubnetMask	1137
15.182.4.25 GevInterfaceGateway	1137
15.182.4.26 GevInterfaceGatewaySelector	1138
15.182.4.27 GevInterfaceMACAddress	1138
15.182.4.28 GevInterfaceMTU	1138
15.182.4.29 GevInterfaceReceiveLinkSpeed	1138
15.182.4.30 GevInterfaceSubnetIPAddress	1138
15.182.4.31 GevInterfaceSubnetMask	1138
15.182.4.32 GevInterfaceSubnetSelector	1139
15.182.4.33 GevInterfaceTransmitLinkSpeed	1139
15.182.4.34 HostAdapterDriverVersion	1139
15.182.4.35 HostAdapterName	1139
15.182.4.36 HostAdapterVendor	1139
15.182.4.37 IncompatibleDeviceCount	1139
15.182.4.38 IncompatibleDeviceID	1140
15.182.4.39 IncompatibleDeviceModelName	1140
15.182.4.40 IncompatibleDeviceSelector	1140
15.182.4.41 IncompatibleDeviceVendorName	1140
15.182.4.42 IncompatibleGevDeviceIPAddress	1140
15.182.4.43 IncompatibleGevDeviceMACAddress	1140
15.182.4.44 IncompatibleGevDeviceSubnetMask	1141
15.182.4.45 InterfaceDisplayName	1141
15.182.4.46 InterfaceID	1141
15.182.4.47 InterfaceType	1141
15.182.4.48 POEStatus	1141
15.183 TransportLayerStream Class Reference	1141
15.183.1 Detailed Description	1143

15.183.2 Constructor & Destructor Documentation	1143
15.183.2.1 TransportLayerStream() [1/2]	1143
15.183.2.2 ~TransportLayerStream()	1144
15.183.2.3 TransportLayerStream() [2/2]	1144
15.183.3 Friends And Related Function Documentation	1144
15.183.3.1 CameraBase	1144
15.183.3.2 CameraInternal	1144
15.183.3.3 ICameraBase	1144
15.183.4 Member Data Documentation	1144
15.183.4.1 StreamAnnounceBufferMinimum	1144
15.183.4.2 StreamAnnouncedBufferCount	1145
15.183.4.3 StreamBlockTransferSize	1145
15.183.4.4 StreamBufferAlignment	1145
15.183.4.5 StreamBufferCountManual	1145
15.183.4.6 StreamBufferCountMax	1145
15.183.4.7 StreamBufferCountMode	1145
15.183.4.8 StreamBufferCountResult	1146
15.183.4.9 StreamBufferHandlingMode	1146
15.183.4.10 StreamChunkCountMaximum	1146
15.183.4.11 StreamCRCCheckEnable	1146
15.183.4.12 StreamDeliveredFrameCount	1146
15.183.4.13 StreamDroppedFrameCount	1146
15.183.4.14 StreamID	1147
15.183.4.15 StreamIncompleteFrameCount	1147
15.183.4.16 StreamInputBufferCount	1147
15.183.4.17 StreamIsGrabbing	1147
15.183.4.18 StreamLostFrameCount	1147
15.183.4.19 StreamMissedPacketCount	1147
15.183.4.20 StreamMode	1148
15.183.4.21 StreamOutputBufferCount	1148
15.183.4.22 StreamPacketResendEnable	1148
15.183.4.23 StreamPacketResendMaxRequests	1148
15.183.4.24 StreamPacketResendReceivedPacketCount	1148
15.183.4.25 StreamPacketResendRequestCount	1148
15.183.4.26 StreamPacketResendRequestedPacketCount	1149
15.183.4.27 StreamPacketResendRequestSuccessCount	1149
15.183.4.28 StreamPacketResendTimeout	1149
15.183.4.29 StreamReceivedFrameCount	1149
15.183.4.30 StreamReceivedPacketCount	1149
15.183.4.31 StreamStartedFrameCount	1149
15.183.4.32 StreamType	1150
15.184 TransportLayerSystem Class Reference	1150

15.184.1 Detailed Description	1151
15.184.2 Constructor & Destructor Documentation	1151
15.184.2.1 TransportLayerSystem() [1/2]	1152
15.184.2.2 ~TransportLayerSystem()	1152
15.184.2.3 TransportLayerSystem() [2/2]	1152
15.184.3 Friends And Related Function Documentation	1152
15.184.3.1 ISystem	1152
15.184.3.2 System	1152
15.184.3.3 SystemPtrInternal	1152
15.184.4 Member Data Documentation	1152
15.184.4.1 EnumerateGen2Cameras	1153
15.184.4.2 EnumerateGEVInterfaces	1153
15.184.4.3 EnumerateUSBInterfaces	1153
15.184.4.4 GenTLFNCVersionMajor	1153
15.184.4.5 GenTLFNCVersionMinor	1153
15.184.4.6 GenTLFNCVersionSubMinor	1153
15.184.4.7 GenTLVersionMajor	1154
15.184.4.8 GenTLVersionMinor	1154
15.184.4.9 GevInterfaceDefaultGateway	1154
15.184.4.10 GevInterfaceDefaultIPAddress	1154
15.184.4.11 GevInterfaceDefaultSubnetMask	1154
15.184.4.12 GevInterfaceMACAddress	1154
15.184.4.13 GevVersionMajor	1155
15.184.4.14 GevVersionMinor	1155
15.184.4.15 InterfaceDisplayName	1155
15.184.4.16 InterfaceID	1155
15.184.4.17 InterfaceSelector	1155
15.184.4.18 InterfaceUpdateList	1155
15.184.4.19 TLDisplayName	1156
15.184.4.20 TLFileName	1156
15.184.4.21 TLID	1156
15.184.4.22 TLModelName	1156
15.184.4.23 TLPath	1156
15.184.4.24 TLType	1156
15.184.4.25 TLVendorName	1157
15.184.4.26 TLVersion	1157
15.185 U3V_CHUNK_TRAILER Struct Reference	1157
15.185.1 Detailed Description	1157
15.185.2 Member Data Documentation	1157
15.185.2.1 ChunkID	1157
15.185.2.2 ChunkLength	1158
15.186 U3V_COMMAND_HEADER Struct Reference	1158

15.186.1 Detailed Description	1158
15.186.2 Member Data Documentation	1158
15.186.2.1 CommandId	1158
15.186.2.2 Flags	1158
15.186.2.3 Length	1158
15.186.2.4 Prefix	1159
15.186.2.5 ReqId	1159
15.187 U3V_EVENT_DATA Struct Reference	1159
15.187.1 Detailed Description	1159
15.187.2 Member Data Documentation	1159
15.187.2.1 EventId	1159
15.187.2.2 Reserved	1159
15.187.2.3 Timestamp	1160
15.188 U3V_EVENT_MESSAGE Struct Reference	1160
15.188.1 Detailed Description	1160
15.188.2 Member Data Documentation	1160
15.188.2.1 CommandHeader	1160
15.188.2.2 EventData	1161
15.189 ValueNode Class Reference	1161
15.189.1 Detailed Description	1162
15.189.2 Constructor & Destructor Documentation	1162
15.189.2.1 ValueNode() [1/2]	1162
15.189.2.2 ValueNode() [2/2]	1162
15.189.2.3 ~ValueNode()	1163
15.189.3 Member Function Documentation	1163
15.189.3.1 FromString()	1163
15.189.3.2 GetNode()	1163
15.189.3.3 IsValueCacheValid()	1163
15.189.3.4 SetReference()	1163
15.189.3.5 ToString()	1164
15.190 Version_t Struct Reference	1164
15.190.1 Detailed Description	1164
15.190.2 Member Data Documentation	1164
15.190.2.1 Major	1164
15.190.2.2 Minor	1165
15.190.2.3 SubMinor	1165
16 File Documentation	1167
16.1 doc/spindocs/C++/GettingStarted.dox File Reference	1167
16.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference	1167
16.3 doc/spindocs/shared/Benefits.dox File Reference	1167
16.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference	1167

16.5 doc/spindocs/shared/GenlCamGenTL.dox File Reference	1167
16.6 doc/spindocs/shared/Licensing.dox File Reference	1167
16.7 doc/spindocs/shared/Maintenance.dox File Reference	1167
16.8 include/AVIRecorder.h File Reference	1167
16.9 include/BasePtr.h File Reference	1168
16.10 include/Camera.h File Reference	1169
16.11 include/CameraBase.h File Reference	1169
16.12 include/CameraDefs.h File Reference	1170
16.13 include/CameraList.h File Reference	1202
16.14 include/CameraPtr.h File Reference	1203
16.15 include/ChunkData.h File Reference	1203
16.16 include/ChunkDataInference.h File Reference	1204
16.17 include/DeviceArrivalEventHandler.h File Reference	1205
16.18 include/DeviceEventHandler.h File Reference	1206
16.19 include/DeviceEventUtility.h File Reference	1207
16.20 include/DeviceRemovalEventHandler.h File Reference	1208
16.21 include/EventHandler.h File Reference	1209
16.22 include/Exception.h File Reference	1209
16.23 include/Image.h File Reference	1210
16.24 include/ImageEventHandler.h File Reference	1211
16.25 include/ImageList.h File Reference	1211
16.26 include/ImageListEventHandler.h File Reference	1212
16.27 include/ImageProcessor.h File Reference	1213
16.28 include/ImagePtr.h File Reference	1214
16.29 include/ImageStatistics.h File Reference	1215
16.30 include/ImageUtility.h File Reference	1216
16.31 include/ImageUtilityCCM.h File Reference	1217
16.32 include/ImageUtilityHeatmap.h File Reference	1218
16.33 include/ImageUtilityPolarization.h File Reference	1219
16.34 include/Interface.h File Reference	1219
16.35 include/Interface/ICameraBase.h File Reference	1220
16.36 include/Interface/ICameraList.h File Reference	1221
16.37 include/Interface/IChunkData.h File Reference	1221
16.38 include/Interface/IDeviceArrivalEventHandler.h File Reference	1222
16.39 include/Interface/IDeviceEventHandler.h File Reference	1223
16.40 include/Interface/IDeviceRemovalEventHandler.h File Reference	1223
16.41 include/Interface/IImage.h File Reference	1224
16.42 include/Interface/IImageEventHandler.h File Reference	1225
16.43 include/Interface/IImageList.h File Reference	1225
16.44 include/Interface/IImageListEventHandler.h File Reference	1226
16.45 include/Interface/IImageProcessor.h File Reference	1227
16.46 include/Interface/IImageStatistics.h File Reference	1228

16.47 include/Interface/IInterface.h File Reference	1228
16.48 include/Interface/IInterfaceArrivalEventHandler.h File Reference	1229
16.49 include/Interface/IInterfaceEventHandler.h File Reference	1230
16.50 include/Interface/IInterfaceList.h File Reference	1231
16.51 include/Interface/IInterfaceRemovalEventHandler.h File Reference	1231
16.52 include/Interface/ILoggingEventHandler.h File Reference	1232
16.53 include/Interface/IStream.h File Reference	1232
16.54 include/Interface/ISystem.h File Reference	1233
16.55 include/Interface/ISystemEventHandler.h File Reference	1234
16.56 include/InterfaceArrivalEventHandler.h File Reference	1235
16.57 include/InterfaceEventHandler.h File Reference	1236
16.58 include/InterfaceList.h File Reference	1237
16.59 include/InterfacePtr.h File Reference	1237
16.60 include/InterfaceRemovalEventHandler.h File Reference	1238
16.61 include/LoggingEventData.h File Reference	1239
16.62 include/LoggingEventDataPtr.h File Reference	1239
16.63 include/LoggingEventHandler.h File Reference	1240
16.64 include/SpinGenApi/Autovector.h File Reference	1241
16.65 include/SpinGenApi/Base.h File Reference	1242
16.66 include/SpinGenApi/BooleanNode.h File Reference	1243
16.67 include/SpinGenApi/CategoryNode.h File Reference	1244
16.68 include/SpinGenApi/ChunkAdapter.h File Reference	1245
16.69 include/SpinGenApi/ChunkAdapterDcam.h File Reference	1246
16.70 include/SpinGenApi/ChunkAdapterGeneric.h File Reference	1247
16.71 include/SpinGenApi/ChunkAdapterGEV.h File Reference	1248
16.72 include/SpinGenApi/ChunkAdapterU3V.h File Reference	1249
16.73 include/SpinGenApi/ChunkPort.h File Reference	1250
16.74 include/SpinGenApi/CommandNode.h File Reference	1250
16.75 include/SpinGenApi/Compatibility.h File Reference	1251
16.75.1 Macro Definition Documentation	1251
16.75.1.1 FMT_I64	1251
16.76 include/SpinGenApi/Container.h File Reference	1252
16.77 include/SpinGenApi/Counter.h File Reference	1252
16.78 include/SpinGenApi/EnumClasses.h File Reference	1253
16.79 include/SpinGenApi/EnumEntryNode.h File Reference	1254
16.80 include/SpinGenApi/EnumNode.h File Reference	1255
16.81 include/SpinGenApi/EnumNodeT.h File Reference	1256
16.82 include/SpinGenApi/EventAdapter.h File Reference	1256
16.83 include/SpinGenApi/EventAdapter1394.h File Reference	1257
16.84 include/SpinGenApi/EventAdapterGeneric.h File Reference	1258
16.85 include/SpinGenApi/EventAdapterGEV.h File Reference	1258
16.86 include/SpinGenApi/EventAdapterU3V.h File Reference	1259

16.87 include/SpinGenApi/EventPort.h File Reference	1260
16.88 include/SpinGenApi/Filestream.h File Reference	1261
16.89 include/SpinGenApi/FloatNode.h File Reference	1262
16.90 include/SpinGenApi/FloatRegNode.h File Reference	1263
16.91 include/SpinGenApi/GCBase.h File Reference	1264
16.92 include/SpinGenApi/GCString.h File Reference	1264
16.92.1 Macro Definition Documentation	1265
16.92.1.1 GCSTRING_NPOS	1265
16.92.2 Function Documentation	1265
16.92.2.1 operator<<()	1266
16.92.2.2 operator>>()	1266
16.93 include/SpinGenApi/GCStringVector.h File Reference	1266
16.94 include/SpinGenApi/GCSynch.h File Reference	1267
16.95 include/SpinGenApi/GCTypes.h File Reference	1268
16.95.1 Macro Definition Documentation	1269
16.95.1.1 __STDC_CONSTANT_MACROS	1269
16.95.1.2 __STDC_LIMIT_MACROS	1269
16.95.1.3 GC_INT32_MAX	1269
16.95.1.4 GC_INT32_MIN	1269
16.95.1.5 GC_INT64_MAX	1269
16.95.1.6 GC_INT64_MIN	1270
16.95.1.7 GC_INT8_MAX	1270
16.95.1.8 GC_INT8_MIN	1270
16.95.1.9 GC_UINT32_MAX	1270
16.95.1.10 GC_UINT64_MAX	1270
16.95.1.11 GC_UINT8_MAX	1270
16.96 include/SpinGenApi/GCUtilities.h File Reference	1271
16.96.1 Macro Definition Documentation	1272
16.96.1.1 __ERR__	1273
16.96.1.2 __LINE_STR__	1273
16.96.1.3 __LOCATION__	1273
16.96.1.4 __OUTPUT_FORMATER__	1273
16.96.1.5 __TODO__	1273
16.96.1.6 __WARN__	1273
16.96.1.7 _TO_STRING	1273
16.96.1.8 EXPAND_TO_STRINGISE	1274
16.96.1.9 GC_COUNTOF	1274
16.96.1.10 GENICAM_DEPRECATED	1274
16.96.1.11 GENICAM_UNUSED	1274
16.96.1.12 USE_TEMP_CACHE_FILE [1/2]	1274
16.96.1.13 USE_TEMP_CACHE_FILE [2/2]	1274
16.97 include/SpinGenApi/IBoolean.h File Reference	1275

16.98 include/SpinGenApi/ICategory.h File Reference	1276
16.99 include/SpinGenApi/IChunkPort.h File Reference	1277
16.99.1 Macro Definition Documentation	1278
16.99.1.1 CHUNK_BASE_ADDRESS_REGISTER	1278
16.99.1.2 CHUNK_BASE_ADDRESS_REGISTER_LEN	1278
16.99.1.3 CHUNK_LENGTH_REGISTER	1278
16.99.1.4 CHUNK_LENGTH_REGISTER_LEN	1278
16.100 include/SpinGenApi/ICommand.h File Reference	1279
16.101 include/SpinGenApi/IDestroy.h File Reference	1280
16.102 include/SpinGenApi/IDeviceInfo.h File Reference	1281
16.103 include/SpinGenApi/IEnumEntry.h File Reference	1282
16.104 include/SpinGenApi/IEnumeration.h File Reference	1283
16.105 include/SpinGenApi/IEnumerationT.h File Reference	1284
16.106 include/SpinGenApi/IFloat.h File Reference	1285
16.107 include/SpinGenApi/IInteger.h File Reference	1287
16.108 include/SpinGenApi/INode.h File Reference	1288
16.109 include/SpinGenApi/INodeMap.h File Reference	1291
16.110 include/SpinGenApi/INodeMapDyn.h File Reference	1293
16.111 include/SpinGenApi/IntegerNode.h File Reference	1294
16.112 include/SpinGenApi/IntRegNode.h File Reference	1295
16.113 include/SpinGenApi/IPort.h File Reference	1296
16.114 include/SpinGenApi/IPortConstruct.h File Reference	1297
16.115 include/SpinGenApi/IPortRecorder.h File Reference	1298
16.116 include/SpinGenApi/IRegister.h File Reference	1299
16.117 include/SpinGenApi/ISelector.h File Reference	1300
16.118 include/SpinGenApi/ISelectorDigit.h File Reference	1301
16.119 include/SpinGenApi/IString.h File Reference	1302
16.120 include/SpinGenApi/IValue.h File Reference	1303
16.121 include/SpinGenApi/Node.h File Reference	1304
16.122 include/SpinGenApi/NodeCallback.h File Reference	1305
16.123 include/SpinGenApi/NodeCallbackImpl.h File Reference	1307
16.124 include/SpinGenApi/NodeMap.h File Reference	1307
16.125 include/SpinGenApi/NodeMapFactory.h File Reference	1308
16.126 include/SpinGenApi/NodeMapRef.h File Reference	1309
16.127 include/SpinGenApi/Persistence.h File Reference	1310
16.128 include/SpinGenApi/Pointer.h File Reference	1311
16.129 include/SpinGenApi/PortImpl.h File Reference	1313
16.130 include/SpinGenApi/PortNode.h File Reference	1314
16.131 include/SpinGenApi/PortRecorder.h File Reference	1315
16.132 include/SpinGenApi/PortReplay.h File Reference	1316
16.133 include/SpinGenApi/PortWriteList.h File Reference	1317
16.134 include/SpinGenApi/Reference.h File Reference	1318

16.135 include/SpinGenApi/RegisterNode.h File Reference	1319
16.136 include/SpinGenApi/RegisterPortImpl.h File Reference	1319
16.137 include/SpinGenApi/SelectorSet.h File Reference	1320
16.138 include/SpinGenApi/SpinnakerGenApi.h File Reference	1320
16.139 include/SpinGenApi/SpinTestCamera.h File Reference	1321
16.140 include/SpinGenApi/StringNode.h File Reference	1321
16.141 include/SpinGenApi/StringRegNode.h File Reference	1322
16.142 include/SpinGenApi/StructPort.h File Reference	1323
16.143 include/SpinGenApi/Synch.h File Reference	1323
16.144 include/SpinGenApi/Types.h File Reference	1324
16.144.1 Macro Definition Documentation	1327
16.144.1.1 _UndefinedRepresentation	1327
16.144.1.2 interface	1327
16.145 include/SpinGenApi/ValueNode.h File Reference	1327
16.146 include/Spinnaker.h File Reference	1328
16.147 include/SpinnakerDefs.h File Reference	1329
16.148 include/SpinnakerDirectShow.h File Reference	1334
16.148.1 Function Documentation	1336
16.148.1.1 BeginAcquisition()	1336
16.148.1.2 BooleanGetValue()	1337
16.148.1.3 BooleanSetValue()	1338
16.148.1.4 CommandExecute()	1338
16.148.1.5 EndAcquisition()	1339
16.148.1.6 EnumerationGetEntry()	1339
16.148.1.7 EnumerationGetNumEntries()	1339
16.148.1.8 EnumerationSetEntry()	1340
16.148.1.9 FloatGetInc()	1340
16.148.1.10 FloatGetIncMode()	1340
16.148.1.11 FloatGetMax()	1341
16.148.1.12 FloatGetMin()	1341
16.148.1.13 FloatGetValue()	1342
16.148.1.14 FloatSetValue()	1342
16.148.1.15 GetCameraInfo()	1342
16.148.1.16 GetSelectedCameraIndex()	1343
16.148.1.17 IntegerGetInc()	1343
16.148.1.18 IntegerGetIncMode()	1344
16.148.1.19 IntegerGetMax()	1344
16.148.1.20 IntegerGetMin()	1344
16.148.1.21 IntegerGetValue()	1345
16.148.1.22 IntegerSetValue()	1345
16.148.1.23 IsStreaming()	1345
16.148.1.24 NodeGetDisplayName()	1346

16.148.1.25 NodeGetType()	1346
16.148.1.26 ModelsAvailable()	1347
16.148.1.27 ModelsImplemented()	1347
16.148.1.28 ModelsReadable()	1347
16.148.1.29 ModelsWritable()	1348
16.148.1.30 NodeMapGetNodeAtIndex()	1348
16.148.1.31 NodeMapGetNumNodes()	1349
16.148.1.32 NodeToString()	1349
16.148.1.33 SetSelectedCameraIndex()	1349
16.148.1.34 StringGetValue()	1350
16.148.1.35 StringSetValue()	1350
16.148.2 Variable Documentation	1351
16.148.2.1 bufferSize	1351
16.148.2.2 entryExist	1351
16.148.2.3 entryIndex	1351
16.148.2.4 enumerationEntryName	1351
16.148.2.5 enumerationName	1351
16.148.2.6 IID_ISpinnakerInterface	1352
16.148.2.7 MAX_LENGTH	1352
16.149 include/SpinnakerPlatform.h File Reference	1352
16.149.1 Macro Definition Documentation	1352
16.149.1.1 SPINNAKER_API	1352
16.149.1.2 SPINNAKER_API_ABSTRACT	1353
16.149.1.3 SPINNAKER_LOCAL	1353
16.150 include/SpinUpdate.h File Reference	1353
16.150.1 Macro Definition Documentation	1354
16.150.1.1 SPINUPDATE_API	1354
16.150.2 Function Documentation	1354
16.150.2.1 GetErrorMessage()	1354
16.150.2.2 SetMessageCallback()	1354
16.150.2.3 SetProgressCallback()	1354
16.150.2.4 UpdateFirmware()	1354
16.150.2.5 UpdateFirmwareConsole()	1354
16.150.2.6 UpdateFirmwareGUI()	1355
16.150.3 Variable Documentation	1355
16.150.3.1 UpdatorMessageCallback	1355
16.150.3.2 UpdatorProgressCallback	1355
16.151 include/SpinVideo.h File Reference	1356
16.152 include/SpinVideoDefs.h File Reference	1357
16.153 include/System.h File Reference	1358
16.153.1 Macro Definition Documentation	1358
16.153.1.1 FLIR_SPINNAKER_VERSION_BUILD	1358

16.153.1.2 FLIR_SPINNAKER_VERSION_MAJOR	1359
16.153.1.3 FLIR_SPINNAKER_VERSION_MINOR	1359
16.153.1.4 FLIR_SPINNAKER_VERSION_TYPE	1359
16.154 include/SystemEventHandler.h File Reference	1359
16.155 include/SystemPtr.h File Reference	1360
16.156 include/TransportLayerDefs.h File Reference	1360
16.157 include/TransportLayerDevice.h File Reference	1362
16.158 include/TransportLayerInterface.h File Reference	1363
16.159 include/TransportLayerStream.h File Reference	1364
16.160 include/TransportLayerSystem.h File Reference	1364
16.161 src/Acquisition/Acquisition.cpp File Reference	1365
16.161.1 Function Documentation	1365
16.161.1.1 AcquireImages()	1365
16.161.1.2 ConfigureGVCPHeartbeat()	1365
16.161.1.3 DisableGVCPHeartbeat()	1366
16.161.1.4 main()	1366
16.161.1.5 PrintDeviceInfo()	1366
16.161.1.6 ResetGVCPHeartbeat()	1366
16.161.1.7 RunSingleCamera()	1366
16.162 src/Acquisition/resource.h File Reference	1368
16.163 src/AcquisitionMultipleCameraRecovery/resource.h File Reference	1368
16.164 src/AcquisitionMultipleCamerasWriteToFile/resource.h File Reference	1368
16.165 src/AcquisitionMultipleThread/resource.h File Reference	1368
16.166 src/ActionCommand/resource.h File Reference	1368
16.167 src/BufferHandling/resource.h File Reference	1368
16.168 src/ChunkData/resource.h File Reference	1368
16.169 src/Compression/resource.h File Reference	1368
16.170 src/CounterAndTimer/resource.h File Reference	1368
16.171 src/DeviceEvents/resource.h File Reference	1368
16.172 src/Enumeration/resource.h File Reference	1368
16.173 src/Enumeration_QuickSpin/resource.h File Reference	1368
16.174 src/EnumerationEvents/resource.h File Reference	1368
16.175 src/ExceptionHandling/resource.h File Reference	1368
16.176 src/Exposure/resource.h File Reference	1368
16.177 src/Exposure_QuickSpin/resource.h File Reference	1368
16.178 src/FileAccess_QuickSpin/resource.h File Reference	1368
16.179 src/GigEVisionPerformance/resource.h File Reference	1368
16.180 src/ImageEvents/resource.h File Reference	1368
16.181 src/ImageFormatControl/resource.h File Reference	1368
16.182 src/ImageFormatControl_QuickSpin/resource.h File Reference	1368
16.183 src/Inference/resource.h File Reference	1368
16.184 src/Logging/resource.h File Reference	1368

16.185 src/LogicBlock/resource.h File Reference	1368
16.186 src/LookupTable/resource.h File Reference	1368
16.187 src/NodeMapCallback/resource.h File Reference	1368
16.188 src/NodeMapInfo/resource.h File Reference	1368
16.189 src/Polarization/resource.h File Reference	1368
16.190 src/SaveToAvi/resource.h File Reference	1368
16.191 src/Sequencer/resource.h File Reference	1368
16.192 src/SerialRxTx/resource.h File Reference	1368
16.193 src/Trigger/resource.h File Reference	1368
16.194 src/Trigger_QuickSpin/resource.h File Reference	1368
16.195 src/Acquisition/stdafx.cpp File Reference	1368
16.196 src/ActionCommand/stdafx.cpp File Reference	1369
16.197 src/BufferHandling/stdafx.cpp File Reference	1369
16.198 src/CounterAndTimer/stdafx.cpp File Reference	1370
16.199 src/DeviceEvents/stdafx.cpp File Reference	1370
16.200 src/Enumeration/stdafx.cpp File Reference	1371
16.201 src/Enumeration_QuickSpin/stdafx.cpp File Reference	1371
16.202 src/ExceptionHandling/stdafx.cpp File Reference	1372
16.203 src/Exposure/stdafx.cpp File Reference	1372
16.204 src/Exposure_QuickSpin/stdafx.cpp File Reference	1373
16.205 src/FileAccess_QuickSpin/stdafx.cpp File Reference	1374
16.206 src/GigEVisionPerformance/stdafx.cpp File Reference	1375
16.207 src/NodeMapInfo/stdafx.cpp File Reference	1375
16.208 src/Sequencer/stdafx.cpp File Reference	1376
16.209 src/SerialRxTx/stdafx.cpp File Reference	1376
16.210 src/Acquisition/stdafx.h File Reference	1377
16.211 src/ActionCommand/stdafx.h File Reference	1378
16.212 src/BufferHandling/stdafx.h File Reference	1379
16.213 src/CounterAndTimer/stdafx.h File Reference	1380
16.214 src/DeviceEvents/stdafx.h File Reference	1381
16.215 src/Enumeration/stdafx.h File Reference	1382
16.216 src/Enumeration_QuickSpin/stdafx.h File Reference	1383
16.217 src/ExceptionHandling/stdafx.h File Reference	1384
16.218 src/Exposure/stdafx.h File Reference	1385
16.219 src/Exposure_QuickSpin/stdafx.h File Reference	1386
16.220 src/FileAccess_QuickSpin/stdafx.h File Reference	1387
16.221 src/GigEVisionPerformance/stdafx.h File Reference	1388
16.222 src/ImageFormatControl/stdafx.h File Reference	1389
16.223 src/ImageFormatControl_QuickSpin/stdafx.h File Reference	1389
16.224 src/NodeMapInfo/stdafx.h File Reference	1390
16.225 src/Polarization/stdafx.h File Reference	1391
16.226 src/Sequencer/stdafx.h File Reference	1391

16.227	src/SerialRxTx/stdafx.h File Reference	1392
16.228	src/Acquisition/targetver.h File Reference	1393
16.229	src/ActionCommand/targetver.h File Reference	1394
16.230	src/BufferHandling/targetver.h File Reference	1395
16.231	src/CounterAndTimer/targetver.h File Reference	1396
16.232	src/DeviceEvents/targetver.h File Reference	1397
16.233	src/Enumeration/targetver.h File Reference	1398
16.234	src/Enumeration_QuickSpin/targetver.h File Reference	1399
16.235	src/ExceptionHandling/targetver.h File Reference	1400
16.236	src/Exposure/targetver.h File Reference	1401
16.237	src/Exposure_QuickSpin/targetver.h File Reference	1402
16.238	src/FileAccess_QuickSpin/targetver.h File Reference	1403
16.239	src/GigEVisionPerformance/targetver.h File Reference	1404
16.240	src/NodeMapInfo/targetver.h File Reference	1405
16.241	src/Sequencer/targetver.h File Reference	1406
16.242	src/SerialRxTx/targetver.h File Reference	1407
16.243	src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference	1407
16.243.1	Function Documentation	1408
16.243.1.1	ConfigureCamera()	1408
16.243.1.2	ConfigureUserSet1()	1408
16.243.1.3	GetDeviceSerial()	1408
16.243.1.4	main()	1408
16.243.1.5	PrintExampleStatistics()	1409
16.243.1.6	RefreshCameraList()	1409
16.243.1.7	ResetCameraUserSetToDefault()	1409
16.243.1.8	SleepyWrapper()	1409
16.243.2	Variable Documentation	1409
16.243.2.1	cameraGrabInfoMap	1409
16.243.2.2	globalCamList	1409
16.244	src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp File Reference	1410
16.244.1	Function Documentation	1410
16.244.1.1	AcquireImagesAndSaveToFile()	1410
16.244.1.2	ConfigureCameras()	1411
16.244.1.3	CreateFiles()	1411
16.244.1.4	main()	1411
16.244.1.5	PrintDeviceInfo()	1411
16.244.1.6	RetrievelImagesFromFiles()	1411
16.244.1.7	RunCameras()	1411
16.244.2	Variable Documentation	1411
16.244.2.1	imageInfos	1412
16.244.2.2	k_numImages	1412

16.244.2.3 kDestinationDirectory	1412
16.245 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference	1412
16.245.1 Function Documentation	1412
16.245.1.1 AcquireImages()	1412
16.245.1.2 ConfigureGVCPHeartbeat()	1413
16.245.1.3 DisableGVCPHeartbeat()	1413
16.245.1.4 main()	1413
16.245.1.5 PrintDeviceInfo()	1413
16.245.1.6 ResetGVCPHeartbeat()	1413
16.245.1.7 RunMultipleCameras()	1413
16.246 src/ActionCommand/ActionCommand.cpp File Reference	1414
16.246.1 Function Documentation	1414
16.246.1.1 AcquireImages()	1414
16.246.1.2 ConfigureActionControl()	1414
16.246.1.3 ConfigureChunkData()	1415
16.246.1.4 ConfigureIEEE1588()	1415
16.246.1.5 ConfigureInterface()	1415
16.246.1.6 ConfigureOtherNodes()	1415
16.246.1.7 ConfigureTrigger()	1415
16.246.1.8 main()	1415
16.246.1.9 PrintDeviceInfo()	1415
16.246.1.10 RunMultipleCameras()	1416
16.246.1.11 SleepyWrapper()	1416
16.247 src/BufferHandling/BufferHandling.cpp File Reference	1416
16.247.1 Macro Definition Documentation	1416
16.247.1.1 k_numLoops	1417
16.247.1.2 numBuffers	1417
16.247.1.3 z_numTriggers	1417
16.247.2 Function Documentation	1417
16.247.2.1 AcquireImages()	1417
16.247.2.2 ConfigureTrigger()	1417
16.247.2.3 GrabNextImageByTrigger()	1417
16.247.2.4 main()	1418
16.247.2.5 PrintDeviceInfo()	1418
16.247.2.6 ResetTrigger()	1418
16.247.2.7 RunSingleCamera()	1418
16.247.2.8 SleepyWrapper()	1418
16.248 src/ChunkData/ChunkData.cpp File Reference	1418
16.248.1 Enumeration Type Documentation	1419
16.248.1.1 chunkDataType	1419
16.248.2 Function Documentation	1419
16.248.2.1 AcquireImages()	1419

16.248.2.2 ConfigureChunkData()	1420
16.248.2.3 DisableChunkData()	1420
16.248.2.4 DisplayChunkData() [1/2]	1420
16.248.2.5 DisplayChunkData() [2/2]	1420
16.248.2.6 main()	1420
16.248.2.7 PrintDeviceInfo()	1420
16.248.2.8 RunSingleCamera()	1420
16.248.3 Variable Documentation	1421
16.248.3.1 chosenChunkData	1421
16.249 src/Compression/Compression.cpp File Reference	1421
16.249.1 Function Documentation	1421
16.249.1.1 AcquireImages()	1422
16.249.1.2 ConfigureGVCPHeartbeat()	1422
16.249.1.3 DisableGVCPHeartbeat()	1422
16.249.1.4 DisableImageChunkData()	1422
16.249.1.5 DisableImageCompression()	1422
16.249.1.6 EnableImageChunkData()	1422
16.249.1.7 EnableImageCompression()	1423
16.249.1.8 main()	1423
16.249.1.9 PrintDeviceInfo()	1423
16.249.1.10 ProcessCompressedImagesFromFile()	1423
16.249.1.11 ResetGVCPHeartbeat()	1423
16.249.1.12 RunSingleCamera()	1423
16.249.2 Variable Documentation	1423
16.249.2.1 enableChunkData	1424
16.250 src/CounterAndTimer/CounterAndTimer.cpp File Reference	1424
16.250.1 Function Documentation	1424
16.250.1.1 AcquireImages()	1424
16.250.1.2 ConfigureDigitalIO()	1424
16.250.1.3 ConfigureExposureandTrigger()	1425
16.250.1.4 main()	1425
16.250.1.5 PrintDeviceInfo()	1425
16.250.1.6 ResetTrigger()	1425
16.250.1.7 RunSingleCamera()	1425
16.250.1.8 SetupCounterAndTimer()	1425
16.251 src/DeviceEvents/DeviceEvents.cpp File Reference	1425
16.251.1 Enumeration Type Documentation	1426
16.251.1.1 eventType	1426
16.251.2 Function Documentation	1426
16.251.2.1 AcquireImages()	1426
16.251.2.2 ConfigureDeviceEvents()	1427
16.251.2.3 InferenceAvailable()	1427

16.251.2.4 main()	1427
16.251.2.5 PrintDeviceInfo()	1427
16.251.2.6 ResetDeviceEvents()	1427
16.251.2.7 RunSingleCamera()	1427
16.251.3 Variable Documentation	1428
16.251.3.1 chosenEvent	1428
16.252 src/Enumeration/Enumeration.cpp File Reference	1428
16.252.1 Function Documentation	1428
16.252.1.1 main()	1428
16.252.1.2 QueryInterface()	1428
16.253 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference	1429
16.253.1 Function Documentation	1429
16.253.1.1 main()	1429
16.253.1.2 QueryInterface()	1429
16.254 src/EnumerationEvents/EnumerationEvents.cpp File Reference	1429
16.254.1 Function Documentation	1430
16.254.1.1 CheckGevEnabled()	1430
16.254.1.2 main()	1430
16.255 src/ExceptionHandling/ExceptionHandling.cpp File Reference	1430
16.255.1 Enumeration Type Documentation	1431
16.255.1.1 exceptionType	1431
16.255.2 Function Documentation	1431
16.255.2.1 causeSpinnakerException()	1431
16.255.2.2 causeStandardException()	1431
16.255.2.3 main()	1431
16.255.3 Variable Documentation	1432
16.255.3.1 chosenException	1432
16.256 src/Exposure/Exposure.cpp File Reference	1432
16.256.1 Function Documentation	1432
16.256.1.1 AcquireImages()	1432
16.256.1.2 ConfigureExposure()	1432
16.256.1.3 main()	1433
16.256.1.4 PrintDeviceInfo()	1433
16.256.1.5 ResetExposure()	1433
16.256.1.6 RunSingleCamera()	1433
16.257 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference	1433
16.257.1 Function Documentation	1434
16.257.1.1 AcquireImages()	1434
16.257.1.2 ConfigureExposure()	1434
16.257.1.3 main()	1434
16.257.1.4 PrintDeviceInfo()	1434
16.257.1.5 ResetExposure()	1434

16.257.1.6 RunSingleCamera()	1434
16.258 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference	1435
16.258.1 Function Documentation	1435
16.258.1.1 AcquireImages()	1435
16.258.1.2 CloseFile()	1436
16.258.1.3 DownloadImage()	1436
16.258.1.4 ExecuteDeleteCommand()	1436
16.258.1.5 ExecuteReadCommand()	1436
16.258.1.6 ExecuteWriteCommand()	1436
16.258.1.7 InitializeSystem()	1436
16.258.1.8 main()	1436
16.258.1.9 OpenFileToRead()	1437
16.258.1.10 OpenFileToWrite()	1437
16.258.1.11 PrintDebugMessage()	1437
16.258.1.12 PrintDeviceInfo()	1437
16.258.1.13 PrintResultMessage()	1437
16.258.1.14 PrintUsage()	1437
16.258.1.15 UploadImage()	1437
16.258.2 Variable Documentation	1438
16.258.2.1 _enableDebug	1438
16.258.2.2 _fileSelector	1438
16.259 src/GigEVisionPerformance/CpuUtil.cpp File Reference	1438
16.260 src/GigEVisionPerformance/CpuUtil.h File Reference	1439
16.261 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference	1440
16.261.1 Function Documentation	1441
16.261.1.1 AcquireImages()	1441
16.261.1.2 EnableManualFramerate()	1441
16.261.1.3 getCameraCategory()	1441
16.261.1.4 main()	1441
16.261.1.5 ParseArguments()	1442
16.261.1.6 PrintAllNodes()	1442
16.261.1.7 PrintCPUUsage()	1442
16.261.1.8 PrintDataStreamInfo()	1442
16.261.1.9 PrintDeviceInfo()	1442
16.261.1.10 PrintUsage()	1442
16.261.1.11 RunSingleCamera()	1442
16.261.1.12 SetFrameRate()	1443
16.261.2 Variable Documentation	1443
16.261.2.1 argBayerRG	1443
16.261.2.2 argDuration	1443
16.261.2.3 argMaxFrames	1443
16.261.2.4 argNumImages	1443

16.261.2.5 argPacketDelay	1443
16.261.2.6 argPacketSize	1443
16.261.2.7 argPrintUsage	1444
16.261.2.8 argRelease	1444
16.261.2.9 argUserSetFrames	1444
16.261.2.10 cpuUsageInfo	1444
16.261.2.11 IsRelease	1444
16.261.2.12 NumImagesToGrab	1444
16.261.2.13 PacketDelayToSet	1444
16.261.2.14 PacketSizeToSet	1444
16.261.2.15 PixelFormatToSet	1445
16.261.2.16 TestDuration	1445
16.261.2.17 UseDuration	1445
16.261.2.18 UseMaxFramerate	1445
16.261.2.19 UserSetFramerate	1445
16.262 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference	1445
16.263 src/ImageEvents/ImageEvents.cpp File Reference	1446
16.263.1 Function Documentation	1446
16.263.1.1 AcquireImages()	1446
16.263.1.2 ConfigureImageEvents()	1446
16.263.1.3 main()	1447
16.263.1.4 PrintDeviceInfo()	1447
16.263.1.5 ResetImageEvents()	1447
16.263.1.6 RunSingleCamera()	1447
16.263.1.7 SleepyWrapper()	1447
16.263.1.8 WaitForImages()	1447
16.264 src/ImageFormatControl/ImageFormatControl.cpp File Reference	1448
16.264.1 Function Documentation	1448
16.264.1.1 AcquireImages()	1448
16.264.1.2 ConfigureCustomImageSettings()	1448
16.264.1.3 main()	1448
16.264.1.4 PrintDeviceInfo()	1449
16.264.1.5 RunSingleCamera()	1449
16.265 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference	1449
16.265.1 Function Documentation	1449
16.265.1.1 AcquireImages()	1449
16.265.1.2 ConfigureCustomImageSettings()	1450
16.265.1.3 main()	1450
16.265.1.4 PrintDeviceInfo()	1450
16.265.1.5 RunSingleCamera()	1450
16.266 src/Inference/Inference.cpp File Reference	1450
16.266.1 Enumeration Type Documentation	1451

16.266.1.1 FileUploadPersistence	1451
16.266.1.2 InferenceNetworkType	1452
16.266.2 Function Documentation	1452
16.266.2.1 AcquireImages()	1452
16.266.2.2 CameraCloseFile()	1452
16.266.2.3 CameraDeleteFile()	1452
16.266.2.4 CameraOpenFile()	1452
16.266.2.5 CameraWriteToFile()	1453
16.266.2.6 ConfigureChunkData()	1453
16.266.2.7 ConfigureInference()	1453
16.266.2.8 ConfigureTestPattern()	1453
16.266.2.9 ConfigureTrigger()	1453
16.266.2.10 DeleteFileOnCamera()	1453
16.266.2.11 DisableChunkData()	1454
16.266.2.12 DisableTrigger()	1454
16.266.2.13 DisplayChunkData()	1454
16.266.2.14 labelClassification()	1454
16.266.2.15 labelDetection()	1454
16.266.2.16 LoadFileIntoMemory()	1454
16.266.2.17 main()	1455
16.266.2.18 PrintDeviceInfo()	1455
16.266.2.19 RunSingleCamera()	1455
16.266.2.20 SetChunkEnable()	1455
16.266.2.21 UploadFileToCamera()	1455
16.266.3 Variable Documentation	1455
16.266.3.1 arrayLabelClassification	1455
16.266.3.2 arrayLabelDetection	1456
16.266.3.3 chosenFileUploadPersistence	1456
16.266.3.4 chosenInferenceNetworkType	1456
16.266.3.5 injectedImageFilePath	1456
16.266.3.6 injectedImageHeight	1456
16.266.3.7 injectedImageWidth	1456
16.266.3.8 networkFilePath	1457
16.267 src/Logging/Logging.cpp File Reference	1457
16.267.1 Function Documentation	1457
16.267.1.1 main()	1457
16.267.2 Variable Documentation	1457
16.267.2.1 k_LoggingLevel	1458
16.268 src/LogicBlock/LogicBlock.cpp File Reference	1458
16.268.1 Function Documentation	1458
16.268.1.1 AcquireImages()	1458
16.268.1.2 ConfigureLogicBlock()	1458

16.268.1.3 ConfigureTrigger()	1459
16.268.1.4 GrabTwoImages()	1459
16.268.1.5 main()	1459
16.268.1.6 PrintDeviceInfo()	1459
16.268.1.7 ResetExposure()	1459
16.268.1.8 ResetTrigger()	1459
16.268.1.9 RunSingleCamera()	1459
16.269 src/LookupTable/LookupTable.cpp File Reference	1460
16.269.1 Function Documentation	1460
16.269.1.1 AcquireImages()	1460
16.269.1.2 ConfigureLookupTables()	1460
16.269.1.3 main()	1460
16.269.1.4 PrintDeviceInfo()	1461
16.269.1.5 PrintRetrieveNodeFailure()	1461
16.269.1.6 ResetLookupTables()	1461
16.269.1.7 RunSingleCamera()	1461
16.270 src/NodeMapCallback/NodeMapCallback.cpp File Reference	1461
16.270.1 Function Documentation	1462
16.270.1.1 ChangeHeightAndGain()	1462
16.270.1.2 ConfigureCallbacks()	1462
16.270.1.3 main()	1462
16.270.1.4 OnGainNodeUpdate()	1462
16.270.1.5 OnHeightNodeUpdate()	1462
16.270.1.6 PrintDeviceInfo()	1462
16.270.1.7 ResetCallbacks()	1463
16.270.1.8 RunSingleCamera()	1463
16.271 src/NodeMapInfo/NodeMapInfo.cpp File Reference	1463
16.271.1 Enumeration Type Documentation	1464
16.271.1.1 readType	1464
16.271.2 Function Documentation	1464
16.271.2.1 Indent()	1464
16.271.2.2 main()	1464
16.271.2.3 PrintBooleanNode()	1464
16.271.2.4 PrintCategoryNodeAndAllFeatures()	1465
16.271.2.5 PrintCommandNode()	1465
16.271.2.6 PrintEnumerationNodeAndCurrentEntry()	1465
16.271.2.7 PrintEnumerationSelector()	1465
16.271.2.8 PrintFloatNode()	1465
16.271.2.9 PrintIntegerNode()	1465
16.271.2.10 PrintNode()	1466
16.271.2.11 PrintStringNode()	1466
16.271.2.12 PrintValueNode()	1466

16.271.2.13 RunSingleCamera()	1466
16.271.3 Variable Documentation	1466
16.271.3.1 chosenRead	1466
16.271.3.2 maxChars	1466
16.272 src/Polarization/Polarization.cpp File Reference	1467
16.272.1 Function Documentation	1467
16.272.1.1 AcquireImages()	1467
16.272.1.2 ConfigureStream()	1468
16.272.1.3 CreateAndSaveAolpDolpImages()	1468
16.272.1.4 CreateAndSaveGlareReducedImage()	1468
16.272.1.5 CreateAndSaveStokesImages()	1468
16.272.1.6 CreateHeatmapImages()	1468
16.272.1.7 CreateNormalizedImage()	1468
16.272.1.8 ExtractAndSavePolarQuadImages()	1469
16.272.1.9 GetQuadFileNameAppendage()	1469
16.272.1.10 main()	1469
16.272.1.11 PrintDeviceInfo()	1469
16.272.1.12 RunSingleCamera()	1469
16.272.1.13 SaveImage()	1469
16.272.2 Variable Documentation	1469
16.272.2.1 isPixelFormatColor	1470
16.273 src/SaveToAvi/SaveToAvi.cpp File Reference	1470
16.273.1 Enumeration Type Documentation	1470
16.273.1.1 videoType	1470
16.273.2 Function Documentation	1471
16.273.2.1 AcquireImages()	1471
16.273.2.2 main()	1471
16.273.2.3 PrintDeviceInfo()	1471
16.273.2.4 RunSingleCamera()	1471
16.273.2.5 SaveVectorToVideo()	1471
16.273.3 Variable Documentation	1471
16.273.3.1 chosenVideoType	1472
16.274 src/Sequencer/Sequencer.cpp File Reference	1472
16.274.1 Function Documentation	1472
16.274.1.1 AcquireImages()	1472
16.274.1.2 ConfigureSequencerPartOne()	1472
16.274.1.3 ConfigureSequencerPartTwo()	1473
16.274.1.4 main()	1473
16.274.1.5 PrintDeviceInfo()	1473
16.274.1.6 PrintRetrieveNodeFailure()	1473
16.274.1.7 ResetSequencer()	1473
16.274.1.8 RunSingleCamera()	1473

16.274.1.9 SetSingleState()	1474
16.275 src/SerialRxTx/SerialRxTx.cpp File Reference	1474
16.275.1 Macro Definition Documentation	1474
16.275.1.1 COM_PORT_COUNT_MAX	1475
16.275.1.2 DATA_BITS	1475
16.275.1.3 MILLISECOND	1475
16.275.1.4 SERIAL_PORT_BAUD_RATE	1475
16.275.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND	1475
16.275.1.6 SERIAL_PORT_DELAY	1475
16.275.1.7 SERIAL_PORT_PARITY_BITS	1475
16.275.1.8 SERIAL_PORT_STOP_BITS	1475
16.275.1.9 TWO_SECOND_DELAY	1476
16.275.2 Function Documentation	1476
16.275.2.1 Cleanup()	1476
16.275.2.2 ConfigureDevice()	1476
16.275.2.3 main()	1476
16.275.2.4 PrintDeviceInfo()	1476
16.275.2.5 RunSingleCamera()	1476
16.275.2.6 SerialRx()	1477
16.275.2.7 SerialTx()	1477
16.276 src/Trigger/Trigger.cpp File Reference	1477
16.276.1 Enumeration Type Documentation	1478
16.276.1.1 triggerType	1478
16.276.2 Function Documentation	1479
16.276.2.1 AcquireImages()	1479
16.276.2.2 ConfigureTrigger()	1479
16.276.2.3 GrabNextImageByTrigger()	1479
16.276.2.4 main()	1479
16.276.2.5 PrintDeviceInfo()	1479
16.276.2.6 ResetTrigger()	1480
16.276.2.7 RunSingleCamera()	1480
16.276.3 Variable Documentation	1480
16.276.3.1 chosenTrigger	1480
16.277 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference	1480
16.277.1 Enumeration Type Documentation	1481
16.277.1.1 triggerType	1481
16.277.2 Function Documentation	1481
16.277.2.1 AcquireImages()	1481
16.277.2.2 ConfigureTrigger()	1481
16.277.2.3 GrabNextImageByTrigger()	1481
16.277.2.4 main()	1482
16.277.2.5 PrintDeviceInfo()	1482

16.277.2.6 ResetTrigger()	1482
16.277.2.7 RunSingleCamera()	1482
16.277.3 Variable Documentation	1482
16.277.3.1 chosenTrigger	1482
17 Example Documentation	1483
17.1 Acquisition.cpp	1483
17.2 AcquisitionMultipleCameraRecovery.cpp	1483
17.3 AcquisitionMultipleCamerasWriteToFiles.cpp	1483
17.4 AcquisitionMultipleThread.cpp	1484
17.5 ActionCommand.cpp	1484
17.6 BufferHandling.cpp	1484
17.7 Compression.cpp	1484
17.8 CounterAndTimer.cpp	1484
17.9 DeviceEvents.cpp	1485
17.10 EnumerationEvents.cpp	1485
17.11 ExceptionHandling.cpp	1485
17.12 Exposure.cpp	1485
17.13 FileAccess_Quickspin.cpp	1486
17.14 GigEVisionPerformance.cpp	1486
17.15 ImageEvents.cpp	1486
17.16 ImageFormatControl.cpp	1486
17.17 ImageFormatControl_QuickSpin.cpp	1486
17.18 Inference.cpp	1487
17.19 Logging.cpp	1487
17.20 LogicBlock.cpp	1487
17.21 LookupTable.cpp	1488
17.22 NodeMapCallback.cpp	1488
17.23 NodeMapInfo.cpp	1488
17.24 Polarization.cpp	1488
17.25 SaveToAvi.cpp	1489
17.26 Sequencer.cpp	1489
17.27 Trigger.cpp	1489
17.28 Trigger_QuickSpin.cpp	1489
Index	1491

Chapter 1

Getting Started

The [Spinnaker](#) application programming interface (API) is used to interface with FLIR's USB3 Vision and GigE Vision cameras.

- [Benefits of Spinnaker](#)
- [Software Licensing Information](#)
- [Software Maintenance Policy](#)
- [FlyCapture2 Feature Comparison with Spinnaker](#)
- [Programmer's Guide](#)
- [SpinViewGuide](#)
- [Working with GenICam GenTL Devices](#)
- [Myricom](#)

Chapter 2

Programmer's Guide

Please see (<http://softwareservices.flir.com/Spinnaker/latest/page2.html>) for the latest version of this document

Chapter 3

Benefits of Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/index.html>) for the latest version of this document

Chapter 4

FlyCapture2 Feature Comparison with Spinnaker

Please see (<http://softwareservices.flir.com/Spinnaker/latest/page3.html>) for the latest version of this document

Chapter 5

Working with GenICam GenTL Devices

5.1 GenTL Overview

FLIR GenTL Producer is a software driver that implements the GenICam™ GenTL 1.5 standard (<https://www.emva.org/>). It allows users to enumerate, communicate and stream from FLIR GigE Vision and USB3 Vision devices in a generic way independent from the underlying transport technology. This allows third-party software such as MATLAB (<https://www.mathworks.com>) and other software libraries to work with FLIR devices in a transport layer agnostic way. These applications are referred to as "GenTL Consumers," which directly use one or more GenTL Producers.

NOTE: Consumer applications must be aware of differences in device capabilities and be prepared to handle specific device models differently.

5.2 Installation

In order to use a FLIR GenTL producer, it needs to be properly registered and installed on the system. **The FLIR Producer comes packaged with the full Spinnaker SDK installer as of 2.x or newer.**

The GenTL Producer is provided as a platform dependent, dynamic loadable library file with the `.cti` ("Common Transport Interface") extension.

The Spinnaker SDK installer stores the folder paths for 32-bit and 64-bit GenTL Producers (`.cti` files) in environment variables named `GENICAM_GENTL32_PATH` and `GENICAM_GENTL64_PATH`, respectively. If there are multiple GenTL Producers installed on the system, path entries must be separated by `;` on Windows and `:` on UNIX-like systems.

NOTE: A 32bit GenTL consumer application will require a 32-bit GenTL producer and a 64-bit application will require a 64-bit producer library.

5.3 Troubleshooting

5.3.1 Enable FLIR GenTL Logging

FLIR GenTL Logging can be enabled if a configuration file with the name "log4cpp.gentl.property" resides in the path of where the consumer application executes from. For MATLAB, this is where the working directory is set and may default to the "Downloads" folder on Windows.

Sample log4cpp.gentl.property configuration file:

```
# FLIR GenTL Property Configuration file
log4cpp.rootCategory=ERROR, rootAppender
log4cpp.category.GenTLCategory=ERROR, GenTLCategory

log4cpp.appender.rootAppender=ConsoleAppender
log4cpp.appender.rootAppender.layout=PatternLayout
log4cpp.appender.rootAppender.layout.ConversionPattern=[%p] %d [%t] %m%n

log4cpp.appender.GenTLCategory=RollingFileAppender
log4cpp.appender.GenTLCategory.fileName=$(ALLUSERSPROFILE)\Spinnaker\Logs\GenTL.log
log4cpp.appender.GenTLCategory.append=true
log4cpp.appender.GenTLCategory.maxFileSize=1000000
log4cpp.appender.GenTLCategory.maxBackupIndex=5
log4cpp.appender.GenTLCategory.layout=PatternLayout
log4cpp.appender.GenTLCategory.layout.ConversionPattern=[%p] %d [%t] %m%n
```

5.3.2 USB3 Device Image Tearing

Image tearing could occur with certain USB3 host controllers when streaming with a GenTL producer. To work around the issue, make sure the size of each buffer announced to the FLIR GenTL producer is 1024 bytes aligned. The size of each buffer should be $(\text{bufferSize} + 1024 - 1) / 1024 * 1024$ where 1024 is the USB3 packet transfer size.

For more information about image tearing causes and solutions, please refer to: <https://www.flir.com/support-center/iis/machine-vision/application-note/image-tearing-causes-and-solutions/>

Chapter 6

Software Licensing Information

Table 6.1 License table

Component	License
Spinnaker	Copyright (c) 2001-2020 FLIR Systems, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
GenICam	GenICam License
AdapterManager	The Code Project Open License (CPO-OL)
Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView	WP:CC-BY-SA License
Work with Bitmaps Faster in C#	The Code Project Open License (CPO-OL) 1.02
FreeImage	FreeImage public license
Boost	Boost Software License
Libusb	LGPLv2.1 License
Libraw1394	LGPLv2.0 License
FFMPEG	LGPLv2.1 License
log4Net	Apache license 2.0
log4Cpp	LGPL License

The licenses mentioned above can also be found in the [Spinnaker](#) installed license folder.

Chapter 7

Software Maintenance Policy

7.1 GenTL Overview

This document outlines the FLIR maintenance policy for [Spinnaker](#) Software Development Kit (SDK). FLIR regularly provides SDK updates that may contain support for new or updated features, enhancements, updated drivers, updated examples, bug fixes or documentation updates. Updates may also address changes with introducing and/or deprecating language runtimes, operating systems and dependencies.

We recommend users to stay up-to-date with SDK releases to keep up with the latest features, bug fixes and performance improvements. Continued use of an unsupported SDK version is not recommended and is done at the user's discretion.

[Spinnaker](#) SDK releases are published through our website and can be found here: <https://www.flir.ca/products/spinnaker-sdk/>

7.2 Platform Support Policy

7.2.1 Windows Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest available version of Windows x86/x64. The latest three versions of Visual Studio compiler toolchain are supported on Windows. Only the latest compiler toolchain on the latest available version of Windows are being actively tested.

7.2.2 Linux Desktop Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest two LTS versions of Ubuntu x86/x64. Only the latest x64 LTS version of Ubuntu is being actively tested.

7.2.3 Linux Embedded Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against the latest supported LTS version of Ubuntu ARMHF/ARM64 for a specific board. Only the latest LTS Ubuntu version on an ARM64 board is being actively tested. Contact sales if you need support for a specific embedded board.

7.2.4 MacOS Support

FLIR will continue to maintain, fix and build the last two major versions of [Spinnaker](#) SDK against MacOS Mojave (10.14). Contact sales if you need newer MacOS support.

7.3 Versioning Policy

[Spinnaker](#) SDK releases use a modified semantic versioning scheme and is indicated by four sets of numbers separated by periods:

MAJOR.MINOR.0.PATCH

- MAJOR: Version change that can include incompatible API changes
- MINOR: Version change that adds functionality in a backwards-compatible manner
- PATCH: Version change with backwards-compatible fixes

Reference: <https://www.flir.com/support-center/iis/machine-vision/knowledge-base/flir-mac>

Chapter 8

Module Index

8.1 Modules

Here is a list of all modules:

Spinnaker Classes	41
AVI Recorder Class	42
BasePtr Class	43
Camera Class	44
Camera Base Class	45
CameraDefs Class	46
Camera List Class	47
CameraPtr Class	48
ChunkData Class	49
Chunk Data Inference Class	50
Spinnaker EventHandler Classes	51
DeviceArrivalEventHandler Class	52
DeviceEventHandler Class	53
DeviceRemovalEventHandler Class	54
EventHandler Class	55
Exception Class	56
Image Class	57
ImageEventHandler Class	58
ImageListEventHandler Class	59
Image Processor Class	60
ImagePtr Class	61
ImageStatistics Class	62
Image Utility Class	63
Image Utility CCM Class	64
Image Utility Heatmap Class	65
Image Utility Polarization Class	66
Interface Class	67
InterfaceArrivalEventHandler Class	68
InterfaceEventHandler Class	69
InterfaceList Class	70
InterfacePtr Class	71
InterfaceRemovalEventHandler Class	72
Logging EventHandler Class	73
LoggingEventDataPtr Class	74
LoggingEventHandler Class	75

Spinnaker Headers	76
Spinnaker.h	77
Spinnaker Definitions	78
SpinnakerDirectShow.h	79
Spinnaker Platform	80
Spinnaker Video Class	81
Spinnaker Video Definitions	82
System Class	83
SystemEventHandler Class	84
SystemPtr Class	85
Spinnaker QuickSpin Classes	86
TransportLayerDefs Class	87
TransportLayerDevice Class	88
TransportLayerInterface Class	89
TransportLayerStream Class	90
TransportLayerSystem Class	91
Camera Base Interface Class	92
ICChunkData Class	93
IImage Class	94
Image List Class	95
IImageStatistics Class	96
IInterface Class	97
IInterfaceList Class	98
ISystem Class	99
AutoVector Class	101
Spinnaker GenApi Interfaces	102
IBase Interface	103
BooleanNode Class	104
CategoryNode Class	105
ChunkAdapter Class	106
ChunkAdapterDcam Class	107
ChunkAdapterGeneric Class	108
ChunkAdapterGEV Class	109
ChunkPort Class	110
CommandNode Class	111
Container Class	112
Counter Class	113
EnumClasses Class	114
EnumEntryNode Class	115
EnumNode Class	116
EnumNodeT Class	117
EventAdapter Class	118
EventAdapter1394 Class	119
EventAdapterGeneric Class	120
EventAdapterGEV Class	121
EventAdapterU3V Class	122
EventPort Class	123
Filestream Class	124
FloatNode Class	125
FloatRegNode Class	126
GCString Class	127
GCSynch Class	128
GCTypes Class	129
Spinnaker GenApi Utilities	130
GCUtilities Utility	131
IBoolean Interface	132
ICategory Interfaces	133
ICChunkPort Interface	134

ICommand Interface	135
IDestroy Interface	136
IDeviceInfo Interface	137
IEnumEntry Interface	138
IEnumeration Interface	139
IEnumerationT Interface	140
IFloat Interface	141
IInteger Interface	142
INode Interface	143
INodeMap Interface	147
INodeMapDyn Interface	148
IntegerNode Class	149
IntRegNode Class	150
IPort Interface	151
IPortConstruct Interface	152
IPortRecorder Interface	153
IRegister Interfaces	154
ISelector Interface	155
ISelectorDigit Interface	156
IString Class	157
IValue Class	158
Node Class	159
NodeCallback Class	160
NodeMap Class	161
NodeMapFactory Class	162
NodeMapRef Class	163
Persistence Class	164
Pointer Class	165
Spinnaker GenApi Classes	100
PortImpl Class	171
PortNode Class	172
PortRecorder Class	173
PortReplay Class	174
PortWriteList Class	175
Reference Interfaces	176
RegisterNode Class	177
RegisterPortImpl Class	178
SelectorSet Class	179
SpinTestCamera Class	180
StringNode Class	181
StringRegNode Class	182
StructPort Class	183
Synch Class	184
Spinnaker GenApi Enums	185
Types Enums	186
ValueNode Class	187
ChunkAdapterU3V Class	188

Chapter 9

Namespace Index

9.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Conversion	189
CpuUtil	190
PerformanceCounter	190
SecondsCounter	191
Spinnaker	192
Spinnaker::GenApi	338
Spinnaker::GenICam	395
Spinnaker::Video	400

Chapter 10

Hierarchical Index

10.1 Class Hierarchy

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

ActionCommandResult	401
AttachStatistics_t	402
AutoLock	403
AutoLock	403
AVIOption	404
BasePtr< T, B >	405
BasePtr< Camera, ICameraBase >	405
CameraPtr	588
BasePtr< IImage >	405
ImagePtr	940
BasePtr< IInterface >	405
InterfacePtr	1004
BasePtr< ISystem >	405
SystemPtr	1116
BasePtr< LoggingEventData >	405
LoggingEventDataPtr	1028
basic_istream	
IDevFileStreamBase< CharType, Traits >	828
basic_ostream	
ODevFileStreamBase< CharType, Traits >	1059
basic_streambuf	
IDevFileStreamBuf< CharType, Traits >	830
ODevFileStreamBuf< CharType, Traits >	1061
BMPOption	411
CChunkAdapter	592
CChunkAdapterDcam	596
CChunkAdapterGeneric	598
CChunkAdapterGEV	601
CChunkAdapterU3V	603
CCMSettings	610
CDataStruct	
CTestPortStruct< CDataStruct >	708
CEventAdapter	617

CEventAdapter1394	619
CEventAdapterGeneric	621
CEventAdapterGEV	624
CEventAdapterU3V	626
CGeneric_XMLLoaderParams	638
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams >	677
CNodeMapRef	674
CGlobalLock	639
CGlobalLockUnlocker	641
CLock	657
CLockEx	661
CLock	658
CLockEx	662
CNodeCallback	663
Function_NodeCallback< Function >	774
Member_NodeCallback< Client, Member >	1034
CNodeMapFactory	666
CompressedImageInfo	686
Counter	688
CPointer< T, B >	690
CPointer< IFloat, IBase >	690
CFloatPtr	636
CPointer< INode, IBase >	690
CpuUsageInfo	701
DCAM_CHECKSUM	711
DCAM_CHUNK_TRAILER	712
DeviceEventExposureEndData	715
DeviceEventInferenceData	721
DeviceEventUtility	722
double_autovector_t	725
EAccessModeClass	728
ECachingModeClass	729
EDisplayNotationClass	730
EEndianessClass	731
EGenApiSchemaVersionClass	733
EInputDirectionClass	734
ENameSpaceClass	735
ERepresentationClass	744
ESignClass	745
ESlopeClass	746
EStandardNameSpaceClass	747
EventHandler	749
IDeviceArrivalEventHandler	833
DeviceArrivalEventHandler	713
IInterfaceEventHandler	875
InterfaceEventHandler	992
InterfaceEventHandlerImpl	995
IDeviceEventHandler	835
DeviceEventHandler	716
DeviceEventHandlerImpl	719
IDeviceRemovalEventHandler	837
DeviceRemovalEventHandler	723
IInterfaceEventHandler	875
IImageEventHandler	852
ImageEventHandler	913
ImageEventHandlerImpl	915

IImageListEventHandler	858
ImageListEventHandler	926
IInterfaceArrivalEventHandler	873
InterfaceArrivalEventHandler	989
ISystemEventHandler	1016
SystemEventHandler	1109
SystemEventHandlerImpl	1112
IInterfaceRemovalEventHandler	881
InterfaceRemovalEventHandler	1006
ISystemEventHandler	1016
ILoggingEventHandler	883
LoggingEventHandler	1030
LoggingEventHandlerImpl	1033
EVisibilityClass	752
exception	
Exception	754
EYesNoClass	759
FileProtocolAdapter	760
gcstring	776
GrabInfo	785
GVCP_CHUNK_TRAILER	786
GVCP_EVENT_ITEM	787
GVCP_EVENT_ITEM_BASIC	788
GVCP_EVENT_ITEM_EXTENDED_ID	789
GVCP_EVENT_REQUEST	791
GVCP_EVENT_REQUEST_EXTENDED_ID	792
GVCP_EVENTDATA_REQUEST	793
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	794
GVCP_REQUEST_HEADER	795
H264Option	796
IBoolean	
BooleanNode	412
ICameraBase	798
CameraBase	565
Camera	415
ICameraList	807
CameraList	581
ICategory	
CategoryNode	590
IChunkData	811
ChunkData	643
IChunkPort	
PortNode	1066
PortReplay	1076
PortRecorder	1072
ICommand	
CommandNode	683
IDataStream	821
IDeviceInfo	
NodeMap	1049
SpinTestCamera	1088
IEnumEntry	
EnumEntryNode	736
IEnumeration	
EnumNode	739
CEnumerationTRef< EnumT >	612

IEnumerationT	
CEnumerationTRef< EnumT >	612
IFloat	
FloatNode	764
FloatRegNode	771
IImage	839
Image	885
IImageList	854
ImageList	920
IImageProcessor	860
ImageProcessor	929
IImageStatistics	864
ImageStatistics	943
IInteger	
IntegerNode	978
IntRegNode	1008
IInterface	868
Interface	984
IInterfaceList	878
InterfaceList	999
ImageInfo	918
ImageUtility	950
ImageUtilityCCM	954
ImageUtilityHeatmap	956
ImageUtilityPolarization	959
InferenceBoundingBox	968
InferenceBoundingBoxResult	969
InferenceBoxCircle	972
InferenceBoxRect	973
InferenceBoxRotatedRect	974
INode	
Node	1038
CSelectorSet	705
PortNode	1066
ValueNode	1161
BooleanNode	412
CategoryNode	590
CommandNode	683
EnumEntryNode	736
EnumNode	739
FloatNode	764
IntegerNode	978
RegisterNode	1080
FloatRegNode	771
IntRegNode	1008
StringRegNode	1097
StringNode	1093
StringRegNode	1097
INodeMap	
NodeMap	1049
int64_autovector_t	975
IPersistScript	
CFeatureBag	633
IPortConstruct	
CChunkPort	605
CEventPort	628

CPortImpl	694
CRegisterPortImpl	701
CTestPortStruct< CDataStruct >	708
PortNode	1066
IPortRecorder	
PortNode	1066
PortRecorder	1072
IPortReplay	
CPortImpl	694
PortReplay	1076
IPortWriteList	
CPortWriteList	698
IRegister	
RegisterNode	1080
IString	
StringNode	1093
ISystem	1011
System	1100
IValue	
ValueNode	1161
JPEGOption	1018
JPG2Option	1020
LibraryVersion	1021
LockableObject< Object >::Lock	1022
LockableObject< Object >	1023
LoggingEventData	1025
MJPGOption	1036
CNodeMapFactory::NodeStatistics_t	1058
PGMOption	1063
PNGOption	1064
PPMOption	1079
SingleChunkData_t	1085
SingleChunkDataStr_t	1086
SIOption	1087
SpinVideo	1088
TCameraParams	
CNodeMapRefT< TCameraParams >	677
TIFFOption	1118
TransportLayerDevice	1119
TransportLayerInterface	1130
TransportLayerStream	1141
TransportLayerSystem	1150
U3V_CHUNK_TRAILER	1157
U3V_COMMAND_HEADER	1158
U3V_EVENT_DATA	1159
U3V_EVENT_MESSAGE	1160
Version_t	1164

Chapter 11

Class Index

11.1 Class List

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

ActionCommandResult	
Action Command Result	401
AttachStatistics_t	
Delivers information about the attached chunks and nodes	402
AutoLock	403
AutoLock	403
AVIOption	
Options for saving AVI files	404
BasePtr< T, B >	
The base class of the SystemPtr , CameraPtr , InterfacePtr , ImagePtr and LoggingEventDataPtr objects	405
BMPOption	
Options for saving Bitmap image	411
BooleanNode	
Interface for string properties	412
Camera	
The camera object class	415
CameraBase	
The base class for the camera object	565
CameraList	
Used to hold a list of camera objects	581
CameraPtr	
A reference tracked pointer to a camera object	588
CategoryNode	
Interface for string properties	590
CChunkAdapter	
Connects a chunked buffer to a node map	592
CChunkAdapterDcam	
Connects a chunked DCAM buffer to a node map	596
CChunkAdapterGeneric	598
CChunkAdapterGEV	
Connects a chunked DCAM buffer to a node map	601
CChunkAdapterU3V	
Connects a chunked U3V buffer to a node map	603
CChunkPort	
Port attachable to a chunk in a buffer	605

CCMSettings	610
CEnumerationTRef< EnumT >	
Interface for string properties	612
CEventAdapter	
Delivers Events to ports	617
CEventAdapter1394	
Distribute the events to the node map	619
CEventAdapterGeneric	
Connects a generic event to a node map	621
CEventAdapterGEV	
Connects a GigE Event to a node map	624
CEventAdapterU3V	
Connects a U3V Event to a node map	626
CEventPort	
Port attachable to an event	628
CFeatureBag	
Bag holding streamable features of a nodetree	633
CFloatPtr	
SmartPointer for IFloat interface pointer	636
CGeneric_XMLLoaderParams	
Empty base class used by class CNodeMapRef as generic template argument	638
CGlobalLock	
Named global lock which can be used over process boundaries	639
CGlobalLockUnlocker	
Unlocks the global lock object on destruction	641
ChunkData	
The chunk data which contains additional information about an image	643
CLock	
A lock class	657
CLock	
A lock class	658
CLockEx	
This class is for testing purposes only	661
CLockEx	
This class is for testing purposes only	662
CNodeCallback	
Callback body instance for INode pointers	663
CNodeMapFactory	
The node map factory is used for creating node maps from camera description files	666
CNodeMapRef	
SmartPointer for NodeMaps with create function	674
CNodeMapRefT< TCameraParams >	
SmartPointer template for NodeMaps with create function	677
CommandNode	
Interface for string properties	683
CompressedImageInfo	686
Counter	
Definition of a simple Counter class	688
CPointer< T, B >	
Encapsulates a GenApi pointer dealing with the dynamic_cast automatically	690
CPortImpl	
Standard implementation for a port	694
CPortWriteList	
Container holding a list of port write commands	698
CpuUsageInfo	701
CRegisterPortImpl	
Standard implementation for a port using a register based transport layer	701

CSelectorSet	
The set of selectors selecting a given node	705
CTestPortStruct< CDataStruct >	
Implements a register spaces based on a C++ struct	708
DCAM_CHECKSUM	711
DCAM_CHUNK_TRAILER	712
DeviceArrivalEventHandler	
An event handler for capturing the device arrival event	713
DeviceEventExposureEndData	
Data Fields for Device Event payload for EventExposureEnd	715
DeviceEventHandler	
A handler to device events	716
DeviceEventHandlerImpl	719
DeviceEventInferenceData	
Data fields for Device Event payload for EventInference	721
DeviceEventUtility	722
DeviceRemovalEventHandler	
An event handler for capturing the device removal event	723
double_autovector_t	
Vector of doubles with reference counting	725
EAccessModeClass	
Holds conversion methods for the access mode enumeration	728
ECachingModeClass	
Holds conversion methods for the caching mode enumeration	729
EDisplayNotationClass	
Holds conversion methods for the notation type of floats	730
EEndianessClass	
Holds conversion methods for the endianess enumeration	731
EGenApiSchemaVersionClass	
Helper class converting EGenApiSchemaVersion from and to string	733
EInputDirectionClass	
Holds conversion methods for the notation type of floats	734
ENameSpaceClass	
Holds conversion methods for the namespace enumeration	735
EnumEntryNode	
Interface for string properties	736
EnumNode	
Interface for string properties	739
ERepresentationClass	
Holds conversion methods for the representation enumeration	744
ESignClass	
Holds conversion methods for the sign enumeration	745
ESlopeClass	
Holds conversion methods for the converter formulas	746
EStandardNameSpaceClass	
Holds conversion methods for the standard namespace enumeration	747
EventHandler	
The base class for all event handler types	749
EVisibilityClass	
Holds conversion methods for the visibility enumeration	752
Exception	
The Exception object represents an error that is returned from the library	754
EYesNoClass	
Holds conversion methods for the standard namespace enumeration	759
FileProtocolAdapter	
Adapter between the std::iostreambuf and the SFNC Features representing the device file system	760

FloatNode	
Interface for string properties	764
FloatRegNode	
Interface for string properties	771
Function_NodeCallback< Function >	
Container for a function pointer	774
gcstring	776
GrabInfo	785
GVCP_CHUNK_TRAILER	
Header of a GVCP request packet	786
GVCP_EVENT_ITEM	
Layout of a GVCP event item (Extended ID flag not set)	787
GVCP_EVENT_ITEM_BASIC	
Layout of a GVCP event item (common to all types)	788
GVCP_EVENT_ITEM_EXTENDED_ID	
Layout of a GVCP event item (Extended ID flag set)	789
GVCP_EVENT_REQUEST	
Layout of a GVCP event request packet (Extended ID flag not set)	791
GVCP_EVENT_REQUEST_EXTENDED_ID	
Layout of a GVCP event request packet (Extended ID flag set)	792
GVCP_EVENTDATA_REQUEST	
Layout of a GVCP event data request packet (Extended ID flag not set)	793
GVCP_EVENTDATA_REQUEST_EXTENDED_ID	
Layout of a GVCP event data request packet (Extended ID flag set)	794
GVCP_REQUEST_HEADER	
Header of a GVCP request packet	795
H264Option	
Options for saving H264 files	796
ICameraBase	
The interface file for base class for the camera object	798
ICameraList	
Used to hold a list of camera objects	807
IChunkData	
The Interface file for ChunkData	811
IDataStream	821
IDevFileStreamBase< CharType, Traits >	828
IDevFileStreamBuf< CharType, Traits >	830
IDeviceArrivalEventHandler	833
IDeviceEventHandler	835
IDeviceRemovalEventHandler	837
IImage	
The interface file for Image	839
IImageEventHandler	852
IImageList	
Used to hold a list of image objects	854
IImageListEventHandler	858
IImageProcessor	
Used to hold a list of image objects	860
IImageStatistics	
The interface file for image statistics	864
IInterface	
The interface file for Interface	868
IInterfaceArrivalEventHandler	873
IInterfaceEventHandler	875
IInterfaceList	
The interface file for InterfaceList class	878
IInterfaceRemovalEventHandler	881
ILoggingEventHandler	883

Image	The image object class	885
ImageEventHandler	A handler for capturing image arrival events	913
ImageEventHandlerImpl		915
ImageInfo		918
ImageList		920
ImageListEventHandler	A handler for capturing image list arrival events	926
ImageProcessor	Image post processing class for converting a source image to another pixel format	929
ImagePtr	A reference tracked pointer to an image object	940
ImageStatistics	Represents image statistics for an image	943
ImageUtility	Static helper functions for the image object class	950
ImageUtilityCCM	Static function to create color corrected images from an image object	954
ImageUtilityHeatmap	Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16	956
ImageUtilityPolarization	Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8	959
InferenceBoundingBox	Inference Bounding Boxes data structure	968
InferenceBoundingBoxResult	An inference bounding boxes object which holds information about the detected bounding boxes	969
InferenceBoxCircle		972
InferenceBoxRect	Inference Bounding Box Type Data Structures	973
InferenceBoxRotatedRect		974
int64_autovector_t	Vector of integers with reference counting	975
IntegerNode	Interface for string properties	978
Interface	An interface object which holds a list of cameras	984
InterfaceArrivalEventHandler	An event handler for capturing the interface arrival event	989
InterfaceEventHandler	A handler to device arrival and removal events on all interfaces	992
InterfaceEventHandlerImpl		995
InterfaceList	A list of the available interfaces on the system	999
InterfacePtr	A reference tracked pointer to the interface object	1004
InterfaceRemovalEventHandler	An event handler for capturing the interface removal event	1006
IntRegNode	Interface for string properties	1008
ISystem	The interface file for System	1011
ISystemEventHandler		1016
JPEGOption	Options for saving JPEG image	1018
JPG2Option	Options for saving JPEG2000 image	1020

LibraryVersion	
Provides easier access to the current version of Spinnaker	1021
LockableObject< Object >::Lock	
A scopelevel Lock class	1022
LockableObject< Object >	
Instance-Lock for an object	1023
LoggingEventData	
The LoggingEventData object	1025
LoggingEventDataPtr	
A reference tracked pointer to the LoggingEvent object	1028
LoggingEventHandler	
An event handler for capturing the device logging event	1030
LoggingEventHandlerImpl	1033
Member_NodeCallback< Client, Member >	
Container for a member function pointer	1034
MJPGOption	
Options for saving MJPG files	1036
Node	
Class common to all nodes	1038
NodeMap	
Smart pointer template for NodeMaps with create function	1049
CNodeMapFactory::NodeStatistics_t	1058
ODevFileStreamBase< CharType, Traits >	1059
ODevFileStreamBuf< CharType, Traits >	1061
PGMOption	
Options for saving PGM images	1063
PNGOption	
Spinnaker struct definitions	1064
PortNode	
Interface for value properties	1066
PortRecorder	
Interface for recording write commands on a port	1072
PortReplay	
Interface for replaying write commands on a port	1076
PPMOption	
Options for saving PPM images	1079
RegisterNode	
Interface for string properties	1080
SingleChunkData_t	1085
SingleChunkDataStr_t	1086
SIOption	
Options for saving Spinnaker image	1087
SpinTestCamera	1088
SpinVideo	
Provides the functionality for the user to record images to an AVI/MP4 file	1088
StringNode	
Interface for string properties	1093
StringRegNode	
Interface for string properties	1097
System	
The system object is used to retrieve the list of interfaces and cameras available	1100
SystemEventHandler	
A handler to interface arrival and removal events on the system	1109
SystemEventHandlerImpl	1112
SystemPtr	
A reference tracked pointer to a system object	1116
TIFFOption	
Options for saving TIFF images	1118

TransportLayerDevice	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1119
TransportLayerInterface	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1130
TransportLayerStream	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1141
TransportLayerSystem	
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera	1150
U3V_CHUNK_TRAILER	
Header of a GVCP request packet	1157
U3V_COMMAND_HEADER	
U3V/GenCP command header	1158
U3V_EVENT_DATA	
U3V/GenCP EVENT_CMD specific command data	1159
U3V_EVENT_MESSAGE	
Entire event data message (without the variable-sized data field)	1160
ValueNode	
Interface for value properties	1161
Version_t	
Version	1164

Chapter 12

File Index

12.1 File List

Here is a list of all files with brief descriptions:

include/AVIRecorder.h	1167
include/BasePtr.h	1168
include/Camera.h	1169
include/CameraBase.h	1169
include/CameraDefs.h	1170
include/CameraList.h	1202
include/CameraPtr.h	1203
include/ChunkData.h	1203
include/ChunkDataInference.h	1204
include/DeviceArrivalEventHandler.h	1205
include/DeviceEventHandler.h	1206
include/DeviceEventUtility.h	1207
include/DeviceRemovalEventHandler.h	1208
include/EventHandler.h	1209
include/Exception.h	1209
include/Image.h	1210
include/ImageEventHandler.h	1211
include/ImageList.h	1211
include/ImageListEventHandler.h	1212
include/ImageProcessor.h	1213
include/ImagePtr.h	1214
include/ImageStatistics.h	1215
include/ImageUtility.h	1216
include/ImageUtilityCCM.h	1217
include/ImageUtilityHeatmap.h	1218
include/ImageUtilityPolarization.h	1219
include/Interface.h	1219
include/InterfaceArrivalEventHandler.h	1235
include/InterfaceEventHandler.h	1236
include/InterfaceList.h	1237
include/InterfacePtr.h	1237
include/InterfaceRemovalEventHandler.h	1238
include/LoggingEventData.h	1239
include/LoggingEventDataPtr.h	1239
include/LoggingEventHandler.h	1240

include/Spinnaker.h	1328
include/SpinnakerDefs.h	1329
include/SpinnakerDirectShow.h	1334
include/SpinnakerPlatform.h	1352
include/SpinUpdate.h	1353
include/SpinVideo.h	1356
include/SpinVideoDefs.h	1357
include/System.h	1358
include/SystemEventHandler.h	1359
include/SystemPtr.h	1360
include/TransportLayerDefs.h	1360
include/TransportLayerDevice.h	1362
include/TransportLayerInterface.h	1363
include/TransportLayerStream.h	1364
include/TransportLayerSystem.h	1364
include/Interface/ICameraBase.h	1220
include/Interface/ICameraList.h	1221
include/Interface/IChunkData.h	1221
include/Interface/IDeviceArrivalEventHandler.h	1222
include/Interface/IDeviceEventHandler.h	1223
include/Interface/IDeviceRemovalEventHandler.h	1223
include/Interface/IImage.h	1224
include/Interface/IImageEventHandler.h	1225
include/Interface/IImageList.h	1225
include/Interface/IImageListEventHandler.h	1226
include/Interface/IImageProcessor.h	1227
include/Interface/IImageStatistics.h	1228
include/Interface/IInterface.h	1228
include/Interface/IInterfaceArrivalEventHandler.h	1229
include/Interface/IInterfaceEventHandler.h	1230
include/Interface/IInterfaceList.h	1231
include/Interface/IInterfaceRemovalEventHandler.h	1231
include/Interface/ILoggingEventHandler.h	1232
include/Interface/IStream.h	1232
include/Interface/ISystem.h	1233
include/Interface/ISystemEventHandler.h	1234
include/SpinGenApi/Autovector.h	1241
include/SpinGenApi/Base.h	1242
include/SpinGenApi/BooleanNode.h	1243
include/SpinGenApi/CategoryNode.h	1244
include/SpinGenApi/ChunkAdapter.h	1245
include/SpinGenApi/ChunkAdapterDcam.h	1246
include/SpinGenApi/ChunkAdapterGeneric.h	1247
include/SpinGenApi/ChunkAdapterGEV.h	1248
include/SpinGenApi/ChunkAdapterU3V.h	1249
include/SpinGenApi/ChunkPort.h	1250
include/SpinGenApi/CommandNode.h	1250
include/SpinGenApi/Compatibility.h	1251
include/SpinGenApi/Container.h	1252
include/SpinGenApi/Counter.h	1252
include/SpinGenApi/EnumClasses.h	1253
include/SpinGenApi/EnumEntryNode.h	1254
include/SpinGenApi/EnumNode.h	1255
include/SpinGenApi/EnumNodeT.h	1256
include/SpinGenApi/EventAdapter.h	1256
include/SpinGenApi/EventAdapter1394.h	1257
include/SpinGenApi/EventAdapterGeneric.h	1258
include/SpinGenApi/EventAdapterGEV.h	1258

include/SpinGenApi/EventAdapterU3V.h	1259
include/SpinGenApi/EventPort.h	1260
include/SpinGenApi/Filestream.h	1261
include/SpinGenApi/FloatNode.h	1262
include/SpinGenApi/FloatRegNode.h	1263
include/SpinGenApi/GCBase.h	1264
include/SpinGenApi/GCString.h	1264
include/SpinGenApi/GCStringVector.h	1266
include/SpinGenApi/GCSynch.h	1267
include/SpinGenApi/GCTypes.h	1268
include/SpinGenApi/GCUtilities.h	1271
include/SpinGenApi/IBoolean.h	1275
include/SpinGenApi/ICategory.h	1276
include/SpinGenApi/IChunkPort.h	1277
include/SpinGenApi/ICommand.h	1279
include/SpinGenApi/IDestroy.h	1280
include/SpinGenApi/IDeviceInfo.h	1281
include/SpinGenApi/IEnumEntry.h	1282
include/SpinGenApi/IEnumeration.h	1283
include/SpinGenApi/IEnumerationT.h	1284
include/SpinGenApi/IFloat.h	1285
include/SpinGenApi/IInteger.h	1287
include/SpinGenApi/INode.h	1288
include/SpinGenApi/INodeMap.h	1291
include/SpinGenApi/INodeMapDyn.h	1293
include/SpinGenApi/IntegerNode.h	1294
include/SpinGenApi/IntRegNode.h	1295
include/SpinGenApi/IPort.h	1296
include/SpinGenApi/IPortConstruct.h	1297
include/SpinGenApi/IPortRecorder.h	1298
include/SpinGenApi/IRegister.h	1299
include/SpinGenApi/ISelector.h	1300
include/SpinGenApi/ISelectorDigit.h	1301
include/SpinGenApi/IString.h	1302
include/SpinGenApi/IValue.h	1303
include/SpinGenApi/Node.h	1304
include/SpinGenApi/NodeCallback.h	1305
include/SpinGenApi/NodeCallbackImpl.h	1307
include/SpinGenApi/NodeMap.h	1307
include/SpinGenApi/NodeMapFactory.h	1308
include/SpinGenApi/NodeMapRef.h	1309
include/SpinGenApi/Persistence.h	1310
include/SpinGenApi/Pointer.h	1311
include/SpinGenApi/PortImpl.h	1313
include/SpinGenApi/PortNode.h	1314
include/SpinGenApi/PortRecorder.h	1315
include/SpinGenApi/PortReplay.h	1316
include/SpinGenApi/PortWriteList.h	1317
include/SpinGenApi/Reference.h	1318
include/SpinGenApi/RegisterNode.h	1319
include/SpinGenApi/RegisterPortImpl.h	1319
include/SpinGenApi/SelectorSet.h	1320
include/SpinGenApi/SpinnakerGenApi.h	1320
include/SpinGenApi/SpinTestCamera.h	1321
include/SpinGenApi/StringNode.h	1321
include/SpinGenApi/StringRegNode.h	1322
include/SpinGenApi/StructPort.h	1323
include/SpinGenApi/Synch.h	1323

include/SpinGenApi/Types.h	1324
include/SpinGenApi/ValueNode.h	1327
src/Acquisition/Acquisition.cpp	1365
src/Acquisition/resource.h	1368
src/Acquisition/stdafx.cpp	1368
src/Acquisition/stdafx.h	1377
src/Acquisition/targetver.h	1393
src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp	1407
src/AcquisitionMultipleCameraRecovery/resource.h	1368
src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp	1410
src/AcquisitionMultipleCamerasWriteToFile/resource.h	1368
src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp	1412
src/AcquisitionMultipleThread/resource.h	1368
src/ActionCommand/ActionCommand.cpp	1414
src/ActionCommand/resource.h	1368
src/ActionCommand/stdafx.cpp	1369
src/ActionCommand/stdafx.h	1378
src/ActionCommand/targetver.h	1394
src/BufferHandling/BufferHandling.cpp	1416
src/BufferHandling/resource.h	1368
src/BufferHandling/stdafx.cpp	1369
src/BufferHandling/stdafx.h	1379
src/BufferHandling/targetver.h	1395
src/ChunkData/ChunkData.cpp	1418
src/ChunkData/resource.h	1368
src/Compression/Compression.cpp	1421
src/Compression/resource.h	1368
src/CounterAndTimer/CounterAndTimer.cpp	1424
src/CounterAndTimer/resource.h	1368
src/CounterAndTimer/stdafx.cpp	1370
src/CounterAndTimer/stdafx.h	1380
src/CounterAndTimer/targetver.h	1396
src/DeviceEvents/DeviceEvents.cpp	1425
src/DeviceEvents/resource.h	1368
src/DeviceEvents/stdafx.cpp	1370
src/DeviceEvents/stdafx.h	1381
src/DeviceEvents/targetver.h	1397
src/Enumeration/Enumeration.cpp	1428
src/Enumeration/resource.h	1368
src/Enumeration/stdafx.cpp	1371
src/Enumeration/stdafx.h	1382
src/Enumeration/targetver.h	1398
src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp	1429
src/Enumeration_QuickSpin/resource.h	1368
src/Enumeration_QuickSpin/stdafx.cpp	1371
src/Enumeration_QuickSpin/stdafx.h	1383
src/Enumeration_QuickSpin/targetver.h	1399
src/EnumerationEvents/EnumerationEvents.cpp	1429
src/EnumerationEvents/resource.h	1368
src/ExceptionHandling/ExceptionHandling.cpp	1430
src/ExceptionHandling/resource.h	1368
src/ExceptionHandling/stdafx.cpp	1372
src/ExceptionHandling/stdafx.h	1384
src/ExceptionHandling/targetver.h	1400
src/Exposure/Exposure.cpp	1432
src/Exposure/resource.h	1368
src/Exposure/stdafx.cpp	1372
src/Exposure/stdafx.h	1385

src/Exposure/targetver.h	1401
src/Exposure_QuickSpin/Exposure_QuickSpin.cpp	1433
src/Exposure_QuickSpin/resource.h	1368
src/Exposure_QuickSpin/stdafx.cpp	1373
src/Exposure_QuickSpin/stdafx.h	1386
src/Exposure_QuickSpin/targetver.h	1402
src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp	1435
src/FileAccess_QuickSpin/resource.h	1368
src/FileAccess_QuickSpin/stdafx.cpp	1374
src/FileAccess_QuickSpin/stdafx.h	1387
src/FileAccess_QuickSpin/targetver.h	1403
src/GigEVisionPerformance/CpuUtil.cpp	1438
src/GigEVisionPerformance/CpuUtil.h	1439
src/GigEVisionPerformance/GigEVisionPerformance.cpp	1440
src/GigEVisionPerformance/GigEVisionPerformance.h	1445
src/GigEVisionPerformance/resource.h	1368
src/GigEVisionPerformance/stdafx.cpp	1375
src/GigEVisionPerformance/stdafx.h	1388
src/GigEVisionPerformance/targetver.h	1404
src/ImageEvents/ImageEvents.cpp	1446
src/ImageEvents/resource.h	1368
src/ImageFormatControl/ImageFormatControl.cpp	1448
src/ImageFormatControl/resource.h	1368
src/ImageFormatControl/stdafx.h	1389
src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp	1449
src/ImageFormatControl_QuickSpin/resource.h	1368
src/ImageFormatControl_QuickSpin/stdafx.h	1389
src/Inference/Inference.cpp	1450
src/Inference/resource.h	1368
src/Logging/Logging.cpp	1457
src/Logging/resource.h	1368
src/LogicBlock/LogicBlock.cpp	1458
src/LogicBlock/resource.h	1368
src/LookupTable/LookupTable.cpp	1460
src/LookupTable/resource.h	1368
src/NodeMapCallback/NodeMapCallback.cpp	1461
src/NodeMapCallback/resource.h	1368
src/NodeMapInfo/NodeMapInfo.cpp	1463
src/NodeMapInfo/resource.h	1368
src/NodeMapInfo/stdafx.cpp	1375
src/NodeMapInfo/stdafx.h	1390
src/NodeMapInfo/targetver.h	1405
src/Polarization/Polarization.cpp	1467
src/Polarization/resource.h	1368
src/Polarization/stdafx.h	1391
src/SaveToAvi/resource.h	1368
src/SaveToAvi/SaveToAvi.cpp	1470
src/Sequencer/resource.h	1368
src/Sequencer/Sequencer.cpp	1472
src/Sequencer/stdafx.cpp	1376
src/Sequencer/stdafx.h	1391
src/Sequencer/targetver.h	1406
src/SerialRxTx/resource.h	1368
src/SerialRxTx/SerialRxTx.cpp	1474
src/SerialRxTx/stdafx.cpp	1376
src/SerialRxTx/stdafx.h	1392
src/SerialRxTx/targetver.h	1407
src/Trigger/resource.h	1368

src/Trigger/ Trigger.cpp	1477
src/Trigger_QuickSpin/ resource.h	1368
src/Trigger_QuickSpin/ Trigger_QuickSpin.cpp	1480

Chapter 13

Module Documentation

13.1 Spinnaker Classes

13.2 AVI Recorder Class

13.3 BasePtr Class

13.4 Camera Class

13.5 Camera Base Class

13.6 CameraDefs Class

13.7 Camera List Class

13.8 CameraPtr Class

13.9 ChunkData Class

13.10 Chunk Data Inference Class

13.11 Spinnaker EventHandler Classes

13.12 DeviceArrivalEventHandler Class

13.13 DeviceEventHandler Class

13.14 DeviceRemovalEventHandler Class

13.15 EventHandler Class

13.16 Exception Class

13.17 Image Class

13.18 ImageEventHandler Class

13.19 ImageListEventHandler Class

13.20 Image Processor Class

13.21 ImagePtr Class

13.22 ImageStatistics Class

13.23 Image Utility Class

13.24 Image Utility CCM Class

13.25 Image Utility Heatmap Class

13.26 Image Utility Polarization Class

13.27 Interface Class

13.28 InterfaceArrivalEventHandler Class

13.29 InterfaceEventHandler Class

13.30 InterfaceList Class

13.31 InterfacePtr Class

13.32 InterfaceRemovalEventHandler Class

13.33 Logging EventHandler Class

13.34 LoggingEventDataPtr Class

13.35 LoggingEventHandler Class

13.36 Spinnaker Headers

13.37 Spinnaker.h

Global header file for [Spinnaker](#).

Global header file for [Spinnaker](#).

By including this file, all required header files for full [Spinnaker](#) operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <https://www.flir.com/support-center/rma/iis-su>

13.38 Spinnaker Definitions

Definitions file for [Spinnaker](#).

Definitions file for [Spinnaker](#).

13.39 SpinnakerDirectShow.h

Global header file for [Spinnaker](#) DirectShow library.

Global header file for [Spinnaker](#) DirectShow library.

By including this file, all required header files for full [Spinnaker](#) DirectShow operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <https://www.flir.com/support-center/rma/iis-su>

13.40 Spinnaker Platform

Platform-specific header file for [Spinnaker](#). All the platform-specific code that is required by individual compilers to produce the appropriate code for each platform.

Platform-specific header file for [Spinnaker](#). All the platform-specific code that is required by individual compilers to produce the appropriate code for each platform.

13.41 Spinnaker Video Class

13.42 Spinnaker Video Definitions

Definitions file for [Spinnaker](#) video recorder.

Definitions file for [Spinnaker](#) video recorder.

13.43 System Class

13.44 SystemEventHandler Class

13.45 SystemPtr Class

13.46 Spinnaker QuickSpin Classes

13.47 TransportLayerDefs Class

13.48 TransportLayerDevice Class

13.49 TransportLayerInterface Class

13.50 TransportLayerStream Class

13.51 TransportLayerSystem Class

13.52 Camera Base Interface Class

13.53 IChunkData Class

13.54 IImage Class

13.55 Image List Class

13.56 IImageStatistics Class

13.57 IInterface Class

13.58 IInterfaceList Class

13.59 ISystem Class

13.60 Spinnaker GenApi Classes

Collaboration diagram for Spinnaker GenApi Classes:



13.61 AutoVector Class

13.62 Spinnaker GenApi Interfaces

13.63 IBase Interface

13.64 BooleanNode Class

13.65 CategoryNode Class

13.66 ChunkAdapter Class

13.67 ChunkAdapterDcam Class

13.68 ChunkAdapterGeneric Class

13.69 ChunkAdapterGEV Class

13.70 ChunkPort Class

13.71 CommandNode Class

13.72 Container Class

13.73 Counter Class

13.74 EnumClasses Class

13.75 EnumEntryNode Class

13.76 EnumNode Class

13.77 EnumNodeT Class

13.78 EventAdapter Class

13.79 EventAdapter1394 Class

13.80 EventAdapterGeneric Class

13.81 EventAdapterGEV Class

13.82 EventAdapterU3V Class

13.83 EventPort Class

13.84 Filestream Class

13.85 FloatNode Class

13.86 FloatRegNode Class

13.87 GCString Class

13.88 GCSynch Class

13.89 GCTypes Class

Typedefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

13.89.1 Detailed Description

13.89.2 Typedef Documentation

13.89.2.1 float32_t

```
typedef float float32\_t
```

32 bit floating point

13.89.2.2 float64_t

```
typedef double float64\_t
```

64 bit floating point

13.90 Spinnaker GenApi Utilities

13.91 GCUtilities Utility

13.92 IBoolean Interface

13.93 ICategory Interfaces

13.94 IChunkPort Interface

13.95 ICommand Interface

13.96 IDestroy Interface

13.97 IDeviceInfo Interface

13.98 IEnumEntry Interface

13.99 IEnumeration Interface

13.100 IEnumerationT Interface

13.101 IFloat Interface

13.102 Integer Interface

13.103 INode Interface

Functions

- bool **IsReadable** (EAccessMode AccessMode)
Tests if readable.
- bool **IsReadable** (const **IBase** *p)
Checks if a node is readable.
- bool **IsReadable** (const **IBase** &r)
Checks if a node is readable.
- bool **IsWritable** (EAccessMode AccessMode)
Tests if writable.
- bool **IsWritable** (const **IBase** *p)
Checks if a node is writable.
- bool **IsWritable** (const **IBase** &r)
Checks if a node is writable.
- bool **IsImplemented** (EAccessMode AccessMode)
Tests if implemented.
- bool **IsImplemented** (const **IBase** *p)
Checks if a node is implemented.
- bool **IsImplemented** (const **IBase** &r)
Checks if a node is implemented.
- bool **IsAvailable** (EAccessMode AccessMode)
Tests if available.
- bool **IsAvailable** (const **IBase** *p)
Checks if a node is available.
- bool **IsAvailable** (const **IBase** &r)
Checks if a node is available.
- EAccessMode **Combine** (EAccessMode Peter, EAccessMode Paul)
Computes which access mode the two guards allow together.
- bool **IsVisible** (EVisibility Visibility, EVisibility MaxVisibility)
Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- EVisibility **Combine** (EVisibility Peter, EVisibility Paul)
Computes which visibility the two guards allow together.
- bool **IsCacheable** (ECachingMode CachingMode)
Tests Cacheability.
- ECachingMode **Combine** (ECachingMode Peter, ECachingMode Paul)
Computes which CachingMode results from a combination.

13.103.1 Detailed Description

13.103.2 Function Documentation

13.103.2.1 Combine() [1/3]

```
EAccessMode Spinnaker::GenApi::Combine (  
    EAccessMode Peter,  
    EAccessMode Paul ) [inline]
```

Computes which access mode the two guards allow together.

13.103.2.2 Combine() [2/3]

```
ECachingMode Spinnaker::GenApi::Combine (  
    ECachingMode Peter,  
    ECachingMode Paul ) [inline]
```

Computes which CachingMode results from a combination.

13.103.2.3 Combine() [3/3]

```
EVisibility Spinnaker::GenApi::Combine (  
    EVisibility Peter,  
    EVisibility Paul ) [inline]
```

Computes which visibility the two guards allow together.

13.103.2.4 IsAvailable() [1/3]

```
bool Spinnaker::GenApi::IsAvailable (  
    const IBase & r ) [inline]
```

Checks if a node is available.

13.103.2.5 IsAvailable() [2/3]

```
bool Spinnaker::GenApi::IsAvailable (  
    const IBase * p ) [inline]
```

Checks if a node is available.

13.103.2.6 IsAvailable() [3/3]

```
bool Spinnaker::GenApi::IsAvailable (
    EAccessMode AccessMode ) [inline]
```

Tests if available.

13.103.2.7 IsCacheable()

```
bool Spinnaker::GenApi::IsCacheable (
    ECachingMode CachingMode ) [inline]
```

Tests Cacheability.

13.103.2.8 IsImplemented() [1/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase & r ) [inline]
```

Checks if a node is implemented.

13.103.2.9 IsImplemented() [2/3]

```
bool Spinnaker::GenApi::IsImplemented (
    const IBase * p ) [inline]
```

Checks if a node is implemented.

13.103.2.10 IsImplemented() [3/3]

```
bool Spinnaker::GenApi::IsImplemented (
    EAccessMode AccessMode ) [inline]
```

Tests if implemented.

13.103.2.11 IsReadable() [1/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase & r ) [inline]
```

Checks if a node is readable.

13.103.2.12 IsReadable() [2/3]

```
bool Spinnaker::GenApi::IsReadable (
    const IBase * p ) [inline]
```

Checks if a node is readable.

13.103.2.13 IsReadable() [3/3]

```
bool Spinnaker::GenApi::IsReadable (
    EAccessMode AccessMode ) [inline]
```

Tests if readable.

13.103.2.14 IsVisible()

```
bool Spinnaker::GenApi::IsVisible (
    EVisibility Visibility,
    EVisibility MaxVisiblity ) [inline]
```

Tests Visibility CAVE : this relies on the EVisibility enum's coding.

13.103.2.15 IsWritable() [1/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase & r ) [inline]
```

Checks if a node is writable.

13.103.2.16 IsWritable() [2/3]

```
bool Spinnaker::GenApi::IsWritable (
    const IBase * p ) [inline]
```

Checks if a node is writable.

13.103.2.17 IsWritable() [3/3]

```
bool Spinnaker::GenApi::IsWritable (
    EAccessMode AccessMode ) [inline]
```

Tests if writable.

13.104 INodeMap Interface

13.105 INodeMapDyn Interface

13.106 IntegerNode Class

13.107 IntRegNode Class

13.108 IPort Interface

13.109 IPortConstruct Interface

13.110 IPortRecorder Interface

13.111 IRegister Interfaces

13.112 ISelector Interface

13.113 ISelectorDigit Interface

13.114 IString Class

13.115 IValue Class

13.116 Node Class

13.117 NodeCallback Class

13.118 NodeMap Class

13.119 NodeMapFactory Class

13.120 NodeMapRef Class

13.121 Persistence Class

13.122 Pointer Class

Collaboration diagram for Pointer Class:



Modules

- [Spinnaker GenApi Classes](#)

Classes

- class [CFloatPtr](#)
SmartPointer for IFloat interface pointer.

Typedefs

- typedef [CPointer](#)< [IBase](#) > [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef [CPointer](#)< [INode](#), [IBase](#) > [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef [CPointer](#)< [IValue](#) > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef [CPointer](#)< [ICategory](#) > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef [CPointer](#)< [IBoolean](#) > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef [CPointer](#)< [IInteger](#) > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef [CPointer](#)< [IString](#) > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef [CPointer](#)< [IRegister](#) > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef [CPointer](#)< [IEnumeration](#) > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef [CPointer](#)< [IEnumEntry](#) > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef [CPointer](#)< [IPort](#) > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef [CPointer](#)< [IPortReplay](#) > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.

- typedef [CPointer](#)< [IPortRecorder](#) > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.

Functions

- template<class T , class B >
bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is readable.
- template<class T , class B >
bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Writable.
- template<class T , class B >
bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Implemented.
- template<class T , class B >
bool [IsAvailable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Available.
- [GenICam::gcstring GetInterfaceName](#) ([IBase](#) *pBase)
Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

13.122.1 Detailed Description

13.122.2 Typedef Documentation

13.122.2.1 CBasePtr

```
typedef CPointer<IBase> CBasePtr
```

SmartPointer for IBase interface pointer.

13.122.2.2 CBooleanPtr

```
typedef CPointer<IBoolean> CBooleanPtr
```

SmartPointer for IBoolean interface pointer.

13.122.2.3 CCategoryPtr

```
typedef CPointer<ICategory> CCategoryPtr
```

SmartPointer for ICategory interface pointer.

13.122.2.4 CChunkPortPtr

```
typedef CPointer<IChunkPort> CChunkPortPtr
```

SmartPointer for IChunkPort interface pointer.

13.122.2.5 CCommandPtr

```
typedef CPointer<ICommand> CCommandPtr
```

SmartPointer for ICommand interface pointer.

13.122.2.6 CDeviceInfoPtr

```
typedef CPointer<IDeviceInfo, INodeMap> CDeviceInfoPtr
```

SmartPointer for IDeviceInfo interface pointer.

13.122.2.7 CEnumEntryPtr

```
typedef CPointer<IEnumEntry> CEnumEntryPtr
```

SmartPointer for IEnumEntry interface pointer.

13.122.2.8 CEnumerationPtr

```
typedef CPointer<IEnumeration> CEnumerationPtr
```

SmartPointer for IEnumeration interface pointer.

13.122.2.9 CIntegerPtr

```
typedef CPointer<IInteger> CIntegerPtr
```

SmartPointer for IInteger interface pointer.

13.122.2.10 CNodeMapDynPtr

```
typedef CPointer<INodeMapDyn, INodeMap> CNodeMapDynPtr
```

SmartPointer for INodeMapDyn interface pointer.

13.122.2.11 CNodeMapPtr

```
typedef CPointer<INodeMap, INodeMap> CNodeMapPtr
```

SmartPointer for INodeMap interface pointer.

13.122.2.12 CNodePtr

```
typedef CPointer<INode, IBase> CNodePtr
```

SmartPointer for INode interface pointer.

13.122.2.13 CPortConstructPtr

```
typedef CPointer<IPortConstruct> CPortConstructPtr
```

SmartPointer for IPortConstruct interface pointer.

13.122.2.14 CPortPtr

```
typedef CPointer<IPort> CPortPtr
```

SmartPointer for IPort interface pointer.

13.122.2.15 CPortRecorderPtr

```
typedef CPointer<IPortRecorder> CPortRecorderPtr
```

SmartPointer for IPortRecorder interface pointer.

13.122.2.16 CPortReplayPtr

```
typedef CPointer<IPortReplay> CPortReplayPtr
```

SmartPointer for IPortReplay interface pointer.

13.122.2.17 CPortWriteListPtr

```
typedef CPointer<IPortWriteList, IPortWriteList> CPortWriteListPtr
```

SmartPointer for IPortWriteList interface pointer.

13.122.2.18 CRegisterPtr

```
typedef CPointer<IRegister> CRegisterPtr
```

SmartPointer for IRegister interface pointer.

13.122.2.19 CSelectorPtr

```
typedef CPointer<ISelector> CSelectorPtr
```

SmartPointer for ISelector interface pointer.

13.122.2.20 CStringPtr

```
typedef CPointer<IString> CStringPtr
```

SmartPointer for IString interface pointer.

13.122.2.21 CValuePtr

```
typedef CPointer<IValue> CValuePtr
```

SmartPointer for IValue interface pointer.

13.122.3 Function Documentation

13.122.3.1 GetInterfaceName()

```
GenICam::gcstring Spinnaker::GenApi::GetInterfaceName (
    IBase * pBase ) [inline]
```

Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

13.122.3.2 IsAvailable()

```
bool Spinnaker::GenApi::IsAvailable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Available.

13.122.3.3 IsImplemented()

```
bool Spinnaker::GenApi::IsImplemented (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Implemented.

13.122.3.4 IsReadable()

```
bool Spinnaker::GenApi::IsReadable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is readable.

13.122.3.5 IsWritable()

```
bool Spinnaker::GenApi::IsWritable (
    const Spinnaker::GenApi::CPointer< T, B > & ptr ) [inline]
```

Checks if a node is Writable.

13.123 PortImpl Class

13.124 PortNode Class

13.125 PortRecorder Class

13.126 PortReplay Class

13.127 PortWriteList Class

13.128 Reference Interfaces

13.129 RegisterNode Class

13.130 RegisterPortImpl Class

13.131 SelectorSet Class

13.132 SpinTestCamera Class

13.133 StringNode Class

13.134 StringRegNode Class

13.135 StructPort Class

13.136 Synch Class

13.137 Spinnaker GenApi Enums

13.138 Types Enums

13.139 ValueNode Class

13.140 ChunkAdapterU3V Class

Chapter 14

Namespace Documentation

14.1 Conversion Namespace Reference

Functions

- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)
- string [NumToCString](#) (float number)

14.1.1 Function Documentation

14.1.1.1 NumToCString() [1/3]

```
string NumToCString (  
    double number )
```

14.1.1.2 NumToCString() [2/3]

```
string Conversion::NumToCString (  
    float number )
```

14.1.1.3 NumToCString() [3/3]

```
string NumToCString (  
    int number )
```

14.2 CpuUtil Namespace Reference

Classes

- struct [CpuUsageInfo](#)

Functions

- bool [StartCpuTracing](#) ([CpuUsageInfo](#) *cpuUsage)
- bool [StopCpuTracing](#) ([CpuUsageInfo](#) *cpuUsage)
- std::string [GetCpuStats](#) ([CpuUsageInfo](#) *cpuUsage)

14.2.1 Function Documentation

14.2.1.1 GetCpuStats()

```
std::string GetCpuStats (  
    CpuUsageInfo * cpuUsage )
```

14.2.1.2 StartCpuTracing()

```
bool StartCpuTracing (  
    CpuUsageInfo * cpuUsage )
```

14.2.1.3 StopCpuTracing()

```
bool StopCpuTracing (  
    CpuUsageInfo * cpuUsage )
```

14.3 PerformanceCounter Namespace Reference

Functions

- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()

Variables

- double [PCFreq](#)
- __int64 [CounterStart](#)

14.3.1 Function Documentation

14.3.1.1 GetPerformanceCounter()

```
double GetPerformanceCounter ( )
```

14.3.1.2 StartPerformanceCounter()

```
void StartPerformanceCounter ( )
```

14.3.2 Variable Documentation

14.3.2.1 CounterStart

```
__int64 CounterStart
```

14.3.2.2 PCFreq

```
double PCFreq
```

14.4 SecondsCounter Namespace Reference

Functions

- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()

Variables

- time_t [startTime](#)
- time_t [endTime](#)
- double [timeDiff](#)

14.4.1 Function Documentation

14.4.1.1 GetSecondsCounter()

```
int GetSecondsCounter ( )
```

14.4.1.2 StartSecondsCounter()

```
void StartSecondsCounter ( )
```

14.4.2 Variable Documentation

14.4.2.1 endTime

```
time_t endTime
```

14.4.2.2 startTime

```
time_t startTime
```

14.4.2.3 timeDiff

```
double timeDiff
```

14.5 Spinnaker Namespace Reference

Namespaces

- [GenApi](#)
- [GenICam](#)
- [Video](#)

Classes

- struct [ActionCommandResult](#)
Action Command Result.
- class [BasePtr](#)
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.
- struct [BMPOption](#)
Options for saving Bitmap image.
- class [Camera](#)
The camera object class.
- class [CameraBase](#)
The base class for the camera object.
- class [CameraList](#)
Used to hold a list of camera objects.
- class [CameraPtr](#)
A reference tracked pointer to a camera object.
- struct [CCMSettings](#)
- class [ChunkData](#)
The chunk data which contains additional information about an image.
- class [DeviceArrivalEventHandler](#)
An event handler for capturing the device arrival event.
- struct [DeviceEventExposureEndData](#)
Data Fields for Device Event payload for EventExposureEnd.
- class [DeviceEventHandler](#)
A handler to device events.
- struct [DeviceEventInferenceData](#)
Data fields for Device Event payload for EventInference.
- class [DeviceEventUtility](#)
- class [DeviceRemovalEventHandler](#)
An event handler for capturing the device removal event.
- class [EventHandler](#)
The base class for all event handler types.
- class [Exception](#)
The [Exception](#) object represents an error that is returned from the library.
- class [ICameraBase](#)
The interface file for base class for the camera object.
- class [ICameraList](#)
Used to hold a list of camera objects.
- class [IChunkData](#)
The [Interface](#) file for [ChunkData](#).
- class [IDataStream](#)
- class [IDeviceArrivalEventHandler](#)
- class [IDeviceEventHandler](#)
- class [IDeviceRemovalEventHandler](#)
- class [IImage](#)
The interface file for [Image](#).
- class [IImageEventHandler](#)
- class [IImageList](#)
Used to hold a list of image objects.
- class [IImageListEventHandler](#)
- class [IImageProcessor](#)

- Used to hold a list of image objects.*
- class [IImageStatistics](#)
 - The interface file for image statistics.*
- class [IInterface](#)
 - The interface file for [Interface](#).*
- class [IInterfaceArrivalEventHandler](#)
- class [IInterfaceEventHandler](#)
- class [IInterfaceList](#)
 - The interface file for [InterfaceList](#) class.*
- class [IInterfaceRemovalEventHandler](#)
- class [ILoggingEventHandler](#)
- class [Image](#)
 - The image object class.*
- class [ImageEventHandler](#)
 - A handler for capturing image arrival events.*
- class [ImageList](#)
- class [ImageListEventHandler](#)
 - A handler for capturing image list arrival events.*
- class [ImageProcessor](#)
 - [Image](#) post processing class for converting a source image to another pixel format.*
- class [ImagePtr](#)
 - A reference tracked pointer to an image object.*
- class [ImageStatistics](#)
 - Represents image statistics for an image.*
- class [ImageUtility](#)
 - Static helper functions for the image object class.*
- class [ImageUtilityCCM](#)
 - Static function to create color corrected images from an image object.*
- class [ImageUtilityHeatmap](#)
 - Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.*
- class [ImageUtilityPolarization](#)
 - Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.*
- struct [InferenceBoundingBox](#)
 - Inference Bounding Boxes data structure.*
- class [InferenceBoundingBoxResult](#)
 - An inference bounding boxes object which holds information about the detected bounding boxes.*
- struct [InferenceBoxCircle](#)
- struct [InferenceBoxRect](#)
 - Inference Bounding Box Type Data Structures.*
- struct [InferenceBoxRotatedRect](#)
- class [Interface](#)
 - An interface object which holds a list of cameras.*
- class [InterfaceArrivalEventHandler](#)
 - An event handler for capturing the interface arrival event.*
- class [InterfaceEventHandler](#)
 - A handler to device arrival and removal events on all interfaces.*
- class [InterfaceList](#)
 - A list of the available interfaces on the system.*
- class [InterfacePtr](#)
 - A reference tracked pointer to the interface object.*
- class [InterfaceRemovalEventHandler](#)

- An event handler for capturing the interface removal event.*
- class [ISystem](#)
 - The interface file for [System](#).*
- class [ISystemEventHandler](#)
- struct [JPEGOption](#)
 - Options for saving JPEG image.*
- struct [JPG2Option](#)
 - Options for saving JPEG2000 image.*
- struct [LibraryVersion](#)
 - Provides easier access to the current version of [Spinnaker](#).*
- class [LoggingEventData](#)
 - The [LoggingEventData](#) object.*
- class [LoggingEventDataPtr](#)
 - A reference tracked pointer to the [LoggingEvent](#) object.*
- class [LoggingEventHandler](#)
 - An event handler for capturing the device logging event.*
- struct [PGMOption](#)
 - Options for saving PGM images.*
- struct [PNGOption](#)
 - [Spinnaker](#) struct definitions.*
- struct [PPMOption](#)
 - Options for saving PPM images.*
- struct [SIOption](#)
 - Options for saving [Spinnaker](#) image.*
- class [System](#)
 - The system object is used to retrieve the list of interfaces and cameras available.*
- class [SystemEventHandler](#)
 - A handler to interface arrival and removal events on the system.*
- class [SystemPtr](#)
 - A reference tracked pointer to a system object.*
- struct [TIFFOption](#)
 - Options for saving TIFF images.*
- class [TransportLayerDevice](#)
 - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*
- class [TransportLayerInterface](#)
 - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*
- class [TransportLayerStream](#)
 - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*
- class [TransportLayerSystem](#)
 - Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

Enumerations

- enum [LUTSelectorEnums](#) {
[LUTSelector_LUT1](#),
[NUM_LUTSELECTOR](#) }
 - The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.*
- enum [ExposureModeEnums](#) {
[ExposureMode_Timed](#),
[ExposureMode_TriggerWidth](#),
[NUM_EXPOSUREMODE](#) }

- enum [AcquisitionModeEnums](#) {
 [AcquisitionMode_Continuous](#),
 [AcquisitionMode_SingleFrame](#),
 [AcquisitionMode_MultiFrame](#),
 [NUM_ACQUISITIONMODE](#) }
- enum [TriggerSourceEnums](#) {
 [TriggerSource_Software](#),
 [TriggerSource_Line0](#),
 [TriggerSource_Line1](#),
 [TriggerSource_Line2](#),
 [TriggerSource_Line3](#),
 [TriggerSource_UserOutput0](#),
 [TriggerSource_UserOutput1](#),
 [TriggerSource_UserOutput2](#),
 [TriggerSource_UserOutput3](#),
 [TriggerSource_Counter0Start](#),
 [TriggerSource_Counter1Start](#),
 [TriggerSource_Counter0End](#),
 [TriggerSource_Counter1End](#),
 [TriggerSource_LogicBlock0](#),
 [TriggerSource_LogicBlock1](#),
 [TriggerSource_Action0](#),
 [NUM_TRIGGERSOURCE](#) }
- enum [TriggerActivationEnums](#) {
 [TriggerActivation_LevelLow](#),
 [TriggerActivation_LevelHigh](#),
 [TriggerActivation_FallingEdge](#),
 [TriggerActivation_RisingEdge](#),
 [TriggerActivation_AnyEdge](#),
 [NUM_TRIGGERACTIVATION](#) }
- enum [SensorShutterModeEnums](#) {
 [SensorShutterMode_Global](#),
 [SensorShutterMode_Rolling](#),
 [SensorShutterMode_GlobalReset](#),
 [NUM_SENSORSHUTTERMODE](#) }
- enum [TriggerModeEnums](#) {
 [TriggerMode_Off](#),
 [TriggerMode_On](#),
 [NUM_TRIGGERMODE](#) }
- enum [TriggerOverlapEnums](#) {
 [TriggerOverlap_Off](#),
 [TriggerOverlap_ReadOut](#),
 [TriggerOverlap_PreviousFrame](#),
 [NUM_TRIGGEROVERLAP](#) }
- enum [TriggerSelectorEnums](#) {
 [TriggerSelector_AcquisitionStart](#),
 [TriggerSelector_FrameStart](#),
 [TriggerSelector_FrameBurstStart](#),
 [NUM_TRIGGERSELECTOR](#) }
- enum [ExposureAutoEnums](#) {
 [ExposureAuto_Off](#),
 [ExposureAuto_Once](#),
 [ExposureAuto_Continuous](#),
 [NUM_EXPOSUREAUTO](#) }
- enum [EventSelectorEnums](#) {
 [EventSelector_Error](#),
 [EventSelector_ExposureEnd](#),

- EventSelector_SerialPortReceive,
NUM_EVENTSELECTOR }
- enum EventNotificationEnums {
EventNotification_On,
EventNotification_Off,
NUM_EVENTNOTIFICATION }
- enum LogicBlockSelectorEnums {
LogicBlockSelector_LogicBlock0,
LogicBlockSelector_LogicBlock1,
NUM_LOGICBLOCKSELECTOR }
- enum LogicBlockLUTInputActivationEnums {
LogicBlockLUTInputActivation_LevelLow,
LogicBlockLUTInputActivation_LevelHigh,
LogicBlockLUTInputActivation_FallingEdge,
LogicBlockLUTInputActivation_RisingEdge,
LogicBlockLUTInputActivation_AnyEdge,
NUM_LOGICBLOCKLUTINPUTACTIVATION }
- enum LogicBlockLUTInputSelectorEnums {
LogicBlockLUTInputSelector_Input0,
LogicBlockLUTInputSelector_Input1,
LogicBlockLUTInputSelector_Input2,
LogicBlockLUTInputSelector_Input3,
NUM_LOGICBLOCKLUTINPUTSELECTOR }
- enum LogicBlockLUTInputSourceEnums {
LogicBlockLUTInputSource_Zero,
LogicBlockLUTInputSource_Line0,
LogicBlockLUTInputSource_Line1,
LogicBlockLUTInputSource_Line2,
LogicBlockLUTInputSource_Line3,
LogicBlockLUTInputSource_UserOutput0,
LogicBlockLUTInputSource_UserOutput1,
LogicBlockLUTInputSource_UserOutput2,
LogicBlockLUTInputSource_UserOutput3,
LogicBlockLUTInputSource_Counter0Start,
LogicBlockLUTInputSource_Counter1Start,
LogicBlockLUTInputSource_Counter0End,
LogicBlockLUTInputSource_Counter1End,
LogicBlockLUTInputSource_LogicBlock0,
LogicBlockLUTInputSource_LogicBlock1,
LogicBlockLUTInputSource_ExposureStart,
LogicBlockLUTInputSource_ExposureEnd,
LogicBlockLUTInputSource_FrameTriggerWait,
LogicBlockLUTInputSource_AcquisitionActive,
NUM_LOGICBLOCKLUTINPUTSOURCE }
- enum LogicBlockLUTSelectorEnums {
LogicBlockLUTSelector_Value,
LogicBlockLUTSelector_Enable,
NUM_LOGICBLOCKLUTSELECTOR }
- enum ColorTransformationSelectorEnums {
ColorTransformationSelector_RGBtoRGB,
ColorTransformationSelector_RGBtoYUV,
NUM_COLORTRANSFORMATIONSELECTOR }
- enum RgbTransformLightSourceEnums {
RgbTransformLightSource_General,
RgbTransformLightSource_Tungsten2800K,
RgbTransformLightSource_WarmFluorescent3000K,
RgbTransformLightSource_CoolFluorescent4000K,
RgbTransformLightSource_Daylight5000K,

- [RgbTransformLightSource_Cloudy6500K,](#)
 - [RgbTransformLightSource_Shade8000K,](#)
 - [RgbTransformLightSource_Custom,](#)
 - [NUM_RGBTRANSFORMLIGHTSOURCE }](#)
- [enum ColorTransformationValueSelectorEnums {](#)
 - [ColorTransformationValueSelector_Gain00,](#)
 - [ColorTransformationValueSelector_Gain01,](#)
 - [ColorTransformationValueSelector_Gain02,](#)
 - [ColorTransformationValueSelector_Gain10,](#)
 - [ColorTransformationValueSelector_Gain11,](#)
 - [ColorTransformationValueSelector_Gain12,](#)
 - [ColorTransformationValueSelector_Gain20,](#)
 - [ColorTransformationValueSelector_Gain21,](#)
 - [ColorTransformationValueSelector_Gain22,](#)
 - [ColorTransformationValueSelector_Offset0,](#)
 - [ColorTransformationValueSelector_Offset1,](#)
 - [ColorTransformationValueSelector_Offset2,](#)
 - [NUM_COLORTRANSFORMATIONVALUESELECTOR }](#)
- [enum DeviceRegistersEndiannessEnums {](#)
 - [DeviceRegistersEndianness_Little,](#)
 - [DeviceRegistersEndianness_Big,](#)
 - [NUM_DEVICEREGISTERSENDIANNESSE }](#)
- [enum DeviceScanTypeEnums {](#)
 - [DeviceScanType_Areascan,](#)
 - [NUM_DEVICESCANTYPE }](#)
- [enum DeviceCharacterSetEnums {](#)
 - [DeviceCharacterSet_UTF8,](#)
 - [DeviceCharacterSet_ASCII,](#)
 - [NUM_DEVICECHARACTERSET }](#)
- [enum DeviceTLTypeEnums {](#)
 - [DeviceTLType_GigEVision,](#)
 - [DeviceTLType_CameraLink,](#)
 - [DeviceTLType_CameraLinkHS,](#)
 - [DeviceTLType_CoaXPress,](#)
 - [DeviceTLType_USB3Vision,](#)
 - [DeviceTLType_Custom,](#)
 - [NUM_DEVICETLTYPE }](#)
- [enum DevicePowerSupplySelectorEnums {](#)
 - [DevicePowerSupplySelector_External,](#)
 - [NUM_DEVICEPOWERSUPPLYSELECTOR }](#)
- [enum DeviceTemperatureSelectorEnums {](#)
 - [DeviceTemperatureSelector_Sensor,](#)
 - [NUM_DEVICETEMPERATURESELECTOR }](#)
- [enum DeviceIndicatorModeEnums {](#)
 - [DeviceIndicatorMode_Inactive,](#)
 - [DeviceIndicatorMode_Active,](#)
 - [DeviceIndicatorMode_ErrorStatus,](#)
 - [NUM_DEVICEINDICATORMODE }](#)
- [enum AutoExposureControlPriorityEnums {](#)
 - [AutoExposureControlPriority_Gain,](#)
 - [AutoExposureControlPriority_ExposureTime,](#)
 - [NUM_AUTOEXPOSURECONTROLPRIORITY }](#)
- [enum AutoExposureMeteringModeEnums {](#)
 - [AutoExposureMeteringMode_Average,](#)
 - [AutoExposureMeteringMode_Spot,](#)
 - [AutoExposureMeteringMode_Partial,](#)
 - [AutoExposureMeteringMode_CenterWeighted,](#)

- AutoExposureMeteringMode_HistogramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,
GevIEEE1588Status_Listening,
GevIEEE1588Status_PreMaster,
GevIEEE1588Status_Master,
GevIEEE1588Status_Passive,
GevIEEE1588Status_Uncalibrated,
GevIEEE1588Status_Slave,
NUM_GEVIEEE1588STATUS }
- enum GevIEEE1588ModeEnums {
GevIEEE1588Mode_Auto,
GevIEEE1588Mode_SlaveOnly,
NUM_GEVIEEE1588MODE }
- enum GevIEEE1588ClockAccuracyEnums {
GevIEEE1588ClockAccuracy_Unknown,
NUM_GEVIEEE1588CLOCKACCURACY }
- enum GevCCPEnums {
GevCCP_OpenAccess,
GevCCP_ExclusiveAccess,
GevCCP_ControlAccess,
NUM_GEVCCP }
- enum GevSupportedOptionSelectorEnums {
GevSupportedOptionSelector_UserDefinedName,
GevSupportedOptionSelector_SerialNumber,
GevSupportedOptionSelector_HeartbeatDisable,
GevSupportedOptionSelector_LinkSpeed,
GevSupportedOptionSelector_CCPApplicationSocket,
GevSupportedOptionSelector_ManifestTable,
GevSupportedOptionSelector_TestData,
GevSupportedOptionSelector_DiscoveryAckDelay,
GevSupportedOptionSelector_DiscoveryAckDelayWritable,
GevSupportedOptionSelector_ExtendedStatusCodes,
GevSupportedOptionSelector_Action,
GevSupportedOptionSelector_PendingAck,
GevSupportedOptionSelector_EventData,
GevSupportedOptionSelector_Event,

```

GevSupportedOptionSelector_PacketResend,
GevSupportedOptionSelector_WriteMem,
GevSupportedOptionSelector_CommandsConcatenation,
GevSupportedOptionSelector_IPConfigurationLLA,
GevSupportedOptionSelector_IPConfigurationDHCP,
GevSupportedOptionSelector_IPConfigurationPersistentIP,
GevSupportedOptionSelector_StreamChannelSourceSocket,
GevSupportedOptionSelector_MessageChannelSourceSocket,
NUM_GEVSUPPORTEDOPTIONSELECTOR }

• enum BlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }

• enum BalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }

• enum GainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }

• enum BalanceRatioSelectorEnums {
    BalanceRatioSelector_Red,
    BalanceRatioSelector_Blue,
    NUM_BALANCERATIOSELECTOR }

• enum GainSelectorEnums {
    GainSelector_All,
    NUM_GAINSELECTOR }

• enum DefectCorrectionModeEnums {
    DefectCorrectionMode_Average,
    DefectCorrectionMode_Highlight,
    DefectCorrectionMode_Zero,
    NUM_DEFECTCORRECTIONMODE }

• enum UserSetSelectorEnums {
    UserSetSelector_Default,
    UserSetSelector_UserSet0,
    UserSetSelector_UserSet1,
    NUM_USERSETSELECTOR }

• enum UserSetDefaultEnums {
    UserSetDefault_Default,
    UserSetDefault_UserSet0,
    UserSetDefault_UserSet1,
    NUM_USERSETDEFAULT }

• enum SerialPortBaudRateEnums {
    SerialPortBaudRate_Baud300,
    SerialPortBaudRate_Baud600,
    SerialPortBaudRate_Baud1200,
    SerialPortBaudRate_Baud2400,
    SerialPortBaudRate_Baud4800,
    SerialPortBaudRate_Baud9600,
    SerialPortBaudRate_Baud14400,
    SerialPortBaudRate_Baud19200,
    SerialPortBaudRate_Baud38400,
    SerialPortBaudRate_Baud57600,
    SerialPortBaudRate_Baud115200,

```


- SerialPortBaudRate_Baud230400,
- SerialPortBaudRate_Baud460800,
- SerialPortBaudRate_Baud921600,
- NUM_SERIALPORTBAUDRATE }
- enum SerialPortParityEnums {
 - SerialPortParity_None,
 - SerialPortParity_Odd,
 - SerialPortParity_Even,
 - SerialPortParity_Mark,
 - SerialPortParity_Space,
 - NUM_SERIALPORTPARITY }
- enum SerialPortSelectorEnums {
 - SerialPortSelector_SerialPort0,
 - NUM_SERIALPORTSELECTOR }
- enum SerialPortStopBitsEnums {
 - SerialPortStopBits_Bits1,
 - SerialPortStopBits_Bits1AndAHalf,
 - SerialPortStopBits_Bits2,
 - NUM_SERIALPORTSTOPBITS }
- enum SerialPortSourceEnums {
 - SerialPortSource_Line0,
 - SerialPortSource_Line1,
 - SerialPortSource_Line2,
 - SerialPortSource_Line3,
 - SerialPortSource_Off,
 - NUM_SERIALPORTSOURCE }
- enum SequencerModeEnums {
 - SequencerMode_Off,
 - SequencerMode_On,
 - NUM_SEQUENCERMODE }
- enum SequencerConfigurationValidEnums {
 - SequencerConfigurationValid_No,
 - SequencerConfigurationValid_Yes,
 - NUM_SEQUENCERCONFIGURATIONVALID }
- enum SequencerSetValidEnums {
 - SequencerSetValid_No,
 - SequencerSetValid_Yes,
 - NUM_SEQUENCERSETVALID }
- enum SequencerTriggerActivationEnums {
 - SequencerTriggerActivation_RisingEdge,
 - SequencerTriggerActivation_FallingEdge,
 - SequencerTriggerActivation_AnyEdge,
 - SequencerTriggerActivation_LevelHigh,
 - SequencerTriggerActivation_LevelLow,
 - NUM_SEQUENCERTRIGGERACTIVATION }
- enum SequencerConfigurationModeEnums {
 - SequencerConfigurationMode_Off,
 - SequencerConfigurationMode_On,
 - NUM_SEQUENCERCONFIGURATIONMODE }
- enum SequencerTriggerSourceEnums {
 - SequencerTriggerSource_Off,
 - SequencerTriggerSource_FrameStart,
 - NUM_SEQUENCERTRIGGERSOURCE }
- enum TransferQueueModeEnums {
 - TransferQueueMode_FirstInFirstOut,
 - NUM_TRANSFERQUEUEMODE }
- enum TransferOperationModeEnums {
 - TransferOperationMode_Continuous,

```

TransferOperationMode_MultiBlock,
NUM_TRANSFEROPERATIONMODE }
• enum TransferControlModeEnums {
TransferControlMode_Basic,
TransferControlMode_Automatic,
TransferControlMode_UserControlled,
NUM_TRANSFERCONTROLMODE }
• enum ChunkGainSelectorEnums {
ChunkGainSelector_All,
ChunkGainSelector_Red,
ChunkGainSelector_Green,
ChunkGainSelector_Blue,
NUM_CHUNKGAINSELECTOR }
• enum ChunkSelectorEnums {
ChunkSelector_Image,
ChunkSelector_CRC,
ChunkSelector_FrameID,
ChunkSelector_OffsetX,
ChunkSelector_OffsetY,
ChunkSelector_Width,
ChunkSelector_Height,
ChunkSelector_ExposureTime,
ChunkSelector_Gain,
ChunkSelector_BlackLevel,
ChunkSelector_PixelFormat,
ChunkSelector_Timestamp,
ChunkSelector_SequencerSetActive,
ChunkSelector_SerialData,
ChunkSelector_ExposureEndLineStatusAll,
NUM_CHUNKSELECTOR }
• enum ChunkBlackLevelSelectorEnums {
ChunkBlackLevelSelector_All,
NUM_CHUNKBLACKLEVELSELECTOR }
• enum ChunkPixelFormatEnums {
ChunkPixelFormat_Mono8,
ChunkPixelFormat_Mono12Packed,
ChunkPixelFormat_Mono16,
ChunkPixelFormat_RGB8Packed,
ChunkPixelFormat_YUV422Packed,
ChunkPixelFormat_BayerGR8,
ChunkPixelFormat_BayerRG8,
ChunkPixelFormat_BayerGB8,
ChunkPixelFormat_BayerBG8,
ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
NUM_CHUNKPIXELFORMAT }
• enum FileOperationStatusEnums {
FileOperationStatus_Success,
FileOperationStatus_Failure,
FileOperationStatus_Overflow,
NUM_FILEOPERATIONSTATUS }
• enum FileOpenModeEnums {
FileOpenMode_Read,
FileOpenMode_Write,
FileOpenMode_ReadWrite,
NUM_FILEOPENMODE }
• enum FileOperationSelectorEnums {
FileOperationSelector_Open,
FileOperationSelector_Close,

```

```
FileOperationSelector_Read,  
FileOperationSelector_Write,  
FileOperationSelector_Delete,  
NUM_FILEOPERATIONSELECTOR }  
  
• enum FileSelectorEnums {  
FileSelector_UserSetDefault,  
FileSelector_UserSet0,  
FileSelector_UserSet1,  
FileSelector_UserFile1,  
FileSelector_SerialPort0,  
NUM_FILESELECTOR }  
  
• enum BinningSelectorEnums {  
BinningSelector_All,  
BinningSelector_Sensor,  
BinningSelector_ISP,  
NUM_BINNINGSELECTOR }  
  
• enum TestPatternGeneratorSelectorEnums {  
TestPatternGeneratorSelector_Sensor,  
TestPatternGeneratorSelector_PipelineStart,  
NUM_TESTPATTERNGENERATORSELECTOR }  
  
• enum CompressionSaturationPriorityEnums {  
CompressionSaturationPriority_DropFrame,  
CompressionSaturationPriority_ReduceFrameRate,  
NUM_COMPRESSIONSATURATIONPRIORITY }  
  
• enum TestPatternEnums {  
TestPattern_Off,  
TestPattern_Increment,  
TestPattern_SensorTestPattern,  
NUM_TESTPATTERN }  
  
• enum PixelColorFilterEnums {  
PixelColorFilter_None,  
PixelColorFilter_BayerRG,  
PixelColorFilter_BayerGB,  
PixelColorFilter_BayerGR,  
PixelColorFilter_BayerBG,  
NUM_PIXELCOLORFILTER }  
  
• enum AdcBitDepthEnums {  
AdcBitDepth_Bit8,  
AdcBitDepth_Bit10,  
AdcBitDepth_Bit12,  
AdcBitDepth_Bit14,  
NUM_ADCBITDEPTH }  
  
• enum DecimationHorizontalModeEnums {  
DecimationHorizontalMode_Discard,  
NUM_DECIMATIONHORIZONTALMODE }  
  
• enum BinningVerticalModeEnums {  
BinningVerticalMode_Sum,  
BinningVerticalMode_Average,  
NUM_BINNINGVERTICALMODE }  
  
• enum PixelSizeEnums {  
PixelSize_Bpp1,  
PixelSize_Bpp2,  
PixelSize_Bpp4,  
PixelSize_Bpp8,  
PixelSize_Bpp10,  
PixelSize_Bpp12,  
PixelSize_Bpp14,  
PixelSize_Bpp16,
```

```
PixelSize_Bpp20,  
PixelSize_Bpp24,  
PixelSize_Bpp30,  
PixelSize_Bpp32,  
PixelSize_Bpp36,  
PixelSize_Bpp48,  
PixelSize_Bpp64,  
PixelSize_Bpp96,  
NUM_PIXELSIZE }  
  
• enum DecimationSelectorEnums {  
    DecimationSelector_All,  
    DecimationSelector_Sensor,  
    NUM_DECIMATIONSELECTOR }  
  
• enum ImageCompressionModeEnums {  
    ImageCompressionMode_Off,  
    ImageCompressionMode_Lossless,  
    NUM_IMAGECOMPRESSIONMODE }  
  
• enum BinningHorizontalModeEnums {  
    BinningHorizontalMode_Sum,  
    BinningHorizontalMode_Average,  
    NUM_BINNINGHORIZONTALMODE }  
  
• enum PixelFormatEnums {  
    PixelFormat_Mono8,  
    PixelFormat_Mono16,  
    PixelFormat_RGB8Packed,  
    PixelFormat_BayerGR8,  
    PixelFormat_BayerRG8,  
    PixelFormat_BayerGB8,  
    PixelFormat_BayerBG8,  
    PixelFormat_BayerGR16,  
    PixelFormat_BayerRG16,  
    PixelFormat_BayerGB16,  
    PixelFormat_BayerBG16,  
    PixelFormat_Mono12Packed,  
    PixelFormat_BayerGR12Packed,  
    PixelFormat_BayerRG12Packed,  
    PixelFormat_BayerGB12Packed,  
    PixelFormat_BayerBG12Packed,  
    PixelFormat_YUV411Packed,  
    PixelFormat_YUV422Packed,  
    PixelFormat_YUV444Packed,  
    PixelFormat_Mono12p,  
    PixelFormat_BayerGR12p,  
    PixelFormat_BayerRG12p,  
    PixelFormat_BayerGB12p,  
    PixelFormat_BayerBG12p,  
    PixelFormat_YCbCr8,  
    PixelFormat_YCbCr422_8,  
    PixelFormat_YCbCr411_8,  
    PixelFormat_BGR8,  
    PixelFormat_BGRa8,  
    PixelFormat_Mono10Packed,  
    PixelFormat_BayerGR10Packed,  
    PixelFormat_BayerRG10Packed,  
    PixelFormat_BayerGB10Packed,  
    PixelFormat_BayerBG10Packed,  
    PixelFormat_Mono10p,  
    PixelFormat_BayerGR10p,
```

[PixelFormat_BayerRG10p,](#)
[PixelFormat_BayerGB10p,](#)
[PixelFormat_BayerBG10p,](#)
[PixelFormat_Mono1p,](#)
[PixelFormat_Mono2p,](#)
[PixelFormat_Mono4p,](#)
[PixelFormat_Mono8s,](#)
[PixelFormat_Mono10,](#)
[PixelFormat_Mono12,](#)
[PixelFormat_Mono14,](#)
[PixelFormat_Mono16s,](#)
[PixelFormat_Mono32f,](#)
[PixelFormat_BayerBG10,](#)
[PixelFormat_BayerBG12,](#)
[PixelFormat_BayerGB10,](#)
[PixelFormat_BayerGB12,](#)
[PixelFormat_BayerGR10,](#)
[PixelFormat_BayerGR12,](#)
[PixelFormat_BayerRG10,](#)
[PixelFormat_BayerRG12,](#)
[PixelFormat_RGBa8,](#)
[PixelFormat_RGBa10,](#)
[PixelFormat_RGBa10p,](#)
[PixelFormat_RGBa12,](#)
[PixelFormat_RGBa12p,](#)
[PixelFormat_RGBa14,](#)
[PixelFormat_RGBa16,](#)
[PixelFormat_RGB8,](#)
[PixelFormat_RGB8_Planar,](#)
[PixelFormat_RGB10,](#)
[PixelFormat_RGB10_Planar,](#)
[PixelFormat_RGB10p,](#)
[PixelFormat_RGB10p32,](#)
[PixelFormat_RGB12,](#)
[PixelFormat_RGB12_Planar,](#)
[PixelFormat_RGB12p,](#)
[PixelFormat_RGB14,](#)
[PixelFormat_RGB16,](#)
[PixelFormat_RGB16s,](#)
[PixelFormat_RGB32f,](#)
[PixelFormat_RGB16_Planar,](#)
[PixelFormat_RGB565p,](#)
[PixelFormat_BGRa10,](#)
[PixelFormat_BGRa10p,](#)
[PixelFormat_BGRa12,](#)
[PixelFormat_BGRa12p,](#)
[PixelFormat_BGRa14,](#)
[PixelFormat_BGRa16,](#)
[PixelFormat_RGBa32f,](#)
[PixelFormat_BGR10,](#)
[PixelFormat_BGR10p,](#)
[PixelFormat_BGR12,](#)
[PixelFormat_BGR12p,](#)
[PixelFormat_BGR14,](#)
[PixelFormat_BGR16,](#)
[PixelFormat_BGR565p,](#)
[PixelFormat_R8,](#)
[PixelFormat_R10,](#)

[PixelFormat_R12,](#)
[PixelFormat_R16,](#)
[PixelFormat_G8,](#)
[PixelFormat_G10,](#)
[PixelFormat_G12,](#)
[PixelFormat_G16,](#)
[PixelFormat_B8,](#)
[PixelFormat_B10,](#)
[PixelFormat_B12,](#)
[PixelFormat_B16,](#)
[PixelFormat_Coord3D_ABC8,](#)
[PixelFormat_Coord3D_ABC8_Planar,](#)
[PixelFormat_Coord3D_ABC10p,](#)
[PixelFormat_Coord3D_ABC10p_Planar,](#)
[PixelFormat_Coord3D_ABC12p,](#)
[PixelFormat_Coord3D_ABC12p_Planar,](#)
[PixelFormat_Coord3D_ABC16,](#)
[PixelFormat_Coord3D_ABC16_Planar,](#)
[PixelFormat_Coord3D_ABC32f,](#)
[PixelFormat_Coord3D_ABC32f_Planar,](#)
[PixelFormat_Coord3D_AC8,](#)
[PixelFormat_Coord3D_AC8_Planar,](#)
[PixelFormat_Coord3D_AC10p,](#)
[PixelFormat_Coord3D_AC10p_Planar,](#)
[PixelFormat_Coord3D_AC12p,](#)
[PixelFormat_Coord3D_AC12p_Planar,](#)
[PixelFormat_Coord3D_AC16,](#)
[PixelFormat_Coord3D_AC16_Planar,](#)
[PixelFormat_Coord3D_AC32f,](#)
[PixelFormat_Coord3D_AC32f_Planar,](#)
[PixelFormat_Coord3D_A8,](#)
[PixelFormat_Coord3D_A10p,](#)
[PixelFormat_Coord3D_A12p,](#)
[PixelFormat_Coord3D_A16,](#)
[PixelFormat_Coord3D_A32f,](#)
[PixelFormat_Coord3D_B8,](#)
[PixelFormat_Coord3D_B10p,](#)
[PixelFormat_Coord3D_B12p,](#)
[PixelFormat_Coord3D_B16,](#)
[PixelFormat_Coord3D_B32f,](#)
[PixelFormat_Coord3D_C8,](#)
[PixelFormat_Coord3D_C10p,](#)
[PixelFormat_Coord3D_C12p,](#)
[PixelFormat_Coord3D_C16,](#)
[PixelFormat_Coord3D_C32f,](#)
[PixelFormat_Confidence1,](#)
[PixelFormat_Confidence1p,](#)
[PixelFormat_Confidence8,](#)
[PixelFormat_Confidence16,](#)
[PixelFormat_Confidence32f,](#)
[PixelFormat_BiColorBGRG8,](#)
[PixelFormat_BiColorBGRG10,](#)
[PixelFormat_BiColorBGRG10p,](#)
[PixelFormat_BiColorBGRG12,](#)
[PixelFormat_BiColorBGRG12p,](#)
[PixelFormat_BiColorRGBG8,](#)
[PixelFormat_BiColorRGBG10,](#)
[PixelFormat_BiColorRGBG10p,](#)

PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,
PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,

```

PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,
PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
PixelFormat_GR12,
PixelFormat_GB12,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

• enum DecimationVerticalModeEnums {
    DecimationVerticalMode_Discard,
    NUM_DECIMATIONVERTICALMODE }

• enum LineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }

• enum LineSourceEnums {
    LineSource_Off,
    LineSource_Line0,
    LineSource_Line1,
    LineSource_Line2,
    LineSource_Line3,

```



```

LineSource_UserOutput0,
LineSource_UserOutput1,
LineSource_UserOutput2,
LineSource_UserOutput3,
LineSource_Counter0Active,
LineSource_Counter1Active,
LineSource_LogicBlock0,
LineSource_LogicBlock1,
LineSource_ExposureActive,
LineSource_FrameTriggerWait,
LineSource_SerialPort0,
LineSource_PPSSignal,
LineSource_AllPixel,
LineSource_AnyPixel,
NUM_LINESOURCE }

• enum LineInputFilterSelectorEnums {
    LineInputFilterSelector_Deg glitch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }

• enum UserOutputSelectorEnums {
    UserOutputSelector_UserOutput0,
    UserOutputSelector_UserOutput1,
    UserOutputSelector_UserOutput2,
    UserOutputSelector_UserOutput3,
    NUM_USEROUTPUTSELECTOR }

• enum LineFormatEnums {
    LineFormat_NoConnect,
    LineFormat_TriState,
    LineFormat_TTL,
    LineFormat_LVDS,
    LineFormat_RS422,
    LineFormat_OptoCoupled,
    LineFormat_OpenDrain,
    NUM_LINEFORMAT }

• enum LineSelectorEnums {
    LineSelector_Line0,
    LineSelector_Line1,
    LineSelector_Line2,
    LineSelector_Line3,
    NUM_LINESELECTOR }

• enum ExposureActiveModeEnums {
    ExposureActiveMode_Line1,
    ExposureActiveMode_AnyPixels,
    ExposureActiveMode_AllPixels,
    NUM_EXPOSUREACTIVEMODE }

• enum CounterTriggerActivationEnums {
    CounterTriggerActivation_LevelLow,
    CounterTriggerActivation_LevelHigh,
    CounterTriggerActivation_FallingEdge,
    CounterTriggerActivation_RisingEdge,
    CounterTriggerActivation_AnyEdge,
    NUM_COUNTERTRIGGERACTIVATION }

• enum CounterSelectorEnums {
    CounterSelector_Counter0,
    CounterSelector_Counter1,
    NUM_COUNTERSELECTOR }

• enum CounterStatusEnums {
    CounterStatus_CounterIdle,

```

```

CounterStatus_CounterTriggerWait,
CounterStatus_CounterActive,
CounterStatus_CounterCompleted,
CounterStatus_CounterOverflow,
NUM_COUNTERSTATUS }

• enum CounterTriggerSourceEnums {
CounterTriggerSource_Off,
CounterTriggerSource_Line0,
CounterTriggerSource_Line1,
CounterTriggerSource_Line2,
CounterTriggerSource_Line3,
CounterTriggerSource_UserOutput0,
CounterTriggerSource_UserOutput1,
CounterTriggerSource_UserOutput2,
CounterTriggerSource_UserOutput3,
CounterTriggerSource_Counter0Start,
CounterTriggerSource_Counter1Start,
CounterTriggerSource_Counter0End,
CounterTriggerSource_Counter1End,
CounterTriggerSource_LogicBlock0,
CounterTriggerSource_LogicBlock1,
CounterTriggerSource_ExposureStart,
CounterTriggerSource_ExposureEnd,
CounterTriggerSource_FrameTriggerWait,
NUM_COUNTERTRIGGERSOURCE }

• enum CounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }

• enum CounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,

```

```

CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
• enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
• enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
• enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }
• enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }
• enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }
• enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }
• enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }
• enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
• enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
• enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }
• enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,

```

```

DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }

• enum SensorTapsEnums {
    SensorTaps_One,
    SensorTaps_Two,
    SensorTaps_Three,
    SensorTaps_Four,
    SensorTaps_Eight,
    SensorTaps_Ten,
    NUM_SENSORTAPS }

• enum SensorDigitizationTapsEnums {
    SensorDigitizationTaps_One,
    SensorDigitizationTaps_Two,
    SensorDigitizationTaps_Three,
    SensorDigitizationTaps_Four,
    SensorDigitizationTaps_Eight,
    SensorDigitizationTaps_Ten,
    NUM_SENSORDIGITIZATIONTAPS }

• enum RegionSelectorEnums {
    RegionSelector_Region0,
    RegionSelector_Region1,
    RegionSelector_Region2,
    RegionSelector_All,
    NUM_REGIONSELECTOR }

• enum RegionModeEnums {
    RegionMode_Off,
    RegionMode_On,
    NUM_REGIONMODE }

• enum RegionDestinationEnums {
    RegionDestination_Stream0,
    RegionDestination_Stream1,
    RegionDestination_Stream2,
    NUM_REGIONDESTINATION }

• enum ImageComponentSelectorEnums {
    ImageComponentSelector_Intensity,
    ImageComponentSelector_Color,
    ImageComponentSelector_Infrared,
    ImageComponentSelector_Ultraviolet,
    ImageComponentSelector_Range,
    ImageComponentSelector_Disparity,
    ImageComponentSelector_Confidence,
    ImageComponentSelector_Scatter,
    NUM_IMAGECOMPONENTSELECTOR }

• enum PixelFormatInfoSelectorEnums {
    PixelFormatInfoSelector_Mono1p,
    PixelFormatInfoSelector_Mono2p,
    PixelFormatInfoSelector_Mono4p,
    PixelFormatInfoSelector_Mono8,
    PixelFormatInfoSelector_Mono8s,
    PixelFormatInfoSelector_Mono10,
    PixelFormatInfoSelector_Mono10p,
    PixelFormatInfoSelector_Mono12,

```

[PixelFormatInfoSelector_Mono12p](#),
[PixelFormatInfoSelector_Mono14](#),
[PixelFormatInfoSelector_Mono16](#),
[PixelFormatInfoSelector_Mono16s](#),
[PixelFormatInfoSelector_Mono32f](#),
[PixelFormatInfoSelector_BayerBG8](#),
[PixelFormatInfoSelector_BayerBG10](#),
[PixelFormatInfoSelector_BayerBG10p](#),
[PixelFormatInfoSelector_BayerBG12](#),
[PixelFormatInfoSelector_BayerBG12p](#),
[PixelFormatInfoSelector_BayerBG16](#),
[PixelFormatInfoSelector_BayerGB8](#),
[PixelFormatInfoSelector_BayerGB10](#),
[PixelFormatInfoSelector_BayerGB10p](#),
[PixelFormatInfoSelector_BayerGB12](#),
[PixelFormatInfoSelector_BayerGB12p](#),
[PixelFormatInfoSelector_BayerGB16](#),
[PixelFormatInfoSelector_BayerGR8](#),
[PixelFormatInfoSelector_BayerGR10](#),
[PixelFormatInfoSelector_BayerGR10p](#),
[PixelFormatInfoSelector_BayerGR12](#),
[PixelFormatInfoSelector_BayerGR12p](#),
[PixelFormatInfoSelector_BayerGR16](#),
[PixelFormatInfoSelector_BayerRG8](#),
[PixelFormatInfoSelector_BayerRG10](#),
[PixelFormatInfoSelector_BayerRG10p](#),
[PixelFormatInfoSelector_BayerRG12](#),
[PixelFormatInfoSelector_BayerRG12p](#),
[PixelFormatInfoSelector_BayerRG16](#),
[PixelFormatInfoSelector_RGBa8](#),
[PixelFormatInfoSelector_RGBa10](#),
[PixelFormatInfoSelector_RGBa10p](#),
[PixelFormatInfoSelector_RGBa12](#),
[PixelFormatInfoSelector_RGBa12p](#),
[PixelFormatInfoSelector_RGBa14](#),
[PixelFormatInfoSelector_RGBa16](#),
[PixelFormatInfoSelector_RGB8](#),
[PixelFormatInfoSelector_RGB8_Planar](#),
[PixelFormatInfoSelector_RGB10](#),
[PixelFormatInfoSelector_RGB10_Planar](#),
[PixelFormatInfoSelector_RGB10p](#),
[PixelFormatInfoSelector_RGB10p32](#),
[PixelFormatInfoSelector_RGB12](#),
[PixelFormatInfoSelector_RGB12_Planar](#),
[PixelFormatInfoSelector_RGB12p](#),
[PixelFormatInfoSelector_RGB14](#),
[PixelFormatInfoSelector_RGB16](#),
[PixelFormatInfoSelector_RGB16s](#),
[PixelFormatInfoSelector_RGB32f](#),
[PixelFormatInfoSelector_RGB16_Planar](#),
[PixelFormatInfoSelector_RGB565p](#),
[PixelFormatInfoSelector_BGRa8](#),
[PixelFormatInfoSelector_BGRa10](#),
[PixelFormatInfoSelector_BGRa10p](#),
[PixelFormatInfoSelector_BGRa12](#),
[PixelFormatInfoSelector_BGRa12p](#),
[PixelFormatInfoSelector_BGRa14](#),
[PixelFormatInfoSelector_BGRa16](#),

[PixelFormatInfoSelector_RGBa32f](#),
[PixelFormatInfoSelector_BGR8](#),
[PixelFormatInfoSelector_BGR10](#),
[PixelFormatInfoSelector_BGR10p](#),
[PixelFormatInfoSelector_BGR12](#),
[PixelFormatInfoSelector_BGR12p](#),
[PixelFormatInfoSelector_BGR14](#),
[PixelFormatInfoSelector_BGR16](#),
[PixelFormatInfoSelector_BGR565p](#),
[PixelFormatInfoSelector_R8](#),
[PixelFormatInfoSelector_R10](#),
[PixelFormatInfoSelector_R12](#),
[PixelFormatInfoSelector_R16](#),
[PixelFormatInfoSelector_G8](#),
[PixelFormatInfoSelector_G10](#),
[PixelFormatInfoSelector_G12](#),
[PixelFormatInfoSelector_G16](#),
[PixelFormatInfoSelector_B8](#),
[PixelFormatInfoSelector_B10](#),
[PixelFormatInfoSelector_B12](#),
[PixelFormatInfoSelector_B16](#),
[PixelFormatInfoSelector_Coord3D_ABC8](#),
[PixelFormatInfoSelector_Coord3D_ABC8_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC10p](#),
[PixelFormatInfoSelector_Coord3D_ABC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC12p](#),
[PixelFormatInfoSelector_Coord3D_ABC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC16](#),
[PixelFormatInfoSelector_Coord3D_ABC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC32f](#),
[PixelFormatInfoSelector_Coord3D_ABC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC8](#),
[PixelFormatInfoSelector_Coord3D_AC8_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC10p](#),
[PixelFormatInfoSelector_Coord3D_AC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC12p](#),
[PixelFormatInfoSelector_Coord3D_AC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC16](#),
[PixelFormatInfoSelector_Coord3D_AC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC32f](#),
[PixelFormatInfoSelector_Coord3D_AC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_A8](#),
[PixelFormatInfoSelector_Coord3D_A10p](#),
[PixelFormatInfoSelector_Coord3D_A12p](#),
[PixelFormatInfoSelector_Coord3D_A16](#),
[PixelFormatInfoSelector_Coord3D_A32f](#),
[PixelFormatInfoSelector_Coord3D_B8](#),
[PixelFormatInfoSelector_Coord3D_B10p](#),
[PixelFormatInfoSelector_Coord3D_B12p](#),
[PixelFormatInfoSelector_Coord3D_B16](#),
[PixelFormatInfoSelector_Coord3D_B32f](#),
[PixelFormatInfoSelector_Coord3D_C8](#),
[PixelFormatInfoSelector_Coord3D_C10p](#),
[PixelFormatInfoSelector_Coord3D_C12p](#),
[PixelFormatInfoSelector_Coord3D_C16](#),
[PixelFormatInfoSelector_Coord3D_C32f](#),
[PixelFormatInfoSelector_Confidence1](#),
[PixelFormatInfoSelector_Confidence1p](#),

[PixelFormatInfoSelector_Confidence8](#),
[PixelFormatInfoSelector_Confidence16](#),
[PixelFormatInfoSelector_Confidence32f](#),
[PixelFormatInfoSelector_BiColorBGRG8](#),
[PixelFormatInfoSelector_BiColorBGRG10](#),
[PixelFormatInfoSelector_BiColorBGRG10p](#),
[PixelFormatInfoSelector_BiColorBGRG12](#),
[PixelFormatInfoSelector_BiColorBGRG12p](#),
[PixelFormatInfoSelector_BiColorRGBG8](#),
[PixelFormatInfoSelector_BiColorRGBG10](#),
[PixelFormatInfoSelector_BiColorRGBG10p](#),
[PixelFormatInfoSelector_BiColorRGBG12](#),
[PixelFormatInfoSelector_BiColorRGBG12p](#),
[PixelFormatInfoSelector_SCF1WBWG8](#),
[PixelFormatInfoSelector_SCF1WBWG10](#),
[PixelFormatInfoSelector_SCF1WBWG10p](#),
[PixelFormatInfoSelector_SCF1WBWG12](#),
[PixelFormatInfoSelector_SCF1WBWG12p](#),
[PixelFormatInfoSelector_SCF1WBWG14](#),
[PixelFormatInfoSelector_SCF1WBWG16](#),
[PixelFormatInfoSelector_SCF1WGWB8](#),
[PixelFormatInfoSelector_SCF1WGWB10](#),
[PixelFormatInfoSelector_SCF1WGWB10p](#),
[PixelFormatInfoSelector_SCF1WGWB12](#),
[PixelFormatInfoSelector_SCF1WGWB12p](#),
[PixelFormatInfoSelector_SCF1WGWB14](#),
[PixelFormatInfoSelector_SCF1WGWB16](#),
[PixelFormatInfoSelector_SCF1WGWR8](#),
[PixelFormatInfoSelector_SCF1WGWR10](#),
[PixelFormatInfoSelector_SCF1WGWR10p](#),
[PixelFormatInfoSelector_SCF1WGWR12](#),
[PixelFormatInfoSelector_SCF1WGWR12p](#),
[PixelFormatInfoSelector_SCF1WGWR14](#),
[PixelFormatInfoSelector_SCF1WGWR16](#),
[PixelFormatInfoSelector_SCF1WRWG8](#),
[PixelFormatInfoSelector_SCF1WRWG10](#),
[PixelFormatInfoSelector_SCF1WRWG10p](#),
[PixelFormatInfoSelector_SCF1WRWG12](#),
[PixelFormatInfoSelector_SCF1WRWG12p](#),
[PixelFormatInfoSelector_SCF1WRWG14](#),
[PixelFormatInfoSelector_SCF1WRWG16](#),
[PixelFormatInfoSelector_YCbCr8](#),
[PixelFormatInfoSelector_YCbCr8_CbYCr](#),
[PixelFormatInfoSelector_YCbCr10_CbYCr](#),
[PixelFormatInfoSelector_YCbCr10p_CbYCr](#),
[PixelFormatInfoSelector_YCbCr12_CbYCr](#),
[PixelFormatInfoSelector_YCbCr12p_CbYCr](#),
[PixelFormatInfoSelector_YCbCr411_8](#),
[PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY](#),
[PixelFormatInfoSelector_YCbCr422_8](#),
[PixelFormatInfoSelector_YCbCr422_8_CbYCrY](#),
[PixelFormatInfoSelector_YCbCr422_10](#),
[PixelFormatInfoSelector_YCbCr422_10_CbYCrY](#),
[PixelFormatInfoSelector_YCbCr422_10p](#),
[PixelFormatInfoSelector_YCbCr422_10p_CbYCrY](#),
[PixelFormatInfoSelector_YCbCr422_12](#),
[PixelFormatInfoSelector_YCbCr422_12_CbYCrY](#),
[PixelFormatInfoSelector_YCbCr422_12p](#),

```

PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

```

- enum DeinterlacingEnums {
 - Deinterlacing_Off,
 - Deinterlacing_LineDuplication,
 - Deinterlacing_Weave,
 - NUM_DEINTERLACING }
- enum ImageCompressionRateOptionEnums {
 - ImageCompressionRateOption_FixBitrate,
 - ImageCompressionRateOption_FixQuality,

- NUM_IMAGECOMPRESSIONRATEOPTION }
- enum ImageCompressionJPEGFormatOptionEnums {
ImageCompressionJPEGFormatOption_Lossless,
ImageCompressionJPEGFormatOption_BaselineStandard,
ImageCompressionJPEGFormatOption_BaselineOptimized,
ImageCompressionJPEGFormatOption_Progressive,
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {
AcquisitionStatusSelector_AcquisitionTriggerWait,
AcquisitionStatusSelector_AcquisitionActive,
AcquisitionStatusSelector_AcquisitionTransfer,
AcquisitionStatusSelector_FrameTriggerWait,
AcquisitionStatusSelector_FrameActive,
AcquisitionStatusSelector_ExposureActive,
NUM_ACQUISITIONSTATUSSELECTOR }
- enum ExposureTimeModeEnums {
ExposureTimeMode_Common,
ExposureTimeMode_Individual,
NUM_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {
ExposureTimeSelector_Common,
ExposureTimeSelector_Red,
ExposureTimeSelector_Green,
ExposureTimeSelector_Blue,
ExposureTimeSelector_Cyan,
ExposureTimeSelector_Magenta,
ExposureTimeSelector_Yellow,
ExposureTimeSelector_Infrared,
ExposureTimeSelector_Ultraviolet,
ExposureTimeSelector_Stage1,
ExposureTimeSelector_Stage2,
NUM_EXPOSURETIMESELECTOR }
- enum GainAutoBalanceEnums {
GainAutoBalance_Off,
GainAutoBalance_Once,
GainAutoBalance_Continuous,
NUM_GAINAUTOBALANCE }
- enum BlackLevelAutoEnums {
BlackLevelAuto_Off,
BlackLevelAuto_Once,
BlackLevelAuto_Continuous,
NUM_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {
BlackLevelAutoBalance_Off,
BlackLevelAutoBalance_Once,
BlackLevelAutoBalance_Continuous,
NUM_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {
WhiteClipSelector_All,
WhiteClipSelector_Red,
WhiteClipSelector_Green,
WhiteClipSelector_Blue,
WhiteClipSelector_Y,
WhiteClipSelector_U,
WhiteClipSelector_V,
WhiteClipSelector_Tap1,
WhiteClipSelector_Tap2,
NUM_WHITECLIPSELECTOR }

- enum [TimerSelectorEnums](#) {
 [TimerSelector_Timer0](#),
 [TimerSelector_Timer1](#),
 [TimerSelector_Timer2](#),
 [NUM_TIMERSELECTOR](#) }
- enum [TimerStatusEnums](#) {
 [TimerStatus_TimerIdle](#),
 [TimerStatus_TimerTriggerWait](#),
 [TimerStatus_TimerActive](#),
 [TimerStatus_TimerCompleted](#),
 [NUM_TIMERSTATUS](#) }
- enum [TimerTriggerSourceEnums](#) {
 [TimerTriggerSource_Off](#),
 [TimerTriggerSource_AcquisitionTrigger](#),
 [TimerTriggerSource_AcquisitionStart](#),
 [TimerTriggerSource_AcquisitionEnd](#),
 [TimerTriggerSource_FrameTrigger](#),
 [TimerTriggerSource_FrameStart](#),
 [TimerTriggerSource_FrameEnd](#),
 [TimerTriggerSource_FrameBurstStart](#),
 [TimerTriggerSource_FrameBurstEnd](#),
 [TimerTriggerSource_LineTrigger](#),
 [TimerTriggerSource_LineStart](#),
 [TimerTriggerSource_LineEnd](#),
 [TimerTriggerSource_ExposureStart](#),
 [TimerTriggerSource_ExposureEnd](#),
 [TimerTriggerSource_Line0](#),
 [TimerTriggerSource_Line1](#),
 [TimerTriggerSource_Line2](#),
 [TimerTriggerSource_UserOutput0](#),
 [TimerTriggerSource_UserOutput1](#),
 [TimerTriggerSource_UserOutput2](#),
 [TimerTriggerSource_Counter0Start](#),
 [TimerTriggerSource_Counter1Start](#),
 [TimerTriggerSource_Counter2Start](#),
 [TimerTriggerSource_Counter0End](#),
 [TimerTriggerSource_Counter1End](#),
 [TimerTriggerSource_Counter2End](#),
 [TimerTriggerSource_Timer0Start](#),
 [TimerTriggerSource_Timer1Start](#),
 [TimerTriggerSource_Timer2Start](#),
 [TimerTriggerSource_Timer0End](#),
 [TimerTriggerSource_Timer1End](#),
 [TimerTriggerSource_Timer2End](#),
 [TimerTriggerSource_Encoder0](#),
 [TimerTriggerSource_Encoder1](#),
 [TimerTriggerSource_Encoder2](#),
 [TimerTriggerSource_SoftwareSignal0](#),
 [TimerTriggerSource_SoftwareSignal1](#),
 [TimerTriggerSource_SoftwareSignal2](#),
 [TimerTriggerSource_Action0](#),
 [TimerTriggerSource_Action1](#),
 [TimerTriggerSource_Action2](#),
 [TimerTriggerSource_LinkTrigger0](#),
 [TimerTriggerSource_LinkTrigger1](#),
 [TimerTriggerSource_LinkTrigger2](#),
 [NUM_TIMERTRIGGERSOURCE](#) }
- enum [TimerTriggerActivationEnums](#) {

```
TimerTriggerActivation_RisingEdge,  
TimerTriggerActivation_FallingEdge,  
TimerTriggerActivation_AnyEdge,  
TimerTriggerActivation_LevelHigh,  
TimerTriggerActivation_LevelLow,  
NUM_TIMERTRIGGERACTIVATION }  
  
• enum EncoderSelectorEnums {  
EncoderSelector_Encoder0,  
EncoderSelector_Encoder1,  
EncoderSelector_Encoder2,  
NUM_ENCODERSELECTOR }  
  
• enum EncoderSourceAEnums {  
EncoderSourceA_Off,  
EncoderSourceA_Line0,  
EncoderSourceA_Line1,  
EncoderSourceA_Line2,  
NUM_ENCODERSOURCEA }  
  
• enum EncoderSourceBEnums {  
EncoderSourceB_Off,  
EncoderSourceB_Line0,  
EncoderSourceB_Line1,  
EncoderSourceB_Line2,  
NUM_ENCODERSOURCEB }  
  
• enum EncoderModeEnums {  
EncoderMode_FourPhase,  
EncoderMode_HighResolution,  
NUM_ENCODERMODE }  
  
• enum EncoderOutputModeEnums {  
EncoderOutputMode_Off,  
EncoderOutputMode_PositionUp,  
EncoderOutputMode_PositionDown,  
EncoderOutputMode_DirectionUp,  
EncoderOutputMode_DirectionDown,  
EncoderOutputMode_Motion,  
NUM_ENCODEROUTPUTMODE }  
  
• enum EncoderStatusEnums {  
EncoderStatus_EncoderUp,  
EncoderStatus_EncoderDown,  
EncoderStatus_EncoderIdle,  
EncoderStatus_EncoderStatic,  
NUM_ENCODERSTATUS }  
  
• enum EncoderResetSourceEnums {  
EncoderResetSource_Off,  
EncoderResetSource_AcquisitionTrigger,  
EncoderResetSource_AcquisitionStart,  
EncoderResetSource_AcquisitionEnd,  
EncoderResetSource_FrameTrigger,  
EncoderResetSource_FrameStart,  
EncoderResetSource_FrameEnd,  
EncoderResetSource_ExposureStart,  
EncoderResetSource_ExposureEnd,  
EncoderResetSource_Line0,  
EncoderResetSource_Line1,  
EncoderResetSource_Line2,  
EncoderResetSource_Counter0Start,  
EncoderResetSource_Counter1Start,  
EncoderResetSource_Counter2Start,  
EncoderResetSource_Counter0End,
```

```

EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }

• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }

• enum TransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }

• enum TransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,

```

```
NUM_TRANSFERTRIGGERSELECTOR }

• enum TransferTriggerModeEnums {
    TransferTriggerMode_Off,
    TransferTriggerMode_On,
    NUM_TRANSFERTRIGGERMODE }

• enum TransferTriggerSourceEnums {
    TransferTriggerSource_Line0,
    TransferTriggerSource_Line1,
    TransferTriggerSource_Line2,
    TransferTriggerSource_Counter0Start,
    TransferTriggerSource_Counter1Start,
    TransferTriggerSource_Counter2Start,
    TransferTriggerSource_Counter0End,
    TransferTriggerSource_Counter1End,
    TransferTriggerSource_Counter2End,
    TransferTriggerSource_Timer0Start,
    TransferTriggerSource_Timer1Start,
    TransferTriggerSource_Timer2Start,
    TransferTriggerSource_Timer0End,
    TransferTriggerSource_Timer1End,
    TransferTriggerSource_Timer2End,
    TransferTriggerSource_SoftwareSignal0,
    TransferTriggerSource_SoftwareSignal1,
    TransferTriggerSource_SoftwareSignal2,
    TransferTriggerSource_Action0,
    TransferTriggerSource_Action1,
    TransferTriggerSource_Action2,
    NUM_TRANSFERTRIGGERSOURCE }

• enum TransferTriggerActivationEnums {
    TransferTriggerActivation_RisingEdge,
    TransferTriggerActivation_FallingEdge,
    TransferTriggerActivation_AnyEdge,
    TransferTriggerActivation_LevelHigh,
    TransferTriggerActivation_LevelLow,
    NUM_TRANSFERTRIGGERACTIVATION }

• enum TransferStatusSelectorEnums {
    TransferStatusSelector_Streaming,
    TransferStatusSelector_Paused,
    TransferStatusSelector_Stopping,
    TransferStatusSelector_Stopped,
    TransferStatusSelector_QueueOverflow,
    NUM_TRANSFERSTATUSSELECTOR }

• enum TransferComponentSelectorEnums {
    TransferComponentSelector_Red,
    TransferComponentSelector_Green,
    TransferComponentSelector_Blue,
    TransferComponentSelector_All,
    NUM_TRANSFERCOMPONENTSELECTOR }

• enum Scan3dDistanceUnitEnums {
    Scan3dDistanceUnit_Millimeter,
    Scan3dDistanceUnit_Inch,
    NUM_SCAN3DDISTANCEUNIT }

• enum Scan3dCoordinateSystemEnums {
    Scan3dCoordinateSystem_Cartesian,
    Scan3dCoordinateSystem_Spherical,
    Scan3dCoordinateSystem_Cylindrical,
    NUM_SCAN3DCOORDINATESYSTEM }
```

- `enum Scan3dOutputModeEnums {`
`Scan3dOutputMode_UncalibratedC,`
`Scan3dOutputMode_CalibratedABC_Grid,`
`Scan3dOutputMode_CalibratedABC_PointCloud,`
`Scan3dOutputMode_CalibratedAC,`
`Scan3dOutputMode_CalibratedAC_Linescan,`
`Scan3dOutputMode_CalibratedC,`
`Scan3dOutputMode_CalibratedC_Linescan,`
`Scan3dOutputMode_RectifiedC,`
`Scan3dOutputMode_RectifiedC_Linescan,`
`Scan3dOutputMode_DisparityC,`
`Scan3dOutputMode_DisparityC_Linescan,`
`NUM_SCAN3DOUTPUTMODE }`
- `enum Scan3dCoordinateSystemReferenceEnums {`
`Scan3dCoordinateSystemReference_Anchor,`
`Scan3dCoordinateSystemReference_Transformed,`
`NUM_SCAN3DCOORDINATESYSTEMREFERENCE }`
- `enum Scan3dCoordinateSelectorEnums {`
`Scan3dCoordinateSelector_CoordinateA,`
`Scan3dCoordinateSelector_CoordinateB,`
`Scan3dCoordinateSelector_CoordinateC,`
`NUM_SCAN3DCOORDINATESELECTOR }`
- `enum Scan3dCoordinateTransformSelectorEnums {`
`Scan3dCoordinateTransformSelector_RotationX,`
`Scan3dCoordinateTransformSelector_RotationY,`
`Scan3dCoordinateTransformSelector_RotationZ,`
`Scan3dCoordinateTransformSelector_TranslationX,`
`Scan3dCoordinateTransformSelector_TranslationY,`
`Scan3dCoordinateTransformSelector_TranslationZ,`
`NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }`
- `enum Scan3dCoordinateReferenceSelectorEnums {`
`Scan3dCoordinateReferenceSelector_RotationX,`
`Scan3dCoordinateReferenceSelector_RotationY,`
`Scan3dCoordinateReferenceSelector_RotationZ,`
`Scan3dCoordinateReferenceSelector_TranslationX,`
`Scan3dCoordinateReferenceSelector_TranslationY,`
`Scan3dCoordinateReferenceSelector_TranslationZ,`
`NUM_SCAN3DCOORDINATEREFERENCESELECTOR }`
- `enum ChunkImageComponentEnums {`
`ChunkImageComponent_Intensity,`
`ChunkImageComponent_Color,`
`ChunkImageComponent_Infrared,`
`ChunkImageComponent_Ultraviolet,`
`ChunkImageComponent_Range,`
`ChunkImageComponent_Disparity,`
`ChunkImageComponent_Confidence,`
`ChunkImageComponent_Scatter,`
`NUM_CHUNKIMAGECOMPONENT }`
- `enum ChunkCounterSelectorEnums {`
`ChunkCounterSelector_Counter0,`
`ChunkCounterSelector_Counter1,`
`ChunkCounterSelector_Counter2,`
`NUM_CHUNKCOUNTERSELECTOR }`
- `enum ChunkTimerSelectorEnums {`
`ChunkTimerSelector_Timer0,`
`ChunkTimerSelector_Timer1,`
`ChunkTimerSelector_Timer2,`
`NUM_CHUNKTIMERSELECTOR }`

- enum [ChunkEncoderSelectorEnums](#) {
[ChunkEncoderSelector_Encoder0](#),
[ChunkEncoderSelector_Encoder1](#),
[ChunkEncoderSelector_Encoder2](#),
[NUM_CHUNKENCODERSELECTOR](#) }
- enum [ChunkEncoderStatusEnums](#) {
[ChunkEncoderStatus_EncoderUp](#),
[ChunkEncoderStatus_EncoderDown](#),
[ChunkEncoderStatus_EncoderIdle](#),
[ChunkEncoderStatus_EncoderStatic](#),
[NUM_CHUNKENCODERSTATUS](#) }
- enum [ChunkExposureTimeSelectorEnums](#) {
[ChunkExposureTimeSelector_Common](#),
[ChunkExposureTimeSelector_Red](#),
[ChunkExposureTimeSelector_Green](#),
[ChunkExposureTimeSelector_Blue](#),
[ChunkExposureTimeSelector_Cyan](#),
[ChunkExposureTimeSelector_Magenta](#),
[ChunkExposureTimeSelector_Yellow](#),
[ChunkExposureTimeSelector_Infrared](#),
[ChunkExposureTimeSelector_Ultraviolet](#),
[ChunkExposureTimeSelector_Stage1](#),
[ChunkExposureTimeSelector_Stage2](#),
[NUM_CHUNKEXPOSURETIMESELECTOR](#) }
- enum [ChunkSourceIDEnums](#) {
[ChunkSourceID_Source0](#),
[ChunkSourceID_Source1](#),
[ChunkSourceID_Source2](#),
[NUM_CHUNKSOURCEID](#) }
- enum [ChunkRegionIDEnums](#) {
[ChunkRegionID_Region0](#),
[ChunkRegionID_Region1](#),
[ChunkRegionID_Region2](#),
[NUM_CHUNKREGIONID](#) }
- enum [ChunkTransferStreamIDEnums](#) {
[ChunkTransferStreamID_Stream0](#),
[ChunkTransferStreamID_Stream1](#),
[ChunkTransferStreamID_Stream2](#),
[ChunkTransferStreamID_Stream3](#),
[NUM_CHUNKTRANSFERSTREAMID](#) }
- enum [ChunkScan3dDistanceUnitEnums](#) {
[ChunkScan3dDistanceUnit_Millimeter](#),
[ChunkScan3dDistanceUnit_Inch](#),
[NUM_CHUNKSCAN3DDISTANCEUNIT](#) }
- enum [ChunkScan3dOutputModeEnums](#) {
[ChunkScan3dOutputMode_UncalibratedC](#),
[ChunkScan3dOutputMode_CalibratedABC_Grid](#),
[ChunkScan3dOutputMode_CalibratedABC_PointCloud](#),
[ChunkScan3dOutputMode_CalibratedAC](#),
[ChunkScan3dOutputMode_CalibratedAC_Linescan](#),
[ChunkScan3dOutputMode_CalibratedC](#),
[ChunkScan3dOutputMode_CalibratedC_Linescan](#),
[ChunkScan3dOutputMode_RectifiedC](#),
[ChunkScan3dOutputMode_RectifiedC_Linescan](#),
[ChunkScan3dOutputMode_DisparityC](#),
[ChunkScan3dOutputMode_DisparityC_Linescan](#),
[NUM_CHUNKSCAN3DOUTPUTMODE](#) }

- `enum ChunkScan3dCoordinateSystemEnums {`
`ChunkScan3dCoordinateSystem_Cartesian,`
`ChunkScan3dCoordinateSystem_Spherical,`
`ChunkScan3dCoordinateSystem_Cylindrical,`
`NUM_CHUNKSCAN3DCOORDINATESYSTEM }`
- `enum ChunkScan3dCoordinateSystemReferenceEnums {`
`ChunkScan3dCoordinateSystemReference_Anchor,`
`ChunkScan3dCoordinateSystemReference_Transformed,`
`NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }`
- `enum ChunkScan3dCoordinateSelectorEnums {`
`ChunkScan3dCoordinateSelector_CoordinateA,`
`ChunkScan3dCoordinateSelector_CoordinateB,`
`ChunkScan3dCoordinateSelector_CoordinateC,`
`NUM_CHUNKSCAN3DCOORDINATESELECTOR }`
- `enum ChunkScan3dCoordinateTransformSelectorEnums {`
`ChunkScan3dCoordinateTransformSelector_RotationX,`
`ChunkScan3dCoordinateTransformSelector_RotationY,`
`ChunkScan3dCoordinateTransformSelector_RotationZ,`
`ChunkScan3dCoordinateTransformSelector_TranslationX,`
`ChunkScan3dCoordinateTransformSelector_TranslationY,`
`ChunkScan3dCoordinateTransformSelector_TranslationZ,`
`NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }`
- `enum ChunkScan3dCoordinateReferenceSelectorEnums {`
`ChunkScan3dCoordinateReferenceSelector_RotationX,`
`ChunkScan3dCoordinateReferenceSelector_RotationY,`
`ChunkScan3dCoordinateReferenceSelector_RotationZ,`
`ChunkScan3dCoordinateReferenceSelector_TranslationX,`
`ChunkScan3dCoordinateReferenceSelector_TranslationY,`
`ChunkScan3dCoordinateReferenceSelector_TranslationZ,`
`NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }`
- `enum DeviceTapGeometryEnums {`
`DeviceTapGeometry_Geometry_1X_1Y,`
`DeviceTapGeometry_Geometry_1X2_1Y,`
`DeviceTapGeometry_Geometry_1X2_1Y2,`
`DeviceTapGeometry_Geometry_2X_1Y,`
`DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,`
`DeviceTapGeometry_Geometry_2XE_1Y2,`
`DeviceTapGeometry_Geometry_2XM_1Y,`
`DeviceTapGeometry_Geometry_2XM_1Y2,`
`DeviceTapGeometry_Geometry_1X_1Y2,`
`DeviceTapGeometry_Geometry_1X_2YE,`
`DeviceTapGeometry_Geometry_1X3_1Y,`
`DeviceTapGeometry_Geometry_3X_1Y,`
`DeviceTapGeometry_Geometry_1X,`
`DeviceTapGeometry_Geometry_1X2,`
`DeviceTapGeometry_Geometry_2X,`
`DeviceTapGeometry_Geometry_2XE,`
`DeviceTapGeometry_Geometry_2XM,`
`DeviceTapGeometry_Geometry_1X3,`
`DeviceTapGeometry_Geometry_3X,`
`DeviceTapGeometry_Geometry_1X4_1Y,`
`DeviceTapGeometry_Geometry_4X_1Y,`
`DeviceTapGeometry_Geometry_2X2_1Y,`
`DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,`
`DeviceTapGeometry_Geometry_1X2_2YE,`
`DeviceTapGeometry_Geometry_2X_2YE,`
`DeviceTapGeometry_Geometry_2XE_2YE,`
`DeviceTapGeometry_Geometry_2XM_2YE,`


```

DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

• enum GevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }

• enum GevGVCPExtendedStatusCodesSelectorEnums {
    GevGVCPExtendedStatusCodesSelector_Version1_1,
    GevGVCPExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum GevGVSPEExtendedIDModeEnums {
    GevGVSPEExtendedIDMode_Off,
    GevGVSPEExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum ClConfigurationEnums {
    ClConfiguration_Base,
    ClConfiguration_Medium,
    ClConfiguration_Full,
    ClConfiguration_DualBase,
    ClConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum ClTimeSlotsCountEnums {
    ClTimeSlotsCount_One,
    ClTimeSlotsCount_Two,
    ClTimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }

```

- `enum CxpLinkConfigurationStatusEnums {`
 - `CxpLinkConfigurationStatus_None,`
 - `CxpLinkConfigurationStatus_Pending,`
 - `CxpLinkConfigurationStatus_CXP1_X1,`
 - `CxpLinkConfigurationStatus_CXP2_X1,`
 - `CxpLinkConfigurationStatus_CXP3_X1,`
 - `CxpLinkConfigurationStatus_CXP5_X1,`
 - `CxpLinkConfigurationStatus_CXP6_X1,`
 - `CxpLinkConfigurationStatus_CXP1_X2,`
 - `CxpLinkConfigurationStatus_CXP2_X2,`
 - `CxpLinkConfigurationStatus_CXP3_X2,`
 - `CxpLinkConfigurationStatus_CXP5_X2,`
 - `CxpLinkConfigurationStatus_CXP6_X2,`
 - `CxpLinkConfigurationStatus_CXP1_X3,`
 - `CxpLinkConfigurationStatus_CXP2_X3,`
 - `CxpLinkConfigurationStatus_CXP3_X3,`
 - `CxpLinkConfigurationStatus_CXP5_X3,`
 - `CxpLinkConfigurationStatus_CXP6_X3,`
 - `CxpLinkConfigurationStatus_CXP1_X4,`
 - `CxpLinkConfigurationStatus_CXP2_X4,`
 - `CxpLinkConfigurationStatus_CXP3_X4,`
 - `CxpLinkConfigurationStatus_CXP5_X4,`
 - `CxpLinkConfigurationStatus_CXP6_X4,`
 - `CxpLinkConfigurationStatus_CXP1_X5,`
 - `CxpLinkConfigurationStatus_CXP2_X5,`
 - `CxpLinkConfigurationStatus_CXP3_X5,`
 - `CxpLinkConfigurationStatus_CXP5_X5,`
 - `CxpLinkConfigurationStatus_CXP6_X5,`
 - `CxpLinkConfigurationStatus_CXP1_X6,`
 - `CxpLinkConfigurationStatus_CXP2_X6,`
 - `CxpLinkConfigurationStatus_CXP3_X6,`
 - `CxpLinkConfigurationStatus_CXP5_X6,`
 - `CxpLinkConfigurationStatus_CXP6_X6,`
 - `NUM_CXPLINKCONFIGURATIONSTATUS }`
- `enum CxpLinkConfigurationPreferredEnums {`
 - `CxpLinkConfigurationPreferred_CXP1_X1,`
 - `CxpLinkConfigurationPreferred_CXP2_X1,`
 - `CxpLinkConfigurationPreferred_CXP3_X1,`
 - `CxpLinkConfigurationPreferred_CXP5_X1,`
 - `CxpLinkConfigurationPreferred_CXP6_X1,`
 - `CxpLinkConfigurationPreferred_CXP1_X2,`
 - `CxpLinkConfigurationPreferred_CXP2_X2,`
 - `CxpLinkConfigurationPreferred_CXP3_X2,`
 - `CxpLinkConfigurationPreferred_CXP5_X2,`
 - `CxpLinkConfigurationPreferred_CXP6_X2,`
 - `CxpLinkConfigurationPreferred_CXP1_X3,`
 - `CxpLinkConfigurationPreferred_CXP2_X3,`
 - `CxpLinkConfigurationPreferred_CXP3_X3,`
 - `CxpLinkConfigurationPreferred_CXP5_X3,`
 - `CxpLinkConfigurationPreferred_CXP6_X3,`
 - `CxpLinkConfigurationPreferred_CXP1_X4,`
 - `CxpLinkConfigurationPreferred_CXP2_X4,`
 - `CxpLinkConfigurationPreferred_CXP3_X4,`
 - `CxpLinkConfigurationPreferred_CXP5_X4,`
 - `CxpLinkConfigurationPreferred_CXP6_X4,`
 - `CxpLinkConfigurationPreferred_CXP1_X5,`
 - `CxpLinkConfigurationPreferred_CXP2_X5,`
 - `CxpLinkConfigurationPreferred_CXP3_X5,`

- CxpLinkConfigurationPreferred_CXP5_X5,
 - CxpLinkConfigurationPreferred_CXP6_X5,
 - CxpLinkConfigurationPreferred_CXP1_X6,
 - CxpLinkConfigurationPreferred_CXP2_X6,
 - CxpLinkConfigurationPreferred_CXP3_X6,
 - CxpLinkConfigurationPreferred_CXP5_X6,
 - CxpLinkConfigurationPreferred_CXP6_X6,
 - NUM_CXPLINKCONFIGURATIONPREFERRED }
- enum CxpLinkConfigurationEnums {
 - CxpLinkConfiguration_Auto,
 - CxpLinkConfiguration_CXP1_X1,
 - CxpLinkConfiguration_CXP2_X1,
 - CxpLinkConfiguration_CXP3_X1,
 - CxpLinkConfiguration_CXP5_X1,
 - CxpLinkConfiguration_CXP6_X1,
 - CxpLinkConfiguration_CXP1_X2,
 - CxpLinkConfiguration_CXP2_X2,
 - CxpLinkConfiguration_CXP3_X2,
 - CxpLinkConfiguration_CXP5_X2,
 - CxpLinkConfiguration_CXP6_X2,
 - CxpLinkConfiguration_CXP1_X3,
 - CxpLinkConfiguration_CXP2_X3,
 - CxpLinkConfiguration_CXP3_X3,
 - CxpLinkConfiguration_CXP5_X3,
 - CxpLinkConfiguration_CXP6_X3,
 - CxpLinkConfiguration_CXP1_X4,
 - CxpLinkConfiguration_CXP2_X4,
 - CxpLinkConfiguration_CXP3_X4,
 - CxpLinkConfiguration_CXP5_X4,
 - CxpLinkConfiguration_CXP6_X4,
 - CxpLinkConfiguration_CXP1_X5,
 - CxpLinkConfiguration_CXP2_X5,
 - CxpLinkConfiguration_CXP3_X5,
 - CxpLinkConfiguration_CXP5_X5,
 - CxpLinkConfiguration_CXP6_X5,
 - CxpLinkConfiguration_CXP1_X6,
 - CxpLinkConfiguration_CXP2_X6,
 - CxpLinkConfiguration_CXP3_X6,
 - CxpLinkConfiguration_CXP5_X6,
 - CxpLinkConfiguration_CXP6_X6,
 - NUM_CXPLINKCONFIGURATION }
- enum CxpConnectionTestModeEnums {
 - CxpConnectionTestMode_Off,
 - CxpConnectionTestMode_Mode1,
 - NUM_CXPCONNECTIONTESTMODE }
- enum CxpPoCxpStatusEnums {
 - CxpPoCxpStatus_Auto,
 - CxpPoCxpStatus_Off,
 - CxpPoCxpStatus_Tripped,
 - NUM_CXPPOCXPSTATUS }
- enum InferenceBoxType {
 - INFERENCE_BOX_TYPE_RECTANGLE = 0,
 - INFERENCE_BOX_TYPE_CIRCLE = 1,
 - INFERENCE_BOX_TYPE_ROTATED_RECTANGLE = 2 }

Inference Bounding Box Type.
- enum Error {
 - SPINNAKER_ERR_SUCCESS = 0,
 - SPINNAKER_ERR_ERROR = -1001,

```

SPINNAKER_ERR_NOT_INITIALIZED = -1002,
SPINNAKER_ERR_NOT_IMPLEMENTED = -1003,
SPINNAKER_ERR_RESOURCE_IN_USE = -1004,
SPINNAKER_ERR_ACCESS_DENIED = -1005,
SPINNAKER_ERR_INVALID_HANDLE = -1006,
SPINNAKER_ERR_INVALID_ID = -1007,
SPINNAKER_ERR_NO_DATA = -1008,
SPINNAKER_ERR_INVALID_PARAMETER = -1009,
SPINNAKER_ERR_IO = -1010,
SPINNAKER_ERR_TIMEOUT = -1011,
SPINNAKER_ERR_ABORT = -1012,
SPINNAKER_ERR_INVALID_BUFFER = -1013,
SPINNAKER_ERR_NOT_AVAILABLE = -1014,
SPINNAKER_ERR_INVALID_ADDRESS = -1015,
SPINNAKER_ERR_BUFFER_TOO_SMALL = -1016,
SPINNAKER_ERR_INVALID_INDEX = -1017,
SPINNAKER_ERR_PARSING_CHUNK_DATA = -1018,
SPINNAKER_ERR_INVALID_VALUE = -1019,
SPINNAKER_ERR_RESOURCE_EXHAUSTED = -1020,
SPINNAKER_ERR_OUT_OF_MEMORY = -1021,
SPINNAKER_ERR_BUSY = -1022,
SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT = -2001,
SPINNAKER_ERR_GENICAM_OUT_OF_RANGE = -2002,
SPINNAKER_ERR_GENICAM_PROPERTY = -2003,
SPINNAKER_ERR_GENICAM_RUN_TIME = -2004,
SPINNAKER_ERR_GENICAM_LOGICAL = -2005,
SPINNAKER_ERR_GENICAM_ACCESS = -2006,
SPINNAKER_ERR_GENICAM_TIMEOUT = -2007,
SPINNAKER_ERR_GENICAM_DYNAMIC_CAST = -2008,
SPINNAKER_ERR_GENICAM_GENERIC = -2009,
SPINNAKER_ERR_GENICAM_BAD_ALLOCATION = -2010,
SPINNAKER_ERR_IM_CONVERT = -3001,
SPINNAKER_ERR_IM_COPY = -3002,
SPINNAKER_ERR_IM_MALLOC = -3003,
SPINNAKER_ERR_IM_NOT_SUPPORTED = -3004,
SPINNAKER_ERR_IM_HISTOGRAM_RANGE = -3005,
SPINNAKER_ERR_IM_HISTOGRAM_MEAN = -3006,
SPINNAKER_ERR_IM_MIN_MAX = -3007,
SPINNAKER_ERR_IM_COLOR_CONVERSION = -3008,
SPINNAKER_ERR_IM_DECOMPRESSION = -3009,
SPINNAKER_ERR_CUSTOM_ID = -10000 }

```

Spinnaker enum definitions.

- enum EventType {


```

SPINNAKER_EVENT_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_DEVICE,
SPINNAKER_EVENT_DEVICE_SPECIFIC,
SPINNAKER_EVENT_NEW_BUFFER,
SPINNAKER_EVENT_NEW_BUFFER_LIST,
SPINNAKER_EVENT_LOGGING_EVENT,
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_UNKNOWN }

```

Event types in Spinnaker.

- enum TLPixelFormatNamespace {


```

SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,

```

```
SPINNAKER_TLPIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }
```

This enum represents the namespace in which the TL specific pixel format resides.

- enum ColorProcessingAlgorithm {
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_AVG,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL_FILTER,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS,
 SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER }

Color processing algorithms.

- enum ImageFileFormat {
 SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT = -1,
 SPINNAKER_IMAGE_FILE_FORMAT_PGM,
 SPINNAKER_IMAGE_FILE_FORMAT_PPM,
 SPINNAKER_IMAGE_FILE_FORMAT_BMP,
 SPINNAKER_IMAGE_FILE_FORMAT_JPEG,
 SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000,
 SPINNAKER_IMAGE_FILE_FORMAT_TIFF,
 SPINNAKER_IMAGE_FILE_FORMAT_PNG,
 SPINNAKER_IMAGE_FILE_FORMAT_RAW,
 SPINNAKER_IMAGE_FILE_FORMAT_JPEG12_C,
 SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum ImageStatus {
 SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR = -1,
 SPINNAKER_IMAGE_STATUS_NO_ERROR = 0,
 SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED = 1,
 SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW = 2,
 SPINNAKER_IMAGE_STATUS_MISSING_PACKETS,
 SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT,
 SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT,
 SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT,
 SPINNAKER_IMAGE_STATUS_MISSING_LEADER = 7,
 SPINNAKER_IMAGE_STATUS_MISSING_TRAILER,
 SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE = 9,
 SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT,
 SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID = 11,
 SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES = 12 }

Status of images returned from GetNextImage() or Convert() calls.

- enum StatisticsChannel {
 SPINNAKER_STATISTICS_CHANNEL_GREY,
 SPINNAKER_STATISTICS_CHANNEL_RED,
 SPINNAKER_STATISTICS_CHANNEL_GREEN,
 SPINNAKER_STATISTICS_CHANNEL_BLUE,
 SPINNAKER_STATISTICS_CHANNEL_HUE,
 SPINNAKER_STATISTICS_CHANNEL_SATURATION,
 SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS,
 SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS }

Channels that allow statistics to be calculated.

- enum SpinnakerLogLevel {
 SPINNAKER_LOG_LEVEL_OFF = -1,
 SPINNAKER_LOG_LEVEL_FATAL = 0,
 SPINNAKER_LOG_LEVEL_ALERT = 100,

```
SPINNAKER_LOG_LEVEL_CRIT = 200,
SPINNAKER_LOG_LEVEL_ERROR = 300,
SPINNAKER_LOG_LEVEL_WARN = 400,
SPINNAKER_LOG_LEVEL_NOTICE = 500,
SPINNAKER_LOG_LEVEL_INFO = 600,
SPINNAKER_LOG_LEVEL_DEBUG = 700,
SPINNAKER_LOG_LEVEL_NOTSET = 800 }
```

Log levels.

- enum `ImagePayloadType` {


```
SPINNAKER_IMAGE_PAYLOAD_TYPE_UNKNOWN = -1,
SPINNAKER_IMAGE_PAYLOAD_TYPE_IMAGE,
SPINNAKER_IMAGE_PAYLOAD_TYPE_EXTENDED_CHUNK,
SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG,
SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSLESS_COMPRESSED,
SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSY_COMPRESSED,
SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED }
```
- enum `TLPayloadType` {


```
SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN = 0,
SPINNAKER_TLPAYLOAD_TYPE_IMAGE = 1,
SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA = 2,
SPINNAKER_TLPAYLOAD_TYPE_FILE = 3,
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA = 4,
SPINNAKER_TLPAYLOAD_TYPE_JPEG = 5,
SPINNAKER_TLPAYLOAD_TYPE_JPEG2000 = 6,
SPINNAKER_TLPAYLOAD_TYPE_H264 = 7,
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY = 8,
SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC = 9,
SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART = 10,
SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID = 1000,
SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_C←
USTOM_ID + 1,
SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUST←
OM_ID + 2,
SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED = SPINNAKER_TLPAYLOAD_T←
YPE_CUSTOM_ID + 3 }
```
- enum `ActionCommandStatus` {


```
SPINNAKER_ACTION_COMMAND_STATUS_OK = 0,
SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME,
SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW = 0x8015,
SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE,
SPINNAKER_ACTION_COMMAND_STATUS_ERROR }
```

Possible Status Codes Returned from Action Command.

- enum `TIFFCompressionMethod` {


```
SPINNAKER_TIFF_COMPRESS_METHOD_NONE = 1,
SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS,
SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE,
SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE,
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3,
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4,
SPINNAKER_TIFF_COMPRESS_METHOD_LZW,
SPINNAKER_TIFF_COMPRESS_METHOD_JPEG }
```

Compression method to use for encoding TIFF images.

- enum `PixelFormatIntType` {


```
SPINNAKER_INT_TYPE_UINT8,
SPINNAKER_INT_TYPE_INT8,
SPINNAKER_INT_TYPE_UINT10,
SPINNAKER_INT_TYPE_UINT10p,
```

```
SPINNAKER_INT_TYPE_UINT10P,
SPINNAKER_INT_TYPE_UINT12,
SPINNAKER_INT_TYPE_UINT12p,
SPINNAKER_INT_TYPE_UINT12P,
SPINNAKER_INT_TYPE_UINT14,
SPINNAKER_INT_TYPE_UINT16,
SPINNAKER_INT_TYPE_INT16,
SPINNAKER_INT_TYPE_FLOAT32,
SPINNAKER_INT_TYPE_UNKNOWN }
```

Possible integer types and packing used in a pixel format.

- enum `BufferOwnership` {
`SPINNAKER_BUFFER_OWNERSHIP_SYSTEM`,
`SPINNAKER_BUFFER_OWNERSHIP_USER` }
- enum `ImageScalingAlgorithm` { `SPINNAKER_IMAGE_SCALING_ALGORITHM_NEAREST_NEIGHBOR` }

Image scaling algorithms.

- enum `SourceDataRange` {
`SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE`,
`SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_DATA_RANGE`,
`SPINNAKER_SOURCE_DATA_RANGE_IMAGE_MIN_ABSOLUTE_MAX`,
`SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_MIN_IMAGE_MAX` }

Image normalization source data options.

- enum `HeatmapColor` {
`SPINNAKER_HEATMAP_COLOR_BLACK` = 1,
`SPINNAKER_HEATMAP_COLOR_BLUE` = 2,
`SPINNAKER_HEATMAP_COLOR_CYAN` = 3,
`SPINNAKER_HEATMAP_COLOR_GREEN` = 4,
`SPINNAKER_HEATMAP_COLOR_YELLOW` = 5,
`SPINNAKER_HEATMAP_COLOR_RED` = 6,
`SPINNAKER_HEATMAP_COLOR_WHITE` = 7 }

Color specifiers for the heatmap color gradient.

- enum `PolarizationQuadrant` {
`SPINNAKER_POLARIZATION_QUADRANT_I0`,
`SPINNAKER_POLARIZATION_QUADRANT_I45`,
`SPINNAKER_POLARIZATION_QUADRANT_I90`,
`SPINNAKER_POLARIZATION_QUADRANT_I135` }

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

- enum `CCMColorTemperature` {
`SPINNAKER_CCM_COLOR_TEMP_TUNGSTEN_2800K`,
`SPINNAKER_CCM_COLOR_TEMP_WARM_FLUORESCENT_3000K`,
`SPINNAKER_CCM_COLOR_TEMP_COOL_FLUORESCENT_4000K`,
`SPINNAKER_CCM_COLOR_TEMP_SUNNY_5000K`,
`SPINNAKER_CCM_COLOR_TEMP_CLOUDY_6500K`,
`SPINNAKER_CCM_COLOR_TEMP_SHADE_8000K`,
`SPINNAKER_CCM_COLOR_TEMP_GENERAL` }
- enum `CCMType` {
`SPINNAKER_CCM_TYPE_LINEAR`,
`SPINNAKER_CCM_TYPE_ADVANCED` }
- enum `CCMSensor` { `SPINNAKER_CCM_SENSOR_IMX250` }
- enum `CCMColorSpace` {
`SPINNAKER_CCM_COLOR_SPACE_OFF`,
`SPINNAKER_CCM_COLOR_SPACE_SRGB` }
- enum `CCMApplication` {
`SPINNAKER_CCM_APPLICATION_GENERIC`,
`SPINNAKER_CCM_APPLICATION_MICROSCOPY` }
- enum `StreamTypeEnum` {
`StreamType_GigEVision`,

```
StreamType_CameraLink,
StreamType_CameraLinkHS,
StreamType_CoaXPress,
StreamType_USB3Vision,
StreamType_Custom,
NUMSTREAMTYPE }
```

The enum definitions for TL Device nodes from the transport layer .xml files.

- enum StreamModeEnum {
StreamMode_Socket,
StreamMode_LWF,
StreamMode_MVA,
NUMSTREAMMODE }
- enum StreamBufferCountModeEnum {
StreamBufferCountMode_Manual,
NUMSTREAMBUFFERCOUNTMODE }
- enum StreamBufferHandlingModeEnum {
StreamBufferHandlingMode_OldestFirst,
StreamBufferHandlingMode_OldestFirstOverwrite,
StreamBufferHandlingMode_NewestOnly,
StreamBufferHandlingMode_NewestFirst,
NUMSTREAMBUFFERHANDLINGMODE }
- enum DeviceTypeEnum {
DeviceType_GigEVision,
DeviceType_CameraLink,
DeviceType_CameraLinkHS,
DeviceType_CoaXPress,
DeviceType_USB3Vision,
DeviceType_Custom,
NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
DeviceAccessStatus_Unknown,
DeviceAccessStatus_ReadWrite,
DeviceAccessStatus_ReadOnly,
DeviceAccessStatus_NoAccess,
DeviceAccessStatus_Busy,
DeviceAccessStatus_OpenReadWrite,
DeviceAccessStatus_OpenReadOnly,
NUMDEVICEACCESSSTATUS }
- enum GenICamXMLLocationEnum {
GenICamXMLLocation_Device,
GenICamXMLLocation_Host,
NUMGENICAMXMLLOCATION }
- enum GUIXMLLocationEnum {
GUIXMLLocation_Device,
GUIXMLLocation_Host,
NUMGUIXMLLOCATION }
- enum GevCCPEnum {
GevCCP_EnumEntry_GevCCP_OpenAccess,
GevCCP_EnumEntry_GevCCP_ExclusiveAccess,
GevCCP_EnumEntry_GevCCP_ControlAccess,
NUMGEVCCP }
- enum DeviceEndiannessMechanismEnum {
DeviceEndiannessMechanism_Legacy,
DeviceEndiannessMechanism_Standard,
NUMDEVICEENDIANESSMECHANISM }
- enum DeviceCurrentSpeedEnum {
DeviceCurrentSpeed_UnknownSpeed,

- DeviceCurrentSpeed_LowSpeed,
- DeviceCurrentSpeed_FullSpeed,
- DeviceCurrentSpeed_HighSpeed,
- DeviceCurrentSpeed_SuperSpeed,
- NUMDEVICECURRENTSPEED }
- enum InterfaceTypeEnum {
 - InterfaceType_GigEVision,
 - InterfaceType_CameraLink,
 - InterfaceType_CameraLinkHS,
 - InterfaceType_CoaXPress,
 - InterfaceType_USB3Vision,
 - InterfaceType_Custom,
 - NUMINTERFACETYPE }
- enum POEStatusEnum {
 - POEStatus_NotSupported,
 - POEStatus_PowerOff,
 - POEStatus_PowerOn,
 - NUMPOESTATUS }
- enum FilterDriverStatusEnum {
 - FilterDriverStatus_NotSupported,
 - FilterDriverStatus_Disabled,
 - FilterDriverStatus_Enabled,
 - NUMFILTERDRIVERSTATUS }
- enum TLTypeEnum {
 - TLType_GigEVision,
 - TLType_CameraLink,
 - TLType_CameraLinkHS,
 - TLType_CoaXPress,
 - TLType_USB3Vision,
 - TLType_Mixed,
 - TLType_Custom,
 - NUMTLTYPE }

Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API A↔
 VIRecorder
Provides the functionality for the user to record images to an AVI file.
- template<class T, class B>
 bool [operator==](#) (const std::nullptr_t, const [BasePtr](#)< T, B > &rhs)
Pointer equal.

Variables

- const uint64_t [EVENT_TIMEOUT_NONE](#) = 0
Timeout values for getting next image, device, or interface event.
- const uint64_t [EVENT_TIMEOUT_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

14.5.1 Enumeration Type Documentation

14.5.1.1 AcquisitionModeEnums

enum [AcquisitionModeEnums](#)

< Sets the acquisition mode of the device. Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition.

Enumerator

AcquisitionMode_Continuous	
AcquisitionMode_SingleFrame	
AcquisitionMode_MultiFrame	
NUM_ACQUISITIONMODE	

14.5.1.2 AcquisitionStatusSelectorEnums

enum [AcquisitionStatusSelectorEnums](#)

< Selects the internal acquisition signal to read using AcquisitionStatus.

Enumerator

AcquisitionStatusSelector_AcquisitionTriggerWait	Device is currently waiting for a trigger for the capture of one or many frames.
AcquisitionStatusSelector_AcquisitionActive	Device is currently doing an acquisition of one or many frames.
AcquisitionStatusSelector_AcquisitionTransfer	Device is currently transferring an acquisition of one or many frames.
AcquisitionStatusSelector_FrameTriggerWait	Device is currently waiting for a frame start trigger.
AcquisitionStatusSelector_FrameActive	Device is currently doing the capture of a frame.
AcquisitionStatusSelector_ExposureActive	Device is doing the exposure of a frame.
NUM_ACQUISITIONSTATUSSELECTION	

14.5.1.3 ActionCommandStatus

enum [ActionCommandStatus](#)

Possible Status Codes Returned from Action Command.

Enumerator

SPINNAKER_ACTION_COMMAND_STATUS_OK	
SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME	
SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW	
SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE	
SPINNAKER_ACTION_COMMAND_STATUS_ERROR	

14.5.1.4 ActionUnconditionalModeEnums

enum [ActionUnconditionalModeEnums](#)

< Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Enumerator

ActionUnconditionalMode_Off	Unconditional mode is disabled.
ActionUnconditionalMode_On	Unconditional mode is enabled.
NUM_ACTIONUNCONDITIONALMODE	

14.5.1.5 AdcBitDepthEnums

enum [AdcBitDepthEnums](#)

< Selects which ADC bit depth to use. A higher ADC bit depth results in better image quality but slower maximum frame rate.

Enumerator

AdcBitDepth_Bit8	
AdcBitDepth_Bit10	
AdcBitDepth_Bit12	
AdcBitDepth_Bit14	
NUM_ADCBITDEPTH	

14.5.1.6 AutoAlgorithmSelectorEnums

enum [AutoAlgorithmSelectorEnums](#)

< Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Enumerator

AutoAlgorithmSelector_Awb	Selects the Auto White Balance algorithm.
AutoAlgorithmSelector_Ae	Selects the Auto Exposure algorithm.
NUM_AUTOALGORITHMSELECTOR	

14.5.1.7 AutoExposureControlPriorityEnums

enum `AutoExposureControlPriorityEnums`

< Selects whether to adjust gain or exposure first. When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Enumerator

<code>AutoExposureControlPriority_Gain</code>	
<code>AutoExposureControlPriority_ExposureTime</code>	
<code>NUM_AUTOEXPOSURECONTROLPRIORITY</code>	

14.5.1.8 AutoExposureLightingModeEnums

enum `AutoExposureLightingModeEnums`

< Selects a lighting mode: Backlight, Frontlight or Normal (default). a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Enumerator

<code>AutoExposureLightingMode_AutoDetect</code>	
<code>AutoExposureLightingMode_Backlight</code>	
<code>AutoExposureLightingMode_Frontlight</code>	
<code>AutoExposureLightingMode_Normal</code>	
<code>NUM_AUTOEXPOSURELIGHTINGMODE</code>	

14.5.1.9 AutoExposureMeteringModeEnums

enum `AutoExposureMeteringModeEnums`

< Selects a metering mode: average, spot, or partial metering. a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Enumerator

AutoExposureMeteringMode_Average	
AutoExposureMeteringMode_Spot	
AutoExposureMeteringMode_Partial	
AutoExposureMeteringMode_CenterWeighted	
AutoExposureMeteringMode_HistogramPeak	
NUM_AUTOEXPOSUREMETERINGMODE	

14.5.1.10 AutoExposureTargetGreyValueAutoEnums

enum [AutoExposureTargetGreyValueAutoEnums](#)

< This indicates whether the target image grey level is automatically set by the camera or manually set by the user. Note that the target grey level is in the linear domain before gamma correction is applied.

Enumerator

AutoExposureTargetGreyValueAuto_Off	Target grey value is manually controlled
AutoExposureTargetGreyValueAuto_Continuous	Target grey value is constantly adapted by the device to maximize the dynamic range.
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO	

14.5.1.11 BalanceRatioSelectorEnums

enum [BalanceRatioSelectorEnums](#)

< Selects a balance ratio to configure once a balance ratio control has been selected.

Enumerator

BalanceRatioSelector_Red	Selects the red balance ratio control for adjustment. The red balance ratio is relative to the green channel.
BalanceRatioSelector_Blue	Selects the blue balance ratio control for adjustment. The blue balance ratio is relative to the green channel.
NUM_BALANCERATIOSELECTOR	

14.5.1.12 BalanceWhiteAutoEnums

enum [BalanceWhiteAutoEnums](#)

< White Balance compensates for color shifts caused by different lighting conditions. It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

Enumerator

BalanceWhiteAuto_Off	Sets operation mode to Off, which is manual control.
BalanceWhiteAuto_Once	Sets operation mode to once. Once runs for a number of iterations and then sets White Balance Auto to Off.
BalanceWhiteAuto_Continuous	Sets operation mode to continuous. Continuous automatically adjusts values if the colors are imbalanced.
NUM_BALANCEWHITEAUTO	

14.5.1.13 BalanceWhiteAutoProfileEnums

enum [BalanceWhiteAutoProfileEnums](#)

< Selects the profile used by BalanceWhiteAuto.

Enumerator

BalanceWhiteAutoProfile_Indoor	Indoor auto white balance Profile. Can be used to compensate for artificial lighting.
BalanceWhiteAutoProfile_Outdoor	Outdoor auto white balance profile. Designed for scenes with natural lighting.
NUM_BALANCEWHITEAUTOPROFILE	

14.5.1.14 BinningHorizontalModeEnums

enum [BinningHorizontalModeEnums](#)

<

Enumerator

BinningHorizontalMode_Sum	The response from the combined horizontal cells is added, resulting in increased sensitivity (a brighter image).
BinningHorizontalMode_Average	The response from the combined horizontal cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.
NUM_BINNINGHORIZONTALMODE	

14.5.1.15 BinningSelectorEnums

enum `BinningSelectorEnums`

< Selects which binning engine is controlled by the `BinningHorizontal` and `BinningVertical` features.

Enumerator

<code>BinningSelector_All</code>	The total amount of binning to be performed on the captured sensor data.
<code>BinningSelector_Sensor</code>	The portion of binning to be performed on the sensor directly.
<code>BinningSelector_ISP</code>	The portion of binning to be performed by the image signal processing engine (ISP) outside of the sensor. Note: the ISP can be disabled.
<code>NUM_BINNINGSELECTOR</code>	

14.5.1.16 BinningVerticalModeEnums

enum `BinningVerticalModeEnums`

<

Enumerator

<code>BinningVerticalMode_Sum</code>	The response from the combined vertical cells is added, resulting in increased sensitivity (a brighter image).
<code>BinningVerticalMode_Average</code>	The response from the combined vertical cells is averaged, resulting in increased signal/noise ratio. Not all sensors support average binning.
<code>NUM_BINNINGVERTICALMODE</code>	

14.5.1.17 BlackLevelAutoBalanceEnums

enum `BlackLevelAutoBalanceEnums`

< Controls the mode for automatic black level balancing between the sensor color channels or taps. The black level coefficients of each channel are adjusted so they are matched.

Enumerator

<code>BlackLevelAutoBalance_Off</code>	Black level tap balancing is user controlled using <code>BlackLevel</code> .
<code>BlackLevelAutoBalance_Once</code>	Black level tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
<code>BlackLevelAutoBalance_Continuous</code>	Black level tap balancing is constantly adjusted by the device.
<code>NUM_BLACKLEVELAUTOBALANCE</code>	

14.5.1.18 BlackLevelAutoEnums

enum `BlackLevelAutoEnums`

< Controls the mode for automatic black level adjustment. The exact algorithm used to implement this adjustment is device-specific.

Enumerator

BlackLevelAuto_Off	Analog black level is user controlled using BlackLevel.
BlackLevelAuto_Once	Analog black level is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
BlackLevelAuto_Continuous	Analog black level is constantly adjusted by the device.
NUM_BLACKLEVELAUTO	

14.5.1.19 BlackLevelSelectorEnums

enum `BlackLevelSelectorEnums`

< Selects which black level to control. Only All can be set by the user. Analog and Digital are read-only.

Enumerator

BlackLevelSelector_All	
BlackLevelSelector_Analog	
BlackLevelSelector_Digital	
NUM_BLACKLEVELSELECTOR	

14.5.1.20 BufferOwnership

enum `BufferOwnership`

Enumerator

SPINNAKER_BUFFER_OWNERSHIP_SYSTEM	
SPINNAKER_BUFFER_OWNERSHIP_USER	

14.5.1.21 CCMAplication

enum `CCMAplication`

Enumerator

SPINNAKER_CCM_APPLICATION_GENERIC	
SPINNAKER_CCM_APPLICATION_MICROSCOPY	

14.5.1.22 CCMColorSpace

enum [CCMColorSpace](#)

Enumerator

SPINNAKER_CCM_COLOR_SPACE_OFF	
SPINNAKER_CCM_COLOR_SPACE_SRGB	

14.5.1.23 CCMColorTemperature

enum [CCMColorTemperature](#)

Enumerator

SPINNAKER_CCM_COLOR_TEMP_TUNGSTEN_2800K	
SPINNAKER_CCM_COLOR_TEMP_WARM_FLUORESCENT_3000K	
SPINNAKER_CCM_COLOR_TEMP_COOL_FLUORESCENT_4000K	
SPINNAKER_CCM_COLOR_TEMP_SUNNY_5000K	
SPINNAKER_CCM_COLOR_TEMP_CLOUDY_6500K	
SPINNAKER_CCM_COLOR_TEMP_SHADE_8000K	
SPINNAKER_CCM_COLOR_TEMP_GENERAL	

14.5.1.24 CCMSensor

enum [CCMSensor](#)

Enumerator

SPINNAKER_CCM_SENSOR_IMX250	

14.5.1.25 CCMTypeenum [CCMType](#)**Enumerator**

SPINNAKER_CCM_TYPE_LINEAR	
SPINNAKER_CCM_TYPE_ADVANCED	

14.5.1.26 ChunkBlackLevelSelectorEnumsenum [ChunkBlackLevelSelectorEnums](#)

< Selects which black level to retrieve

Enumerator

ChunkBlackLevelSelector_All	
NUM_CHUNKBLACKLEVELSELECTOR	

14.5.1.27 ChunkCounterSelectorEnumsenum [ChunkCounterSelectorEnums](#)

< Selects which counter to retrieve data from.

Enumerator

ChunkCounterSelector_Counter0	Selects the counter 0.
ChunkCounterSelector_Counter1	Selects the counter 1.
ChunkCounterSelector_Counter2	Selects the counter 2.
NUM_CHUNKCOUNTERSELECTOR	

14.5.1.28 ChunkEncoderSelectorEnumsenum [ChunkEncoderSelectorEnums](#)

< Selects which Encoder to retrieve data from.

Enumerator

ChunkEncoderSelector_Encoder0	Selects the first Encoder.

Enumerator

ChunkEncoderSelector_Encoder1	Selects the first Encoder.
ChunkEncoderSelector_Encoder2	Selects the second Encoder.
NUM_CHUNKENCODERSELECTOR	

14.5.1.29 ChunkEncoderStatusEnums

enum [ChunkEncoderStatusEnums](#)

< Returns the motion status of the selected encoder.

Enumerator

ChunkEncoderStatus_EncoderUp	The encoder counter last incremented.
ChunkEncoderStatus_EncoderDown	The encoder counter last decremented.
ChunkEncoderStatus_EncoderIdle	The encoder is not active.
ChunkEncoderStatus_EncoderStatic	No motion within the EncoderTimeout time.
NUM_CHUNKENCODERSTATUS	

14.5.1.30 ChunkExposureTimeSelectorEnums

enum [ChunkExposureTimeSelectorEnums](#)

< Selects which exposure time is read by the ChunkExposureTime feature.

Enumerator

ChunkExposureTimeSelector_Common	Selects the common ExposureTime.
ChunkExposureTimeSelector_Red	Selects the red common ExposureTime.
ChunkExposureTimeSelector_Green	Selects the green ExposureTime.
ChunkExposureTimeSelector_Blue	Selects the blue ExposureTime.
ChunkExposureTimeSelector_Cyan	Selects the cyan common ExposureTime..
ChunkExposureTimeSelector_Magenta	Selects the magenta ExposureTime..
ChunkExposureTimeSelector_Yellow	Selects the yellow ExposureTime..
ChunkExposureTimeSelector_Infrared	Selects the infrared ExposureTime.
ChunkExposureTimeSelector_Ultraviolet	Selects the ultraviolet ExposureTime.
ChunkExposureTimeSelector_Stage1	Selects the first stage ExposureTime.
ChunkExposureTimeSelector_Stage2	Selects the second stage ExposureTime.
NUM_CHUNKEXPOSURETIMESELECTOR	

14.5.1.31 ChunkGainSelectorEnums

enum [ChunkGainSelectorEnums](#)

< Selects which gain to retrieve

Enumerator

ChunkGainSelector_All	
ChunkGainSelector_Red	
ChunkGainSelector_Green	
ChunkGainSelector_Blue	
NUM_CHUNKGAINSELECTOR	

14.5.1.32 ChunkImageComponentEnums

enum [ChunkImageComponentEnums](#)

< Returns the component of the payload image. This can be used to identify the image component of a generic part in a multipart transfer.

Enumerator

ChunkImageComponent_Intensity	The image data is the intensity component.
ChunkImageComponent_Color	The image data is color component.
ChunkImageComponent_Infrared	The image data is infrared component.
ChunkImageComponent_Ultraviolet	The image data is the ultraviolet component.
ChunkImageComponent_Range	The image data is the range (distance) component.
ChunkImageComponent_Disparity	The image data is the disparity component.
ChunkImageComponent_Confidence	The image data is the confidence map component.
ChunkImageComponent_Scatter	The image data is the scatter component.
NUM_CHUNKIMAGECOMPONENT	

14.5.1.33 ChunkPixelFormatEnums

enum [ChunkPixelFormatEnums](#)

< Format of the pixel provided by the camera

Enumerator

ChunkPixelFormat_Mono8	
ChunkPixelFormat_Mono12Packed	
ChunkPixelFormat_Mono16	
ChunkPixelFormat_RGB8Packed	

Enumerator

ChunkPixelFormat_YUV422Packed	
ChunkPixelFormat_BayerGR8	
ChunkPixelFormat_BayerRG8	
ChunkPixelFormat_BayerGB8	
ChunkPixelFormat_BayerBG8	
ChunkPixelFormat_YCbCr601_422_8_CbYCrY	
NUM_CHUNKPIXELFORMAT	

14.5.1.34 ChunkRegionIDEnums

enum [ChunkRegionIDEnums](#)

< Returns the identifier of Region that the image comes from.

Enumerator

ChunkRegionID_Region0	Image comes from the Region 0.
ChunkRegionID_Region1	Image comes from the Region 1.
ChunkRegionID_Region2	Image comes from the Region 2.
NUM_CHUNKREGIONID	

14.5.1.35 ChunkScan3dCoordinateReferenceSelectorEnums

enum [ChunkScan3dCoordinateReferenceSelectorEnums](#)

< Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

Enumerator

ChunkScan3dCoordinateReferenceSelector_RotationX	Rotation around X axis.
ChunkScan3dCoordinateReferenceSelector_RotationY	Rotation around Y axis.
ChunkScan3dCoordinateReferenceSelector_RotationZ	Rotation around Z axis.
ChunkScan3dCoordinateReferenceSelector_TranslationX	X axis translation.
ChunkScan3dCoordinateReferenceSelector_TranslationY	Y axis translation.
ChunkScan3dCoordinateReferenceSelector_TranslationZ	Z axis translation.
NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR	

14.5.1.36 ChunkScan3dCoordinateSelectorEnums

enum [ChunkScan3dCoordinateSelectorEnums](#)

< Selects which Coordinate to retrieve data from.

Enumerator

ChunkScan3dCoordinateSelector_CoordinateA	The first (X or Theta) coordinate
ChunkScan3dCoordinateSelector_CoordinateB	The second (Y or Phi) coordinate
ChunkScan3dCoordinateSelector_CoordinateC	The third (Z or Rho) coordinate.
NUM_CHUNKSCAN3DCOORDINATESELECTOR	

14.5.1.37 ChunkScan3dCoordinateSystemEnums

enum [ChunkScan3dCoordinateSystemEnums](#)

< Returns the Coordinate [System](#) of the image included in the payload.

Enumerator

ChunkScan3dCoordinateSystem_Cartesian	Default value. 3-axis orthogonal, right-hand X-Y-Z.
ChunkScan3dCoordinateSystem_Spherical	A Theta-Phi-Rho coordinate system.
ChunkScan3dCoordinateSystem_Cylindrical	A Theta-Y-Rho coordinate system.
NUM_CHUNKSCAN3DCOORDINATESYSTEM	

14.5.1.38 ChunkScan3dCoordinateSystemReferenceEnums

enum [ChunkScan3dCoordinateSystemReferenceEnums](#)

< Returns the Coordinate [System](#) Position of the image included in the payload.

Enumerator

ChunkScan3dCoordinateSystemReference_Anchor	Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.
ChunkScan3dCoordinateSystemReference_↔ Transformed	Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.
NUM_CHUNKSCAN3DCOORDINATESYSTEMRE↔ FERENCE	

14.5.1.39 ChunkScan3dCoordinateTransformSelectorEnums

enum [ChunkScan3dCoordinateTransformSelectorEnums](#)

< Selector for transform values.

Enumerator

ChunkScan3dCoordinateTransformSelector_RotationX	Rotation around X axis.
ChunkScan3dCoordinateTransformSelector_RotationY	Rotation around Y axis.
ChunkScan3dCoordinateTransformSelector_RotationZ	Rotation around Z axis.
ChunkScan3dCoordinateTransformSelector_TranslationX	Translation along X axis.
ChunkScan3dCoordinateTransformSelector_TranslationY	Translation along Y axis.
ChunkScan3dCoordinateTransformSelector_TranslationZ	Translation along Z axis.
NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR	

14.5.1.40 ChunkScan3dDistanceUnitEnums

enum [ChunkScan3dDistanceUnitEnums](#)

< Returns the Distance Unit of the payload image.

Enumerator

ChunkScan3dDistanceUnit_Millimeter	Default value. Distance values are in millimeter units.
ChunkScan3dDistanceUnit_Inch	Distance values are in inch units.
NUM_CHUNKSCAN3DDISTANCEUNIT	

14.5.1.41 ChunkScan3dOutputModeEnums

enum [ChunkScan3dOutputModeEnums](#)

< Returns the Calibrated Mode of the payload image.

Enumerator

ChunkScan3dOutputMode_UncalibratedC	Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.
ChunkScan3dOutputMode_CalibratedABC_Grid	3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.
ChunkScan3dOutputMode_CalibratedABC_Point↔ Cloud	3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.

Enumerator

ChunkScan3dOutputMode_CalibratedAC	2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.
ChunkScan3dOutputMode_CalibratedAC_Linescan	2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_CalibratedC	Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.
ChunkScan3dOutputMode_CalibratedC_Linescan	Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_RectifiedC	Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.
ChunkScan3dOutputMode_RectifiedC_Linescan	Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.
ChunkScan3dOutputMode_DisparityC	Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.
ChunkScan3dOutputMode_DisparityC_Linescan	Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.
NUM_CHUNKSCAN3DOUTPUTMODE	

14.5.1.42 ChunkSelectorEnums

enum [ChunkSelectorEnums](#)

< Selects which chunk data to enable or disable.

Enumerator

ChunkSelector_Image	
ChunkSelector_CRC	
ChunkSelector_FrameID	
ChunkSelector_OffsetX	
ChunkSelector_OffsetY	
ChunkSelector_Width	
ChunkSelector_Height	
ChunkSelector_ExposureTime	

Enumerator

ChunkSelector_Gain	
ChunkSelector_BlackLevel	
ChunkSelector_PixelFormat	
ChunkSelector_Timestamp	
ChunkSelector_SequencerSetActive	
ChunkSelector_SerialData	
ChunkSelector_ExposureEndLineStatusAll	
NUM_CHUNKSELECTOR	

14.5.1.43 ChunkSourceIDEnums

enum [ChunkSourceIDEnums](#)

< Returns the identifier of Source that the image comes from.

Enumerator

ChunkSourceID_Source0	Image comes from the Source 0.
ChunkSourceID_Source1	Image comes from the Source 1.
ChunkSourceID_Source2	Image comes from the Source 2.
NUM_CHUNKSOURCEID	

14.5.1.44 ChunkTimerSelectorEnums

enum [ChunkTimerSelectorEnums](#)

< Selects which Timer to retrieve data from.

Enumerator

ChunkTimerSelector_Timer0	Selects the first Timer.
ChunkTimerSelector_Timer1	Selects the first Timer.
ChunkTimerSelector_Timer2	Selects the second Timer.
NUM_CHUNKTIMERSELECTOR	

14.5.1.45 ChunkTransferStreamIDEnums

enum [ChunkTransferStreamIDEnums](#)

< Returns identifier of the stream that generated this block.

Enumerator

ChunkTransferStreamID_Stream0	Data comes from Stream0.
ChunkTransferStreamID_Stream1	Data comes from Stream1.
ChunkTransferStreamID_Stream2	Data comes from Stream2.
ChunkTransferStreamID_Stream3	Data comes from Stream3.
NUM_CHUNKTRANSFERSTREAMID	

14.5.1.46 ClConfigurationEnums

enum [ClConfigurationEnums](#)

< This [Camera](#) Link specific feature describes the configuration used by the camera. It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitization↔Taps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera.

Enumerator

ClConfiguration_Base	Standard base configuration described by the Camera Link standard.
ClConfiguration_Medium	Standard medium configuration described by the Camera Link standard.
ClConfiguration_Full	Standard full configuration described by the Camera Link standard.
ClConfiguration_DualBase	The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two Camera Link base ports. It is responsibility of the application or frame grabber to reconstruct the full image. Only one of the ports (fixed) serves as the "master" for serial communication and triggering.
ClConfiguration_EightyBit	Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the Camera Link standard.
NUM_CLCONFIGURATION	

14.5.1.47 ClTimeSlotsCountEnums

enum [ClTimeSlotsCountEnums](#)

< This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Enumerator

ClTimeSlotsCount_One	One
ClTimeSlotsCount_Two	Two
ClTimeSlotsCount_Three	Three
NUM_CLTIMESLOTSCOUNT	

14.5.1.48 ColorProcessingAlgorithm

enum `ColorProcessingAlgorithm`

Color processing algorithms.

Please refer to our knowledge base at article at <https://www.flir.com/support-center/iis/machine-vision/k> for complete details for each algorithm.

Enumerator

SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_NONE	No color processing.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_NEAREST_NEIGHBOR	Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_NEAREST_NEIGHBOR_AVG	Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_BILINEAR	Weighted average of surrounding 4 pixels in a 2x2 neighborhood.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_EDGE_SENSING	Weights surrounding pixels based on localized edge orientation.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_HQ_LINEAR	Well-balanced speed and quality.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_IPP	Multi-threaded with similar results to edge sensing.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_DIRECTIONAL_FILTER	Best quality but much faster than rigorous. More memory intensive than other color processing algorithms.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_RIGOROUS	Slowest but produces good results.
SPINNAKER_COLOR_PROCESSING_ALGORITHM↔ HM_WEIGHTED_DIRECTIONAL_FILTER	Weighted pixel average from different directions.

14.5.1.49 ColorTransformationSelectorEnums

enum `ColorTransformationSelectorEnums`

< Selects which Color Transformation module is controlled by the various Color Transformation features

Enumerator

ColorTransformationSelector_RGBtoRGB	
ColorTransformationSelector_RGBtoYUV	
NUM_COLORTRANSFORMATIONSELECTOR	

14.5.1.50 ColorTransformationValueSelectorEnums

enum [ColorTransformationValueSelectorEnums](#)

< Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Enumerator

ColorTransformationValueSelector_Gain00	
ColorTransformationValueSelector_Gain01	
ColorTransformationValueSelector_Gain02	
ColorTransformationValueSelector_Gain10	
ColorTransformationValueSelector_Gain11	
ColorTransformationValueSelector_Gain12	
ColorTransformationValueSelector_Gain20	
ColorTransformationValueSelector_Gain21	
ColorTransformationValueSelector_Gain22	
ColorTransformationValueSelector_Offset0	
ColorTransformationValueSelector_Offset1	
ColorTransformationValueSelector_Offset2	
NUM_COLORTRANSFORMATIONVALUESELECTOR	

14.5.1.51 CompressionSaturationPriorityEnums

enum [CompressionSaturationPriorityEnums](#)

< When FrameRate is enabled, camera drops frames if datarate is saturated. If FrameRate is disabled, camera adjusts the framerate to match the maximum achievable datarate.

Enumerator

CompressionSaturationPriority_DropFrame	Frames which will cause the MaxDatarateThreshold to be exceeded will not be transmitted. Requires FrameRateEnable to be True
CompressionSaturationPriority_ReduceFrameRate	AcquisitionFrameRate is dynamically adjusted to the highest possible value without exceeding the MaxDatarateThreshold.
NUM_COMPRESSIONSATURATIONPRIORITY	

14.5.1.52 CounterEventActivationEnums

enum [CounterEventActivationEnums](#)

< Selects the activation mode of the event to increment the Counter.

Enumerator

CounterEventActivation_LevelLow	
CounterEventActivation_LevelHigh	
CounterEventActivation_FallingEdge	
CounterEventActivation_RisingEdge	
CounterEventActivation_AnyEdge	
NUM_COUNTEREVENTACTIVATION	

14.5.1.53 CounterEventSourceEnums

enum [CounterEventSourceEnums](#)

< Selects the event that will increment the counter

Enumerator

CounterEventSource_Off	Off
CounterEventSource_MHzTick	MHzTick
CounterEventSource_Line0	Line0
CounterEventSource_Line1	Line1
CounterEventSource_Line2	Line2
CounterEventSource_Line3	Line3
CounterEventSource_UserOutput0	UserOutput0
CounterEventSource_UserOutput1	UserOutput1
CounterEventSource_UserOutput2	UserOutput2
CounterEventSource_UserOutput3	UserOutput3
CounterEventSource_Counter0Start	Counter0Start
CounterEventSource_Counter1Start	Counter1Start
CounterEventSource_Counter0End	Counter0End
CounterEventSource_Counter1End	Counter1End
CounterEventSource_LogicBlock0	LogicBlock0
CounterEventSource_LogicBlock1	LogicBlock1
CounterEventSource_ExposureStart	ExposureStart
CounterEventSource_ExposureEnd	ExposureEnd
CounterEventSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTEREVENTSOURCE	

14.5.1.54 CounterResetActivationEnums

enum [CounterResetActivationEnums](#)

< Selects the Activation mode of the Counter Reset Source signal.

Enumerator

CounterResetActivation_LevelLow	
CounterResetActivation_LevelHigh	
CounterResetActivation_FallingEdge	
CounterResetActivation_RisingEdge	
CounterResetActivation_AnyEdge	
NUM_COUNTERRESETACTIVATION	

14.5.1.55 CounterResetSourceEnums

enum [CounterResetSourceEnums](#)

< Selects the signal that will be the source to reset the counter.

Enumerator

CounterResetSource_Off	Off
CounterResetSource_Line0	Line0
CounterResetSource_Line1	Line1
CounterResetSource_Line2	Line2
CounterResetSource_Line3	Line3
CounterResetSource_UserOutput0	UserOutput0
CounterResetSource_UserOutput1	UserOutput1
CounterResetSource_UserOutput2	UserOutput2
CounterResetSource_UserOutput3	UserOutput3
CounterResetSource_Counter0Start	Counter0Start
CounterResetSource_Counter1Start	Counter1Start
CounterResetSource_Counter0End	Counter0End
CounterResetSource_Counter1End	Counter1End
CounterResetSource_LogicBlock0	LogicBlock0
CounterResetSource_LogicBlock1	LogicBlock1
CounterResetSource_ExposureStart	ExposureStart
CounterResetSource_ExposureEnd	ExposureEnd
CounterResetSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTERRESETSOURCE	

14.5.1.56 CounterSelectorEnums

enum [CounterSelectorEnums](#)

< Selects which counter to configure

Enumerator

CounterSelector_Counter0	
CounterSelector_Counter1	
NUM_COUNTERSELECTOR	

14.5.1.57 CounterStatusEnums

enum [CounterStatusEnums](#)

< Returns the current status of the counter.

Enumerator

CounterStatus_CounterIdle	The counter is idle.
CounterStatus_CounterTriggerWait	The counter is waiting for a start trigger.
CounterStatus_CounterActive	The counter is counting for the specified duration.
CounterStatus_CounterCompleted	The counter reached the CounterDuration count.
CounterStatus_CounterOverflow	The counter reached its maximum possible count.
NUM_COUNTERSTATUS	

14.5.1.58 CounterTriggerActivationEnums

enum [CounterTriggerActivationEnums](#)

< Selects the activation mode of the trigger to start the counter.

Enumerator

CounterTriggerActivation_LevelLow	
CounterTriggerActivation_LevelHigh	
CounterTriggerActivation_FallingEdge	
CounterTriggerActivation_RisingEdge	
CounterTriggerActivation_AnyEdge	
NUM_COUNTERTRIGGERACTIVATION	

14.5.1.59 CounterTriggerSourceEnums

enum [CounterTriggerSourceEnums](#)

< Selects the source of the trigger to start the counter

Enumerator

CounterTriggerSource_Off	Off
CounterTriggerSource_Line0	Line0
CounterTriggerSource_Line1	Line1
CounterTriggerSource_Line2	Line2
CounterTriggerSource_Line3	Line3
CounterTriggerSource_UserOutput0	UserOutput0
CounterTriggerSource_UserOutput1	UserOutput1
CounterTriggerSource_UserOutput2	UserOutput2
CounterTriggerSource_UserOutput3	UserOutput3
CounterTriggerSource_Counter0Start	Counter0Start
CounterTriggerSource_Counter1Start	Counter1Start
CounterTriggerSource_Counter0End	Counter0End
CounterTriggerSource_Counter1End	Counter1End
CounterTriggerSource_LogicBlock0	LogicBlock0
CounterTriggerSource_LogicBlock1	LogicBlock1
CounterTriggerSource_ExposureStart	ExposureStart
CounterTriggerSource_ExposureEnd	ExposureEnd
CounterTriggerSource_FrameTriggerWait	FrameTriggerWait
NUM_COUNTERTRIGGERSOURCE	

14.5.1.60 CxpConnectionTestModeEnums

enum [CxpConnectionTestModeEnums](#)

< Enables the test mode for an individual physical connection of the Device.

Enumerator

CxpConnectionTestMode_Off	Off
CxpConnectionTestMode_Mode1	Mode 1
NUM_CXPCONNECTIONTESTMODE	

14.5.1.61 CxpLinkConfigurationEnums

enum [CxpLinkConfigurationEnums](#)

< This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device. In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus.

Enumerator

CxpLinkConfiguration_Auto	Sets Automatic discovery for the Link Configuration.
CxpLinkConfiguration_CXP1_X1	Force the Link to 1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X1	Force the Link to 1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X1	Force the Link to 1 Connection operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X1	Force the Link to 1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X1	Force the Link to 1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X2	Force the Link to 2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X2	Force the Link to 2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X2	Force the Link to 2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X2	Force the Link to 2 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X2	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X3	Force the Link to 3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X3	Force the Link to 3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X3	Force the Link to 3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X3	Force the Link to 3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X3	Force the Link to 3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X4	Force the Link to 4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X4	Force the Link to 4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X4	Force the Link to 4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X4	Force the Link to 4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X4	Force the Link to 4 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X5	Force the Link to 5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X5	Force the Link to 5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X5	Force the Link to 5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X5	Force the Link to 5 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X5	Force the Link to 5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfiguration_CXP1_X6	Force the Link to 6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfiguration_CXP2_X6	Force the Link to 6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfiguration_CXP3_X6	Force the Link to 6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfiguration_CXP5_X6	Force the Link to 6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfiguration_CXP6_X6	Force the Link to 6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATION	

14.5.1.62 CxpLinkConfigurationPreferredEnums

enum [CxpLinkConfigurationPreferredEnums](#)

< Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

Enumerator

CxpLinkConfigurationPreferred_CXP1_X1	1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X1	1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X1	1 Connection operating at CXP-3 speed (3.125 Gbps).

Enumerator

CxpLinkConfigurationPreferred_CXP5_X1	1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X1	1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X2	2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X2	2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X2	2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X2	2 Connections operating at CXP-4 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X2	3 Connections operating at CXP-5 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X3	3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X3	3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X3	3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X3	3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X3	3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X4	4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X4	4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X4	4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X4	4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X4	4 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X5	5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X5	5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X5	5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X5	5 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X5	5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationPreferred_CXP1_X6	6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationPreferred_CXP2_X6	6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationPreferred_CXP3_X6	6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationPreferred_CXP5_X6	6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationPreferred_CXP6_X6	6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATIONPREFERRED	

14.5.1.63 CxpLinkConfigurationStatusEnums

enum `CxpLinkConfigurationStatusEnums`

< This feature indicates the current and active Link configuration used by the Device.

Enumerator

CxpLinkConfigurationStatus_None	The Link configuration of the Device is unknown. Either the configuration operation has failed or there is nothing connected.
CxpLinkConfigurationStatus_Pending	The Device is in the process of configuring the Link. The Link cannot be used yet.
CxpLinkConfigurationStatus_CXP1_X1	1 Connection operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X1	1 Connection operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X1	1 Connection operating at CXP-3 speed (3.125 Gbps).

Enumerator

CxpLinkConfigurationStatus_CXP5_X1	1 Connection operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X1	1 Connection operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X2	2 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X2	2 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X2	2 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X2	2 Connections operating at CXP-4 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X2	3 Connections operating at CXP-5 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X3	3 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X3	3 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X3	3 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X3	3 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X3	3 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X4	4 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X4	4 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X4	4 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X4	4 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X4	4 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X5	5 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X5	5 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X5	5 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X5	5 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X5	5 Connections operating at CXP-6 speed (6.25 Gbps).
CxpLinkConfigurationStatus_CXP1_X6	6 Connections operating at CXP-1 speed (1.25 Gbps).
CxpLinkConfigurationStatus_CXP2_X6	6 Connections operating at CXP-2 speed (2.50 Gbps).
CxpLinkConfigurationStatus_CXP3_X6	6 Connections operating at CXP-3 speed (3.125 Gbps).
CxpLinkConfigurationStatus_CXP5_X6	6 Connections operating at CXP-5 speed (5.00 Gbps).
CxpLinkConfigurationStatus_CXP6_X6	6 Connections operating at CXP-6 speed (6.25 Gbps).
NUM_CXPLINKCONFIGURATIONSTATUS	

14.5.1.64 CxpPoCxpStatusEnums

```
enum CxpPoCxpStatusEnums
```

< Returns the Power over CoaXPress (PoCXP) status of the Device.

Enumerator

CxpPoCxpStatus_Auto	Normal automatic PoCXP operation.
CxpPoCxpStatus_Off	PoCXP is forced off.
CxpPoCxpStatus_Tripped	The Link has shut down because of an over-current trip.
NUM_CXPPOCXPSTATUS	

14.5.1.65 DecimationHorizontalModeEnums

enum [DecimationHorizontalModeEnums](#)

< The mode used to reduce the horizontal resolution when DecimationHorizontal is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Enumerator

DecimationHorizontalMode_Discard	The value of every Nth pixel is kept, others are discarded.
NUM_DECIMATIONHORIZONTALMODE	

14.5.1.66 DecimationSelectorEnums

enum [DecimationSelectorEnums](#)

< Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

Enumerator

DecimationSelector_All	The total amount of decimation to be performed on the captured image data.
DecimationSelector_Sensor	The portion of decimation to be performed on the sensor directly. Currently this is the only decimation layer available and hence is identical to the "All" layer. All decimation modification should therefore be done via the "All" layer only.
NUM_DECIMATIONSELECTOR	

14.5.1.67 DecimationVerticalModeEnums

enum [DecimationVerticalModeEnums](#)

< The mode used to reduce the vertical resolution when DecimationVertical is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Enumerator

DecimationVerticalMode_Discard	The value of every Nth pixel is kept, others are discarded.
NUM_DECIMATIONVERTICALMODE	

14.5.1.68 DefectCorrectionModeEnums

enum [DefectCorrectionModeEnums](#)

< Controls the method used for replacing defective pixels.

Enumerator

DefectCorrectionMode_Average	Pixels are replaced with the average of their neighbours. This is the normal mode of operation.
DefectCorrectionMode_Highlight	Pixels are replaced with the maximum pixel value (i.e., 255 for 8-bit images). Can be used for debugging the table.
DefectCorrectionMode_Zero	Pixels are replaced by the value zero. Can be used for testing the table.
NUM_DEFECTCORRECTIONMODE	

14.5.1.69 DeinterlacingEnums

enum [DeinterlacingEnums](#)

< Controls how the device performs de-interlacing.

Enumerator

Deinterlacing_Off	The device doesn't perform de-interlacing.
Deinterlacing_LineDuplication	The device performs de-interlacing by outputting each line of each field twice.
Deinterlacing_Weave	The device performs de-interlacing by interleaving the lines of all fields.
NUM_DEINTERLACING	

14.5.1.70 DeviceAccessStatusEnum

enum [DeviceAccessStatusEnum](#)

< Gets the access status the transport layer Producer has on the device.

Enumerator

DeviceAccessStatus_Unknown	Not known to producer.
DeviceAccessStatus_ReadWrite	Full access
DeviceAccessStatus_ReadOnly	Read-only access
DeviceAccessStatus_NoAccess	Not available to connect
DeviceAccessStatus_Busy	The device is already opened by another entity
DeviceAccessStatus_OpenReadWrite	Open in Read/Write mode by this GenTL host
DeviceAccessStatus_OpenReadOnly	Open in Read access mode by this GenTL host
NUMDEVICEACCESSSTATUS	

14.5.1.71 DeviceCharacterSetEnums

enum [DeviceCharacterSetEnums](#)

< Character set used by the strings of the device's bootstrap registers.

Enumerator

DeviceCharacterSet_UTF8	
DeviceCharacterSet_ASCII	
NUM_DEVICECHARACTERSET	

14.5.1.72 DeviceClockSelectorEnums

enum [DeviceClockSelectorEnums](#)

< Selects the clock frequency to access from the device.

Enumerator

DeviceClockSelector_Sensor	Clock frequency of the image sensor of the camera.
DeviceClockSelector_SensorDigitization	Clock frequency of the camera A/D conversion stage.
DeviceClockSelector_CameraLink	Frequency of the Camera Link clock.
NUM_DEVICECLOCKSELECTOR	

14.5.1.73 DeviceConnectionStatusEnums

enum [DeviceConnectionStatusEnums](#)

< Indicates the status of the specified Connection.

Enumerator

DeviceConnectionStatus_Active	Connection is in use.
DeviceConnectionStatus_Inactive	Connection is not in use.
NUM_DEVICECONNECTIONSTATUS	

14.5.1.74 DeviceCurrentSpeedEnum

enum [DeviceCurrentSpeedEnum](#)

< The USB Speed that the device is currently operating at.

Enumerator

DeviceCurrentSpeed_UnknownSpeed	Unknown-Speed.
DeviceCurrentSpeed_LowSpeed	Low-Speed.
DeviceCurrentSpeed_FullSpeed	Full-Speed.
DeviceCurrentSpeed_HighSpeed	High-Speed.
DeviceCurrentSpeed_SuperSpeed	Super-Speed.
NUMDEVICECURRENTSPEED	

14.5.1.75 DeviceEndiannessMechanismEnum

enum [DeviceEndiannessMechanismEnum](#)

< Identifies the endianness handling mode.

Enumerator

DeviceEndiannessMechanism_Legacy	Handling the device endianness according to GenICam Schema 1.0
DeviceEndiannessMechanism_Standard	Handling the device endianness according to GenICam Schema 1.1 and later
NUMDEVICEENDIANESSMECHANISM	

14.5.1.76 DeviceIndicatorModeEnums

enum [DeviceIndicatorModeEnums](#)

< Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Enumerator

DeviceIndicatorMode_Inactive	
DeviceIndicatorMode_Active	
DeviceIndicatorMode_ErrorStatus	
NUM_DEVICEINDICATORMODE	

14.5.1.77 DeviceLinkHeartbeatModeEnums

enum [DeviceLinkHeartbeatModeEnums](#)

< Activate or deactivate the Link's heartbeat.

Enumerator

DeviceLinkHeartbeatMode_On	Enables the Link heartbeat.
DeviceLinkHeartbeatMode_Off	Disables the Link heartbeat.
NUM_DEVICELINKHEARTBEATMODE	

14.5.1.78 DeviceLinkThroughputLimitModeEnums

enum [DeviceLinkThroughputLimitModeEnums](#)

< Controls if the DeviceLinkThroughputLimit is active. When disabled, lower level TL specific features are expected to control the throughput. When enabled, DeviceLinkThroughputLimit controls the overall throughput.

Enumerator

DeviceLinkThroughputLimitMode_On	Enables the DeviceLinkThroughputLimit feature.
DeviceLinkThroughputLimitMode_Off	Disables the DeviceLinkThroughputLimit feature.
NUM_DEVICELINKTHROUGHPUTLIMITMODE	

14.5.1.79 DevicePowerSupplySelectorEnums

enum [DevicePowerSupplySelectorEnums](#)

< Selects the power supply source to control or read.

Enumerator

DevicePowerSupplySelector_External	
NUM_DEVICEPOWERSUPPLYSELECTOR	

14.5.1.80 DeviceRegistersEndiannessEnums

enum [DeviceRegistersEndiannessEnums](#)

< Endianness of the registers of the device.

Enumerator

DeviceRegistersEndianness_Little	
DeviceRegistersEndianness_Big	
NUM_DEVICEREGISTERSENDIANNESSE	

14.5.1.81 DeviceScanTypeEnums

enum [DeviceScanTypeEnums](#)

< Scan type of the sensor of the device.

Enumerator

DeviceScanType_Areascan	
NUM_DEVICESCANTYPE	

14.5.1.82 DeviceSerialPortBaudRateEnums

enum [DeviceSerialPortBaudRateEnums](#)

< This feature controls the baud rate used by the selected serial port.

Enumerator

DeviceSerialPortBaudRate_Baud9600	Serial port speed of 9600 baud.
DeviceSerialPortBaudRate_Baud19200	Serial port speed of 19200 baud.
DeviceSerialPortBaudRate_Baud38400	Serial port speed of 38400 baud.
DeviceSerialPortBaudRate_Baud57600	Serial port speed of 57600 baud.
DeviceSerialPortBaudRate_Baud115200	Serial port speed of 115200 baud.
DeviceSerialPortBaudRate_Baud230400	Serial port speed of 230400 baud.
DeviceSerialPortBaudRate_Baud460800	Serial port speed of 460800 baud.
DeviceSerialPortBaudRate_Baud921600	Serial port speed of 921600 baud.
NUM_DEVICESTERIALPORTBAUDRATE	

14.5.1.83 DeviceSerialPortSelectorEnums

enum [DeviceSerialPortSelectorEnums](#)

< Selects which serial port of the device to control.

Enumerator

DeviceSerialPortSelector_CameraLink	Serial port associated to the Camera link connection.
NUM_DEVICESTERIALPORTSELECTOR	

14.5.1.84 DeviceStreamChannelEndiannessEnums

enum [DeviceStreamChannelEndiannessEnums](#)

< Endianness of multi-byte pixel data for this stream.

Enumerator

DeviceStreamChannelEndianness_Big	Stream channel data is big Endian.
DeviceStreamChannelEndianness_Little	Stream channel data is little Endian.
NUM_DEVICESTREAMCHANNELENDIANNESS	

14.5.1.85 DeviceStreamChannelTypeEnums

enum [DeviceStreamChannelTypeEnums](#)

< Reports the type of the stream channel.

Enumerator

DeviceStreamChannelType_Transmitter	Data stream transmitter channel.
DeviceStreamChannelType_Receiver	Data stream receiver channel.
NUM_DEVICESTREAMCHANNELTYPE	

14.5.1.86 DeviceTapGeometryEnums

enum [DeviceTapGeometryEnums](#)

< This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Enumerator

DeviceTapGeometry_Geometry_1X_1Y	Geometry_1X_1Y
DeviceTapGeometry_Geometry_1X2_1Y	Geometry_1X2_1Y
DeviceTapGeometry_Geometry_1X2_1Y2	Geometry_1X2_1Y2
DeviceTapGeometry_Geometry_2X_1Y	Geometry_2X_1Y
DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y	Geometry_2X_1Y2Geometry_2XE_1Y
DeviceTapGeometry_Geometry_2XE_1Y2	Geometry_2XE_1Y2
DeviceTapGeometry_Geometry_2XM_1Y	Geometry_2XM_1Y
DeviceTapGeometry_Geometry_2XM_1Y2	Geometry_2XM_1Y2
DeviceTapGeometry_Geometry_1X_1Y2	Geometry_1X_1Y2
DeviceTapGeometry_Geometry_1X_2YE	Geometry_1X_2YE
DeviceTapGeometry_Geometry_1X3_1Y	Geometry_1X3_1Y

Enumerator

DeviceTapGeometry_Geometry_3X_1Y	Geometry_3X_1Y
DeviceTapGeometry_Geometry_1X	Geometry_1X
DeviceTapGeometry_Geometry_1X2	Geometry_1X2
DeviceTapGeometry_Geometry_2X	Geometry_2X
DeviceTapGeometry_Geometry_2XE	Geometry_2XE
DeviceTapGeometry_Geometry_2XM	Geometry_2XM
DeviceTapGeometry_Geometry_1X3	Geometry_1X3
DeviceTapGeometry_Geometry_3X	Geometry_3X
DeviceTapGeometry_Geometry_1X4_1Y	Geometry_1X4_1Y
DeviceTapGeometry_Geometry_4X_1Y	Geometry_4X_1Y
DeviceTapGeometry_Geometry_2X2_1Y	Geometry_2X2_1Y
DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y	Geometry_2X2E_1YGeometry_2X2M_1Y
DeviceTapGeometry_Geometry_1X2_2YE	Geometry_1X2_2YE
DeviceTapGeometry_Geometry_2X_2YE	Geometry_2X_2YE
DeviceTapGeometry_Geometry_2XE_2YE	Geometry_2XE_2YE
DeviceTapGeometry_Geometry_2XM_2YE	Geometry_2XM_2YE
DeviceTapGeometry_Geometry_1X4	Geometry_1X4
DeviceTapGeometry_Geometry_4X	Geometry_4X
DeviceTapGeometry_Geometry_2X2	Geometry_2X2
DeviceTapGeometry_Geometry_2X2E	Geometry_2X2E
DeviceTapGeometry_Geometry_2X2M	Geometry_2X2M
DeviceTapGeometry_Geometry_1X8_1Y	Geometry_1X8_1Y
DeviceTapGeometry_Geometry_8X_1Y	Geometry_8X_1Y
DeviceTapGeometry_Geometry_4X2_1Y	Geometry_4X2_1Y
DeviceTapGeometry_Geometry_2X2E_2YE	Geometry_2X2E_2YE
DeviceTapGeometry_Geometry_1X8	Geometry_1X8
DeviceTapGeometry_Geometry_8X	Geometry_8X
DeviceTapGeometry_Geometry_4X2	Geometry_4X2
DeviceTapGeometry_Geometry_4X2E	Geometry_4X2E
DeviceTapGeometry_Geometry_4X2E_1Y	Geometry_4X2E_1Y
DeviceTapGeometry_Geometry_1X10_1Y	Geometry_1X10_1Y
DeviceTapGeometry_Geometry_10X_1Y	Geometry_10X_1Y
DeviceTapGeometry_Geometry_1X10	Geometry_1X10
DeviceTapGeometry_Geometry_10X	Geometry_10X
NUM_DEVICE_TAPGEOMETRY	

14.5.1.87 DeviceTemperatureSelectorEnums

```
enum DeviceTemperatureSelectorEnums
```

< Selects the location within the device, where the temperature will be measured.

Enumerator

DeviceTemperatureSelector_Sensor	
NUM_DEVICETEMPERATURESELECTOR	

14.5.1.88 DeviceTLTypeEnums

enum [DeviceTLTypeEnums](#)

< Transport Layer type of the device.

Enumerator

DeviceTLType_GigEVision	
DeviceTLType_CameraLink	
DeviceTLType_CameraLinkHS	
DeviceTLType_CoaXPress	
DeviceTLType_USB3Vision	
DeviceTLType_Custom	
NUM_DEVICETLTYPE	

14.5.1.89 DeviceTypeEnum

enum [DeviceTypeEnum](#)

< Transport layer type of the device.

Enumerator

DeviceType_GigEVision	GigE Vision
DeviceType_CameraLink	Camera Link
DeviceType_CameraLinkHS	Camera Link High Speed
DeviceType_CoaXPress	CoaXPress
DeviceType_USB3Vision	USB3 Vision
DeviceType_Custom	Custom transport layer
NUMDEVICETYPE	

14.5.1.90 DeviceTypeEnums

enum [DeviceTypeEnums](#)

< Returns the device type.

Enumerator

DeviceType_Transmitter	Data stream transmitter device.
DeviceType_Receiver	Data stream receiver device.
DeviceType_Transceiver	Data stream receiver and transmitter device.
DeviceType_Peripheral	Controllable device (with no data stream handling).
NUM_DEVICETYPE	

14.5.1.91 EncoderModeEnums

enum [EncoderModeEnums](#)

< Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Enumerator

EncoderMode_FourPhase	The counter increments or decrements 1 for every full quadrature cycle with jitter filtering.
EncoderMode_HighResolution	The counter increments or decrements every quadrature phase for high resolution counting, but without jitter filtering.
NUM_ENCODERMODE	

14.5.1.92 EncoderOutputModeEnums

enum [EncoderOutputModeEnums](#)

< Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Enumerator

EncoderOutputMode_Off	No output pulse are generated.
EncoderOutputMode_PositionUp	Output pulses are generated at all new positions in the positive direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.
EncoderOutputMode_PositionDown	Output pulses are generated at all new positions in the negative direction. If the encoder reverses no output pulse are generated until it has again passed the position where the reversal started.
EncoderOutputMode_DirectionUp	Output pulses are generated at all position increments in the positive direction while ignoring negative direction motion.
EncoderOutputMode_DirectionDown	Output pulses are generated at all position increments in the negative direction while ignoring positive direction motion.
EncoderOutputMode_Motion	Output pulses are generated at all motion increments in both directions.
NUM_ENCODEROUTPUTMODE	

14.5.1.93 EncoderResetActivationEnums

enum [EncoderResetActivationEnums](#)

< Selects the Activation mode of the Encoder Reset Source signal.

Enumerator

EncoderResetActivation_RisingEdge	Resets the Encoder on the Rising Edge of the signal.
EncoderResetActivation_FallingEdge	Resets the Encoder on the Falling Edge of the signal.
EncoderResetActivation_AnyEdge	Resets the Encoder on the Falling or rising Edge of the selected signal.
EncoderResetActivation_LevelHigh	Resets the Encoder as long as the selected signal level is High.
EncoderResetActivation_LevelLow	Resets the Encoder as long as the selected signal level is Low.
NUM_ENCODERRESETACTIVATION	

14.5.1.94 EncoderResetSourceEnums

enum [EncoderResetSourceEnums](#)

< Selects the signals that will be the source to reset the Encoder.

Enumerator

EncoderResetSource_Off	Disable the Encoder Reset trigger.
EncoderResetSource_AcquisitionTrigger	Resets with the reception of the Acquisition Trigger.
EncoderResetSource_AcquisitionStart	Resets with the reception of the Acquisition Start.
EncoderResetSource_AcquisitionEnd	Resets with the reception of the Acquisition End.
EncoderResetSource_FrameTrigger	Resets with the reception of the Frame Start Trigger.
EncoderResetSource_FrameStart	Resets with the reception of the Frame Start.
EncoderResetSource_FrameEnd	Resets with the reception of the Frame End.
EncoderResetSource_ExposureStart	Resets with the reception of the Exposure Start.
EncoderResetSource_ExposureEnd	Resets with the reception of the Exposure End.
EncoderResetSource_Line0	Resets by the chosen I/O Line.
EncoderResetSource_Line1	Resets by the chosen I/O Line.
EncoderResetSource_Line2	Resets by the chosen I/O Line.
EncoderResetSource_Counter0Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter1Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter2Start	Resets with the reception of the Counter Start.
EncoderResetSource_Counter0End	Resets with the reception of the Counter End.
EncoderResetSource_Counter1End	Resets with the reception of the Counter End.
EncoderResetSource_Counter2End	Resets with the reception of the Counter End.
EncoderResetSource_Timer0Start	Resets with the reception of the Timer Start.
EncoderResetSource_Timer1Start	Resets with the reception of the Timer Start.
EncoderResetSource_Timer2Start	Resets with the reception of the Timer Start.

Enumerator

EncoderResetSource_Timer0End	Resets with the reception of the Timer End.
EncoderResetSource_Timer1End	Resets with the reception of the Timer End.
EncoderResetSource_Timer2End	Resets with the reception of the Timer End.
EncoderResetSource_UserOutput0	Resets by the chosen User Output bit.
EncoderResetSource_UserOutput1	Resets by the chosen User Output bit.
EncoderResetSource_UserOutput2	Resets by the chosen User Output bit.
EncoderResetSource_SoftwareSignal0	Resets on the reception of the Software Signal.
EncoderResetSource_SoftwareSignal1	Resets on the reception of the Software Signal.
EncoderResetSource_SoftwareSignal2	Resets on the reception of the Software Signal.
EncoderResetSource_Action0	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_Action1	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_Action2	Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer).
EncoderResetSource_LinkTrigger0	Resets on the reception of the chosen Link Trigger (received from the transport layer).
EncoderResetSource_LinkTrigger1	Resets on the reception of the chosen Link Trigger (received from the transport layer).
EncoderResetSource_LinkTrigger2	Resets on the reception of the chosen Link Trigger (received from the transport layer).
NUM_ENCODERRESETSOURCE	

14.5.1.95 EncoderSelectorEnums

```
enum EncoderSelectorEnums
```

< Selects which Encoder to configure.

Enumerator

EncoderSelector_Encoder0	Selects Encoder 0.
EncoderSelector_Encoder1	Selects Encoder 1.
EncoderSelector_Encoder2	Selects Encoder 2.
NUM_ENCODERSELECTOR	

14.5.1.96 EncoderSourceAEnums

```
enum EncoderSourceAEnums
```

< Selects the signal which will be the source of the A input of the Encoder.

Enumerator

EncoderSourceA_Off	Counter is stopped.
EncoderSourceA_Line0	Encoder Forward input is taken from the chosen I/O Line.
EncoderSourceA_Line1	Encoder Forward input is taken from the chosen I/O Line.
EncoderSourceA_Line2	Encoder Forward input is taken from the chosen I/O Line.
NUM_ENCODERSOURCEA	

14.5.1.97 EncoderSourceBEnums

enum [EncoderSourceBEnums](#)

< Selects the signal which will be the source of the B input of the Encoder.

Enumerator

EncoderSourceB_Off	Counter is stopped.
EncoderSourceB_Line0	Encoder Reverse input is taken from the chosen I/O Line..
EncoderSourceB_Line1	Encoder Reverse input is taken from the chosen I/O Line..
EncoderSourceB_Line2	Encoder Reverse input is taken from the chosen I/O Line..
NUM_ENCODERSOURCEB	

14.5.1.98 EncoderStatusEnums

enum [EncoderStatusEnums](#)

< Returns the motion status of the encoder.

Enumerator

EncoderStatus_EncoderUp	The encoder counter last incremented.
EncoderStatus_EncoderDown	The encoder counter last decremented.
EncoderStatus_EncoderIdle	The encoder is not active.
EncoderStatus_EncoderStatic	No motion within the EncoderTimeout time.
NUM_ENCODERSTATUS	

14.5.1.99 Error

enum [Error](#)

[Spinnaker](#) enum definitions.

The error codes used in [Spinnaker](#). These codes are returned as part of [Spinnaker::Exception](#). The error codes in the range of -1000 to -1999 are reserved for exceptions that map directly to GenTL values. The error codes in the range of -2000 to -2999 are reserved for [GenICam](#) related errors. The error codes in the range of -3000 to -3999 are reserved for image processing related errors.

Enumerator

SPINNAKER_ERR_SUCCESS	
SPINNAKER_ERR_ERROR	
SPINNAKER_ERR_NOT_INITIALIZED	
SPINNAKER_ERR_NOT_IMPLEMENTED	
SPINNAKER_ERR_RESOURCE_IN_USE	
SPINNAKER_ERR_ACCESS_DENIED	
SPINNAKER_ERR_INVALID_HANDLE	
SPINNAKER_ERR_INVALID_ID	
SPINNAKER_ERR_NO_DATA	
SPINNAKER_ERR_INVALID_PARAMETER	
SPINNAKER_ERR_IO	
SPINNAKER_ERR_TIMEOUT	
SPINNAKER_ERR_ABORT	
SPINNAKER_ERR_INVALID_BUFFER	
SPINNAKER_ERR_NOT_AVAILABLE	
SPINNAKER_ERR_INVALID_ADDRESS	
SPINNAKER_ERR_BUFFER_TOO_SMALL	
SPINNAKER_ERR_INVALID_INDEX	
SPINNAKER_ERR_PARSING_CHUNK_DATA	
SPINNAKER_ERR_INVALID_VALUE	
SPINNAKER_ERR_RESOURCE_EXHAUSTED	
SPINNAKER_ERR_OUT_OF_MEMORY	
SPINNAKER_ERR_BUSY	
SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT	
SPINNAKER_ERR_GENICAM_OUT_OF_RANGE	
SPINNAKER_ERR_GENICAM_PROPERTY	
SPINNAKER_ERR_GENICAM_RUN_TIME	
SPINNAKER_ERR_GENICAM_LOGICAL	
SPINNAKER_ERR_GENICAM_ACCESS	
SPINNAKER_ERR_GENICAM_TIMEOUT	
SPINNAKER_ERR_GENICAM_DYNAMIC_CAST	
SPINNAKER_ERR_GENICAM_GENERIC	
SPINNAKER_ERR_GENICAM_BAD_ALLOCATION	
SPINNAKER_ERR_IM_CONVERT	
SPINNAKER_ERR_IM_COPY	
SPINNAKER_ERR_IM_MALLOC	
SPINNAKER_ERR_IM_NOT_SUPPORTED	
SPINNAKER_ERR_IM_HISTOGRAM_RANGE	
SPINNAKER_ERR_IM_HISTOGRAM_MEAN	
SPINNAKER_ERR_IM_MIN_MAX	
SPINNAKER_ERR_IM_COLOR_CONVERSION	
SPINNAKER_ERR_IM_DECOMPRESSION	
SPINNAKER_ERR_CUSTOM_ID	

14.5.1.100 EventNotificationEnums

enum [EventNotificationEnums](#)

< Enables/Disables the selected event.

Enumerator

EventNotification_On	
EventNotification_Off	
NUM_EVENTNOTIFICATION	

14.5.1.101 EventSelectorEnums

enum [EventSelectorEnums](#)

< Selects which Event to enable or disable.

Enumerator

EventSelector_Error	
EventSelector_ExposureEnd	
EventSelector_SerialPortReceive	
NUM_EVENTSELECTOR	

14.5.1.102 EventType

enum [EventType](#)

Event types in [Spinnaker](#).

See also

[EventHandler::GetEventType\(\)](#)

Enumerator

SPINNAKER_EVENT_ARRIVAL_REMOVAL	
SPINNAKER_EVENT_DEVICE	
SPINNAKER_EVENT_DEVICE_SPECIFIC	
SPINNAKER_EVENT_NEW_BUFFER	
SPINNAKER_EVENT_NEW_BUFFER_LIST	
SPINNAKER_EVENT_LOGGING_EVENT	
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL	
SPINNAKER_EVENT_UNKNOWN	

14.5.1.103 ExposureActiveModeEnums

enum [ExposureActiveModeEnums](#)

< Control sensor active exposure mode.

Enumerator

ExposureActiveMode_Line1	
ExposureActiveMode_AnyPixels	
ExposureActiveMode_AllPixels	
NUM_EXPOSUREACTIVEMODE	

14.5.1.104 ExposureAutoEnums

enum [ExposureAutoEnums](#)

< Sets the automatic exposure mode

Enumerator

ExposureAuto_Off	Exposure time is manually controlled using ExposureTime
ExposureAuto_Once	Exposure time is adapted once by the device. Once it has converged, it returns to the Off state.
ExposureAuto_Continuous	Exposure time is constantly adapted by the device to maximize the dynamic range.
NUM_EXPOSUREAUTO	

14.5.1.105 ExposureModeEnums

enum [ExposureModeEnums](#)

< Sets the operation mode of the Exposure.

Enumerator

ExposureMode_Timed	Timed exposure. The exposure time is set using the ExposureTime or ExposureAuto features and the exposure starts with the FrameStart or LineStart.
ExposureMode_TriggerWidth	Uses the width of the current Frame trigger signal pulse to control the exposure time.
NUM_EXPOSUREMODE	

14.5.1.106 ExposureTimeModeEnums

enum [ExposureTimeModeEnums](#)

< Sets the configuration mode of the ExposureTime feature.

Enumerator

ExposureTimeMode_Common	The exposure time is common to all the color components. The common ExposureTime value to use can be set selecting it with ExposureTimeSelector[Common].
ExposureTimeMode_Individual	The exposure time is individual for each color component. Each individual ExposureTime values to use can be set by selecting them with ExposureTimeSelector.
NUM_EXPOSURETIMEMODE	

14.5.1.107 ExposureTimeSelectorEnums

enum [ExposureTimeSelectorEnums](#)

< Selects which exposure time is controlled by the ExposureTime feature. This allows for independent control over the exposure components.

Enumerator

ExposureTimeSelector_Common	Selects the common ExposureTime.
ExposureTimeSelector_Red	Selects the red common ExposureTime.
ExposureTimeSelector_Green	Selects the green ExposureTime.
ExposureTimeSelector_Blue	Selects the blue ExposureTime.
ExposureTimeSelector_Cyan	Selects the cyan common ExposureTime.
ExposureTimeSelector_Magenta	Selects the magenta ExposureTime.
ExposureTimeSelector_Yellow	Selects the yellow ExposureTime.
ExposureTimeSelector_Infrared	Selects the infrared ExposureTime.
ExposureTimeSelector_Ultraviolet	Selects the ultraviolet ExposureTime.
ExposureTimeSelector_Stage1	Selects the first stage ExposureTime.
ExposureTimeSelector_Stage2	Selects the second stage ExposureTime.
NUM_EXPOSURETIMESELECTOR	

14.5.1.108 FileOpenModeEnums

enum [FileOpenModeEnums](#)

< The mode of the file when it is opened. The file can be opened for reading, writing or both. This must be set before opening the file.

Enumerator

FileOpenMode_Read	
FileOpenMode_Write	
FileOpenMode_ReadWrite	
NUM_FILEOPENMODE	

14.5.1.109 FileOperationSelectorEnums

enum `FileOperationSelectorEnums`

< Sets operation to execute on the selected file when the execute command is given.

Enumerator

FileOperationSelector_Open	
FileOperationSelector_Close	
FileOperationSelector_Read	
FileOperationSelector_Write	
FileOperationSelector_Delete	
NUM_FILEOPERATIONSELECTOR	

14.5.1.110 FileOperationStatusEnums

enum `FileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

FileOperationStatus_Success	File Operation was sucessful.
FileOperationStatus_Failure	File Operation failed.
FileOperationStatus_Overflow	An overflow occurred while executing the File Operation.
NUM_FILEOPERATIONSTATUS	

14.5.1.111 FileSelectorEnums

enum `FileSelectorEnums`

< Selects which file is being operated on. This must be set before performing any file operations.

Enumerator

FileSelector_UserSetDefault	
FileSelector_UserSet0	
FileSelector_UserSet1	
FileSelector_UserFile1	
FileSelector_SerialPort0	
NUM_FILESELECTOR	

14.5.1.112 FilterDriverStatusEnum

enum `FilterDriverStatusEnum`

< Reports whether FLIR Light Weight Filter Driver is enabled, disabled, or not installed.

Enumerator

FilterDriverStatus_NotSupported	Not Installed
FilterDriverStatus_Disabled	FLIR Light Weight Filter Driver is disabled across all interfaces
FilterDriverStatus_Enabled	FLIR Light Weight Filter Driver is enabled
NUMFILTERDRIVERSTATUS	

14.5.1.113 GainAutoBalanceEnums

enum `GainAutoBalanceEnums`

< Sets the mode for automatic gain balancing between the sensor color channels or taps. The gain coefficients of each channel or tap are adjusted so they are matched.

Enumerator

GainAutoBalance_Off	Gain tap balancing is user controlled using Gain .
GainAutoBalance_Once	Gain tap balancing is automatically adjusted once by the device. Once it has converged, it automatically returns to the Off state.
GainAutoBalance_Continuous	Gain tap balancing is constantly adjusted by the device.
NUM_GAINAUTOBALANCE	

14.5.1.114 GainAutoEnums

enum `GainAutoEnums`

< Sets the automatic gain mode. Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range.

Enumerator

GainAuto_Off	Gain is manually controlled
GainAuto_Once	Gain is adapted once by the device. Once it has converged, it returns to the Off state.
GainAuto_Continuous	Gain is constantly adapted by the device to maximize the dynamic range.
NUM_GAINAUTO	

14.5.1.115 GainSelectorEnums

enum [GainSelectorEnums](#)

< Selects which gain to control. The All selection is a total amplification across all channels (or taps).

Enumerator

GainSelector_All	
NUM_GAINSELECTOR	

14.5.1.116 GenICamXMLLocationEnum

enum [GenICamXMLLocationEnum](#)

< Sets the location to load [GenICam](#) XML.

Enumerator

GenICamXMLLocation_Device	Load GenICam XML from device
GenICamXMLLocation_Host	Load GenICam XML from host
NUMGENICAMXMLLOCATION	

14.5.1.117 GevCCPEnum

enum [GevCCPEnum](#)

< Controls the device access privilege of an application.

Enumerator

GevCCP_EnumEntry_GevCCP_OpenAccess	Open access privilege.
GevCCP_EnumEntry_GevCCP_ExclusiveAccess	Exclusive access privilege.
GevCCP_EnumEntry_GevCCP_ControlAccess	Control access privilege.
NUMGEVCCP	

14.5.1.118 GevCCPEnums

enum [GevCCPEnums](#)

< Controls the device access privilege of an application.

Enumerator

GevCCP_OpenAccess	
GevCCP_ExclusiveAccess	
GevCCP_ControlAccess	
NUM_GEVCCP	

14.5.1.119 GevCurrentPhysicalLinkConfigurationEnums

enum [GevCurrentPhysicalLinkConfigurationEnums](#)

< Indicates the current physical link configuration of the device.

Enumerator

GevCurrentPhysicalLinkConfiguration_SingleLink	Single Link
GevCurrentPhysicalLinkConfiguration_MultiLink	Multi Link
GevCurrentPhysicalLinkConfiguration_StaticLAG	Static LAG
GevCurrentPhysicalLinkConfiguration_DynamicLAG	Dynamic LAG
NUM_GEVCURRENTPHYSICALLINKCONFIGURATION	

14.5.1.120 GevGVCPExtendedStatusCodesSelectorEnums

enum [GevGVCPExtendedStatusCodesSelectorEnums](#)

< Selects the GigE Vision version to control extended status codes for.

Enumerator

GevGVCPExtendedStatusCodesSelector_Version1_1	Version 1 1
GevGVCPExtendedStatusCodesSelector_Version2_0	Version 2 0
NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR	

14.5.1.121 GevGVSPExtendedIDModeEnums

enum [GevGVSPExtendedIDModeEnums](#)

< Enables the extended IDs mode.

Enumerator

GevGVSPExtendedIDMode_Off	Off
GevGVSPExtendedIDMode_On	On
NUM_GEVGVSPEXTENDEDIDMODE	

14.5.1.122 GevIEEE1588ClockAccuracyEnums

enum [GevIEEE1588ClockAccuracyEnums](#)

< Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

Enumerator

GevIEEE1588ClockAccuracy_Unknown	Unknown Accuracy
NUM_GEVIEEE1588CLOCKACCURACY	

14.5.1.123 GevIEEE1588ModeEnums

enum [GevIEEE1588ModeEnums](#)

< Provides the mode of the IEEE 1588 clock.

Enumerator

GevIEEE1588Mode_Auto	Automatic
GevIEEE1588Mode_SlaveOnly	Slave Only
NUM_GEVIEEE1588MODE	

14.5.1.124 GevIEEE1588StatusEnums

enum [GevIEEE1588StatusEnums](#)

< Provides the status of the IEEE 1588 clock.

Enumerator

GevIEEE1588Status_Initializing	Initializing
GevIEEE1588Status_Faulty	Faulty
GevIEEE1588Status_Disabled	Disabled
GevIEEE1588Status_Listening	Listening
GevIEEE1588Status_PreMaster	Pre Master
GevIEEE1588Status_Master	Master
GevIEEE1588Status_Passive	Passive
GevIEEE1588Status_Uncalibrated	Uncalibrated
GevIEEE1588Status_Slave	Slave
NUM_GEVIEEE1588STATUS	

14.5.1.125 GevIPConfigurationStatusEnums

enum [GevIPConfigurationStatusEnums](#)

< Reports the current IP configuration status.

Enumerator

GevIPConfigurationStatus_None	None
GevIPConfigurationStatus_PersistentIP	Persistent IP
GevIPConfigurationStatus_DHCP	DHCP
GevIPConfigurationStatus_LLA	LLA
GevIPConfigurationStatus_ForceIP	Force IP
NUM_GEVIPCONFIGURATIONSTATUS	

14.5.1.126 GevPhysicalLinkConfigurationEnums

enum [GevPhysicalLinkConfigurationEnums](#)

< Controls the principal physical link configuration to use on next restart/power-up of the device.

Enumerator

GevPhysicalLinkConfiguration_SingleLink	Single Link
GevPhysicalLinkConfiguration_MultiLink	Multi Link
GevPhysicalLinkConfiguration_StaticLAG	Static LAG
GevPhysicalLinkConfiguration_DynamicLAG	Dynamic LAG
NUM_GEVPHYSICALLINKCONFIGURATION	

14.5.1.127 GevSupportedOptionSelectorEnums

enum [GevSupportedOptionSelectorEnums](#)

< Selects the GEV option to interrogate for existing support.

Enumerator

GevSupportedOptionSelector_UserDefinedName	
GevSupportedOptionSelector_SerialNumber	
GevSupportedOptionSelector_HeartbeatDisable	
GevSupportedOptionSelector_LinkSpeed	
GevSupportedOptionSelector_CCPApplicationSocket	
GevSupportedOptionSelector_ManifestTable	
GevSupportedOptionSelector_TestData	
GevSupportedOptionSelector_DiscoveryAckDelay	
GevSupportedOptionSelector_DiscoveryAckDelayWritable	
GevSupportedOptionSelector_ExtendedStatusCodes	
GevSupportedOptionSelector_Action	
GevSupportedOptionSelector_PendingAck	
GevSupportedOptionSelector_EventData	
GevSupportedOptionSelector_Event	
GevSupportedOptionSelector_PacketResend	
GevSupportedOptionSelector_WriteMem	
GevSupportedOptionSelector_CommandsConcatenation	
GevSupportedOptionSelector_IPConfigurationLLA	
GevSupportedOptionSelector_IPConfigurationDHCP	
GevSupportedOptionSelector_IPConfigurationPersistentIP	
GevSupportedOptionSelector_StreamChannelSourceSocket	
GevSupportedOptionSelector_MessageChannelSourceSocket	
NUM_GEVSUPPORTEDOPTIONSELECTOR	

14.5.1.128 GUIXMLLocationEnum

enum [GUIXMLLocationEnum](#)

< Sets the location to load GUI XML.

Enumerator

GUIXMLLocation_Device	Load XML from device
GUIXMLLocation_Host	Load XML from host
NUMGUIXMLLOCATION	

14.5.1.129 HeatmapColor

enum [HeatmapColor](#)

Color specifiers for the heatmap color gradient.

Enumerator

SPINNAKER_HEATMAP_COLOR_BLACK	
SPINNAKER_HEATMAP_COLOR_BLUE	
SPINNAKER_HEATMAP_COLOR_CYAN	
SPINNAKER_HEATMAP_COLOR_GREEN	
SPINNAKER_HEATMAP_COLOR_YELLOW	
SPINNAKER_HEATMAP_COLOR_RED	
SPINNAKER_HEATMAP_COLOR_WHITE	

14.5.1.130 ImageComponentSelectorEnums

enum [ImageComponentSelectorEnums](#)

< Selects a component to activate data streaming from.

Enumerator

ImageComponentSelector_Intensity	The acquisition of intensity of the reflected light is controlled.
ImageComponentSelector_Color	The acquisition of color of the reflected light is controlled
ImageComponentSelector_Infrared	The acquisition of non-visible infrared light is controlled.
ImageComponentSelector_Ultraviolet	The acquisition of non-visible ultraviolet light is controlled.
ImageComponentSelector_Range	The acquisition of range (distance) data is controlled. The data produced may be only range (2.5D) or a point cloud 3D coordinates depending on the Scan3dControl.
ImageComponentSelector_Disparity	The acquisition of stereo camera disparity data is controlled. Disparity is a more specific range format approximately inversely proportional to distance. Disparity is typically given in pixel units.
ImageComponentSelector_Confidence	The acquisition of confidence map of the acquired image is controlled. Confidence data may be binary (0 - invalid) or an integer where 0 is invalid and increasing value is increased confidence in the data in the corresponding pixel. If floating point representation is used the confidence image is normalized to the range [0,1], for integer representation the maximum possible integer represents maximum confidence.

Enumerator

ImageComponentSelector_Scatter	The acquisition of data measuring how much light is scattered around the reflected light. In processing this is used as an additional intensity image, often together with the standard intensity.
NUM_IMAGECOMPONENTSELECTOR	

14.5.1.131 ImageCompressionJPEGFormatOptionEnums

enum [ImageCompressionJPEGFormatOptionEnums](#)

< When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Enumerator

ImageCompressionJPEGFormatOption_Lossless	Selects lossless JPEG compression based on a predictive coding model.
ImageCompressionJPEGFormatOption_Baseline↔ Standard	Indicates this is a baseline sequential (single-scan) DCT-based JPEG.
ImageCompressionJPEGFormatOption_Baseline↔ Optimized	Provides optimized color and slightly better compression than baseline standard by using custom Huffman tables optimized after statistical analysis of the image content.
ImageCompressionJPEGFormatOption_Progressive	Indicates this is a progressive (multi-scan) DCT-based JPEG.
NUM_IMAGECOMPRESSIONJPEGFORMATOPT↔ ION	

14.5.1.132 ImageCompressionModeEnums

enum [ImageCompressionModeEnums](#)

<

Enumerator

ImageCompressionMode_Off	
ImageCompressionMode_Lossless	
NUM_IMAGECOMPRESSIONMODE	

14.5.1.133 ImageCompressionRateOptionEnums

enum [ImageCompressionRateOptionEnums](#)

< Two rate controlling options are offered: fixed bit rate or fixed quality. The exact implementation to achieve one or the other is vendor-specific.

Enumerator

ImageCompressionRateOption_FixBitrate	Output stream follows a constant bit rate. Allows easy bandwidth management on the link.
ImageCompressionRateOption_FixQuality	Output stream has a constant image quality. Can be used when image processing algorithms are sensitive to image degradation caused by excessive data compression.
NUM_IMAGECOMPRESSIONRATEOPTION	

14.5.1.134 ImageFileFormat

enum [ImageFileFormat](#)

File formats to be used for saving images to disk.

Enumerator

SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT	Determine file format from file extension.
SPINNAKER_IMAGE_FILE_FORMAT_PGM	Portable gray map.
SPINNAKER_IMAGE_FILE_FORMAT_PPM	Portable pixmap.
SPINNAKER_IMAGE_FILE_FORMAT_BMP	Bitmap.
SPINNAKER_IMAGE_FILE_FORMAT_JPEG	JPEG.
SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000	JPEG 2000.
SPINNAKER_IMAGE_FILE_FORMAT_TIFF	Tagged image file format.
SPINNAKER_IMAGE_FILE_FORMAT_PNG	Portable network graphics.
SPINNAKER_IMAGE_FILE_FORMAT_RAW	Raw data.
SPINNAKER_IMAGE_FILE_FORMAT_JPEG12_C	12 bit compressed JPEG data.
SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS	

14.5.1.135 ImagePayloadType

enum [ImagePayloadType](#)

Enumerator

SPINNAKER_IMAGE_PAYLOAD_TYPE_UNKNOWN	
SPINNAKER_IMAGE_PAYLOAD_TYPE_IMAGE	
SPINNAKER_IMAGE_PAYLOAD_TYPE_EXTENDED_CHUNK	

Enumerator

SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG	
SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSLESS_COMPRESSED	
SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSY_COMPRESSED	
SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED	

14.5.1.136 ImageScalingAlgorithm

enum [ImageScalingAlgorithm](#)

[Image](#) scaling algorithms.

Enumerator

SPINNAKER_IMAGE_SCALING_ALGORITHM_NEAREST_NEIGHBOR	

14.5.1.137 ImageStatus

enum [ImageStatus](#)

Status of images returned from `GetNextImage()` or `Convert()` calls.

Enumerator

SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR OR	Image has an unknown error.
SPINNAKER_IMAGE_STATUS_NO_ERROR	Image is returned from <code>GetNextImage()</code> or <code>Convert()</code> calls without any errors.
SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED	Image failed CRC check.
SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW	Received more data than the size of the image.
SPINNAKER_IMAGE_STATUS_MISSING_PACKETS	Image has missing packets. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see https://www.flir.com/support-center/iis/machine-vision/application
SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT	Image leader is incomplete. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT	Image trailer is incomplete. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT	Image has an inconsistent packet id. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_MISSING_LEADER	Image leader is missing. Could be caused by missing packet(s). See link above.

Enumerator

SPINNAKER_IMAGE_STATUS_MISSING_TRAILER	Image trailer is missing. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE	Image data is incomplete. Could be caused by missing packet(s) or decompression error. See link above.
SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT	Image info is corrupted. Could be caused by missing packet(s). See link above.
SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID	Image chunk data is invalid.
SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES	Image cannot be processed due to lack of system resources.

14.5.1.138 InferenceBoxType

enum [InferenceBoxType](#)

Inference Bounding Box Type.

Enumerator

INFERENCE_BOX_TYPE_RECTANGLE	
INFERENCE_BOX_TYPE_CIRCLE	
INFERENCE_BOX_TYPE_ROTATED_RECTANGLE	

14.5.1.139 InterfaceTypeEnum

enum [InterfaceTypeEnum](#)

< Transport layer type of the interface.

Enumerator

InterfaceType_GigEVision	GigE Vision
InterfaceType_CameraLink	Camera Link
InterfaceType_CameraLinkHS	Camera Link High Speed
InterfaceType_CoaXPress	CoaXPress
InterfaceType_USB3Vision	USB3 Vision
InterfaceType_Custom	Custom transport layer
NUMINTERFACETYPE	

14.5.1.140 LineFormatEnums

enum [LineFormatEnums](#)

< Displays the current electrical format of the selected physical input or output Line.

Enumerator

LineFormat_NoConnect	
LineFormat_TriState	
LineFormat_TTL	
LineFormat_LVDS	
LineFormat_RS422	
LineFormat_OptoCoupled	
LineFormat_OpenDrain	
NUM_LINEFORMAT	

14.5.1.141 LineInputFilterSelectorEnums

enum [LineInputFilterSelectorEnums](#)

< Selects the kind of input filter to configure: Deglitch or Debounce.

Enumerator

LineInputFilterSelector_Deglitch	
LineInputFilterSelector_Debounce	
NUM_LINEINPUTFILTERSELECTOR	

14.5.1.142 LineModeEnums

enum [LineModeEnums](#)

< Controls if the physical Line is used to Input or Output a signal.

Enumerator

LineMode_Input	
LineMode_Output	
NUM_LINEMODE	

14.5.1.143 LineSelectorEnums

enum [LineSelectorEnums](#)

< Selects the physical line (or pin) of the external device connector to configure

Enumerator

LineSelector_Line0	
LineSelector_Line1	
LineSelector_Line2	
LineSelector_Line3	
NUM_LINESELECTOR	

14.5.1.144 LineSourceEnums

enum [LineSourceEnums](#)

< Selects which internal acquisition or I/O source signal to output on the selected line. LineMode must be Output.

Enumerator

LineSource_Off	
LineSource_Line0	
LineSource_Line1	
LineSource_Line2	
LineSource_Line3	
LineSource_UserOutput0	
LineSource_UserOutput1	
LineSource_UserOutput2	
LineSource_UserOutput3	
LineSource_Counter0Active	
LineSource_Counter1Active	
LineSource_LogicBlock0	
LineSource_LogicBlock1	
LineSource_ExposureActive	
LineSource_FrameTriggerWait	
LineSource_SerialPort0	
LineSource_PPSSignal	
LineSource_AllPixel	
LineSource_AnyPixel	
NUM_LINESOURCE	

14.5.1.145 LogicBlockLUTInputActivationEnums

enum [LogicBlockLUTInputActivationEnums](#)

< Selects the activation mode of the Logic Input Source signal.

Enumerator

LogicBlockLUTInputActivation_LevelLow	
LogicBlockLUTInputActivation_LevelHigh	
LogicBlockLUTInputActivation_FallingEdge	
LogicBlockLUTInputActivation_RisingEdge	
LogicBlockLUTInputActivation_AnyEdge	
NUM_LOGICBLOCKLUTINPUTACTIVATION	

14.5.1.146 LogicBlockLUTInputSelectorEnums

enum [LogicBlockLUTInputSelectorEnums](#)

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

LogicBlockLUTInputSelector_Input0	
LogicBlockLUTInputSelector_Input1	
LogicBlockLUTInputSelector_Input2	
LogicBlockLUTInputSelector_Input3	
NUM_LOGICBLOCKLUTINPUTSELECTOR	

14.5.1.147 LogicBlockLUTInputSourceEnums

enum [LogicBlockLUTInputSourceEnums](#)

< Selects the source for the input into the Logic LUT.

Enumerator

LogicBlockLUTInputSource_Zero	Zero
LogicBlockLUTInputSource_Line0	Line0
LogicBlockLUTInputSource_Line1	Line1
LogicBlockLUTInputSource_Line2	Line2
LogicBlockLUTInputSource_Line3	Line3
LogicBlockLUTInputSource_UserOutput0	UserOutput0
LogicBlockLUTInputSource_UserOutput1	UserOutput1

Enumerator

LogicBlockLUTInputSource_UserOutput2	UserOutput2
LogicBlockLUTInputSource_UserOutput3	UserOutput3
LogicBlockLUTInputSource_Counter0Start	Counter0Start
LogicBlockLUTInputSource_Counter1Start	Counter1Start
LogicBlockLUTInputSource_Counter0End	Counter0End
LogicBlockLUTInputSource_Counter1End	Counter1End
LogicBlockLUTInputSource_LogicBlock0	LogicBlock0
LogicBlockLUTInputSource_LogicBlock1	LogicBlock1
LogicBlockLUTInputSource_ExposureStart	ExposureStart
LogicBlockLUTInputSource_ExposureEnd	ExposureEnd
LogicBlockLUTInputSource_FrameTriggerWait	FrameTriggerWait
LogicBlockLUTInputSource_AcquisitionActive	AcquisitionActive
NUM_LOGICBLOCKLUTINPUTSOURCE	

14.5.1.148 LogicBlockLUTSelectorEnums

```
enum LogicBlockLUTSelectorEnums
```

< Selects which LogicBlock LUT to configure

Enumerator

LogicBlockLUTSelector_Value	
LogicBlockLUTSelector_Enable	
NUM_LOGICBLOCKLUTSELECTOR	

14.5.1.149 LogicBlockSelectorEnums

```
enum LogicBlockSelectorEnums
```

< Selects which LogicBlock to configure

Enumerator

LogicBlockSelector_LogicBlock0	
LogicBlockSelector_LogicBlock1	
NUM_LOGICBLOCKSELECTOR	

14.5.1.150 LUTSelectorEnums

enum `LUTSelectorEnums`

The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.

< Selects which LUT to control.

Enumerator

LUTSelector_LUT1	This LUT is for re-mapping pixels of all formats (mono, Bayer, red, green and blue).
NUM_LUTSELECTOR	

14.5.1.151 PixelColorFilterEnums

enum `PixelColorFilterEnums`

< Type of color filter that is applied to the image. Only applies to Bayer pixel formats. All others have no color filter.

Enumerator

PixelColorFilter_None	No color filter.
PixelColorFilter_BayerRG	Bayer Red Green filter.
PixelColorFilter_BayerGB	Bayer Green Blue filter.
PixelColorFilter_BayerGR	Bayer Green Red filter.
PixelColorFilter_BayerBG	Bayer Blue Green filter.
NUM_PIXELCOLORFILTER	

14.5.1.152 PixelFormatEnums

enum `PixelFormatEnums`

< Format of the pixel provided by the camera.

Enumerator

PixelFormat_Mono8	
PixelFormat_Mono16	
PixelFormat_RGB8Packed	
PixelFormat_BayerGR8	
PixelFormat_BayerRG8	
PixelFormat_BayerGB8	
PixelFormat_BayerBG8	
PixelFormat_BayerGR16	
PixelFormat_BayerRG16	

Enumerator

PixelFormat_BayerGB16	
PixelFormat_BayerBG16	
PixelFormat_Mono12Packed	
PixelFormat_BayerGR12Packed	
PixelFormat_BayerRG12Packed	
PixelFormat_BayerGB12Packed	
PixelFormat_BayerBG12Packed	
PixelFormat_YUV411Packed	
PixelFormat_YUV422Packed	
PixelFormat_YUV444Packed	
PixelFormat_Mono12p	
PixelFormat_BayerGR12p	
PixelFormat_BayerRG12p	
PixelFormat_BayerGB12p	
PixelFormat_BayerBG12p	
PixelFormat_YCbCr8	
PixelFormat_YCbCr422_8	
PixelFormat_YCbCr411_8	
PixelFormat_BGR8	
PixelFormat_BGRa8	
PixelFormat_Mono10Packed	
PixelFormat_BayerGR10Packed	
PixelFormat_BayerRG10Packed	
PixelFormat_BayerGB10Packed	
PixelFormat_BayerBG10Packed	
PixelFormat_Mono10p	
PixelFormat_BayerGR10p	
PixelFormat_BayerRG10p	
PixelFormat_BayerGB10p	
PixelFormat_BayerBG10p	
PixelFormat_Mono1p	Monochrome 1-bit packed
PixelFormat_Mono2p	Monochrome 2-bit packed
PixelFormat_Mono4p	Monochrome 4-bit packed
PixelFormat_Mono8s	Monochrome 8-bit signed
PixelFormat_Mono10	Monochrome 10-bit unpacked
PixelFormat_Mono12	Monochrome 12-bit unpacked
PixelFormat_Mono14	Monochrome 14-bit unpacked
PixelFormat_Mono16s	Monochrome 16-bit signed
PixelFormat_Mono32f	Monochrome 32-bit float
PixelFormat_BayerBG10	Bayer Blue-Green 10-bit unpacked
PixelFormat_BayerBG12	Bayer Blue-Green 12-bit unpacked
PixelFormat_BayerGB10	Bayer Green-Blue 10-bit unpacked
PixelFormat_BayerGB12	Bayer Green-Blue 12-bit unpacked
PixelFormat_BayerGR10	Bayer Green-Red 10-bit unpacked
PixelFormat_BayerGR12	Bayer Green-Red 12-bit unpacked
PixelFormat_BayerRG10	Bayer Red-Green 10-bit unpacked
PixelFormat_BayerRG12	Bayer Red-Green 12-bit unpacked

Enumerator

PixelFormat_RGBA8	Red-Green-Blue-alpha 8-bit
PixelFormat_RGBA10	Red-Green-Blue-alpha 10-bit unpacked
PixelFormat_RGBA10p	Red-Green-Blue-alpha 10-bit packed
PixelFormat_RGBA12	Red-Green-Blue-alpha 12-bit unpacked
PixelFormat_RGBA12p	Red-Green-Blue-alpha 12-bit packed
PixelFormat_RGBA14	Red-Green-Blue-alpha 14-bit unpacked
PixelFormat_RGBA16	Red-Green-Blue-alpha 16-bit
PixelFormat_RGB8	Red-Green-Blue 8-bit
PixelFormat_RGB8_Planar	Red-Green-Blue 8-bit planar
PixelFormat_RGB10	Red-Green-Blue 10-bit unpacked
PixelFormat_RGB10_Planar	Red-Green-Blue 10-bit unpacked planar
PixelFormat_RGB10p	Red-Green-Blue 10-bit packed
PixelFormat_RGB10p32	Red-Green-Blue 10-bit packed into 32-bit
PixelFormat_RGB12	Red-Green-Blue 12-bit unpacked
PixelFormat_RGB12_Planar	Red-Green-Blue 12-bit unpacked planar
PixelFormat_RGB12p	Red-Green-Blue 12-bit packed
PixelFormat_RGB14	Red-Green-Blue 14-bit unpacked
PixelFormat_RGB16	Red-Green-Blue 16-bit
PixelFormat_RGB16s	Red-Green-Blue 16-bit signed
PixelFormat_RGB32f	Red-Green-Blue 32-bit float
PixelFormat_RGB16_Planar	Red-Green-Blue 16-bit planar
PixelFormat_RGB565p	Red-Green-Blue 5/6/5-bit packed
PixelFormat_BGRA10	Blue-Green-Red-alpha 10-bit unpacked
PixelFormat_BGRA10p	Blue-Green-Red-alpha 10-bit packed
PixelFormat_BGRA12	Blue-Green-Red-alpha 12-bit unpacked
PixelFormat_BGRA12p	Blue-Green-Red-alpha 12-bit packed
PixelFormat_BGRA14	Blue-Green-Red-alpha 14-bit unpacked
PixelFormat_BGRA16	Blue-Green-Red-alpha 16-bit
PixelFormat_RGBA32f	Red-Green-Blue-alpha 32-bit float
PixelFormat_BGR10	Blue-Green-Red 10-bit unpacked
PixelFormat_BGR10p	Blue-Green-Red 10-bit packed
PixelFormat_BGR12	Blue-Green-Red 12-bit unpacked
PixelFormat_BGR12p	Blue-Green-Red 12-bit packed
PixelFormat_BGR14	Blue-Green-Red 14-bit unpacked
PixelFormat_BGR16	Blue-Green-Red 16-bit
PixelFormat_BGR565p	Blue-Green-Red 5/6/5-bit packed
PixelFormat_R8	Red 8-bit
PixelFormat_R10	Red 10-bit
PixelFormat_R12	Red 12-bit
PixelFormat_R16	Red 16-bit
PixelFormat_G8	Green 8-bit
PixelFormat_G10	Green 10-bit
PixelFormat_G12	Green 12-bit
PixelFormat_G16	Green 16-bit
PixelFormat_B8	Blue 8-bit
PixelFormat_B10	Blue 10-bit

Enumerator

PixelFormat_B12	Blue 12-bit
PixelFormat_B16	Blue 16-bit
PixelFormat_Coord3D_ABC8	3D coordinate A-B-C 8-bit
PixelFormat_Coord3D_ABC8_Planar	3D coordinate A-B-C 8-bit planar
PixelFormat_Coord3D_ABC10p	3D coordinate A-B-C 10-bit packed
PixelFormat_Coord3D_ABC10p_Planar	3D coordinate A-B-C 10-bit packed planar
PixelFormat_Coord3D_ABC12p	3D coordinate A-B-C 12-bit packed
PixelFormat_Coord3D_ABC12p_Planar	3D coordinate A-B-C 12-bit packed planar
PixelFormat_Coord3D_ABC16	3D coordinate A-B-C 16-bit
PixelFormat_Coord3D_ABC16_Planar	3D coordinate A-B-C 16-bit planar
PixelFormat_Coord3D_ABC32f	3D coordinate A-B-C 32-bit floating point
PixelFormat_Coord3D_ABC32f_Planar	3D coordinate A-B-C 32-bit floating point planar
PixelFormat_Coord3D_AC8	3D coordinate A-C 8-bit
PixelFormat_Coord3D_AC8_Planar	3D coordinate A-C 8-bit planar
PixelFormat_Coord3D_AC10p	3D coordinate A-C 10-bit packed
PixelFormat_Coord3D_AC10p_Planar	3D coordinate A-C 10-bit packed planar
PixelFormat_Coord3D_AC12p	3D coordinate A-C 12-bit packed
PixelFormat_Coord3D_AC12p_Planar	3D coordinate A-C 12-bit packed planar
PixelFormat_Coord3D_AC16	3D coordinate A-C 16-bit
PixelFormat_Coord3D_AC16_Planar	3D coordinate A-C 16-bit planar
PixelFormat_Coord3D_AC32f	3D coordinate A-C 32-bit floating point
PixelFormat_Coord3D_AC32f_Planar	3D coordinate A-C 32-bit floating point planar
PixelFormat_Coord3D_A8	3D coordinate A 8-bit
PixelFormat_Coord3D_A10p	3D coordinate A 10-bit packed
PixelFormat_Coord3D_A12p	3D coordinate A 12-bit packed
PixelFormat_Coord3D_A16	3D coordinate A 16-bit
PixelFormat_Coord3D_A32f	3D coordinate A 32-bit floating point
PixelFormat_Coord3D_B8	3D coordinate B 8-bit
PixelFormat_Coord3D_B10p	3D coordinate B 10-bit packed
PixelFormat_Coord3D_B12p	3D coordinate B 12-bit packed
PixelFormat_Coord3D_B16	3D coordinate B 16-bit
PixelFormat_Coord3D_B32f	3D coordinate B 32-bit floating point
PixelFormat_Coord3D_C8	3D coordinate C 8-bit
PixelFormat_Coord3D_C10p	3D coordinate C 10-bit packed
PixelFormat_Coord3D_C12p	3D coordinate C 12-bit packed
PixelFormat_Coord3D_C16	3D coordinate C 16-bit
PixelFormat_Coord3D_C32f	3D coordinate C 32-bit floating point
PixelFormat_Confidence1	Confidence 1-bit unpacked
PixelFormat_Confidence1p	Confidence 1-bit packed
PixelFormat_Confidence8	Confidence 8-bit
PixelFormat_Confidence16	Confidence 16-bit
PixelFormat_Confidence32f	Confidence 32-bit floating point
PixelFormat_BiColorBGRG8	Bi-color Blue/Green - Red/Green 8-bit
PixelFormat_BiColorBGRG10	Bi-color Blue/Green - Red/Green 10-bit unpacked
PixelFormat_BiColorBGRG10p	Bi-color Blue/Green - Red/Green 10-bit packed
PixelFormat_BiColorBGRG12	Bi-color Blue/Green - Red/Green 12-bit unpacked
PixelFormat_BiColorBGRG12p	Bi-color Blue/Green - Red/Green 12-bit packed

Enumerator

PixelFormat_BiColorRGBG8	Bi-color Red/Green - Blue/Green 8-bit
PixelFormat_BiColorRGBG10	Bi-color Red/Green - Blue/Green 10-bit unpacked
PixelFormat_BiColorRGBG10p	Bi-color Red/Green - Blue/Green 10-bit packed
PixelFormat_BiColorRGBG12	Bi-color Red/Green - Blue/Green 12-bit unpacked
PixelFormat_BiColorRGBG12p	Bi-color Red/Green - Blue/Green 12-bit packed
PixelFormat_SCF1WBWG8	Sparse Color Filter #1 White-Blue-White-Green 8-bit
PixelFormat_SCF1WBWG10	Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked
PixelFormat_SCF1WBWG10p	Sparse Color Filter #1 White-Blue-White-Green 10-bit packed
PixelFormat_SCF1WBWG12	Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked
PixelFormat_SCF1WBWG12p	Sparse Color Filter #1 White-Blue-White-Green 12-bit packed
PixelFormat_SCF1WBWG14	Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked
PixelFormat_SCF1WBWG16	Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked
PixelFormat_SCF1WGWB8	Sparse Color Filter #1 White-Green-White-Blue 8-bit
PixelFormat_SCF1WGWB10	Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked
PixelFormat_SCF1WGWB10p	Sparse Color Filter #1 White-Green-White-Blue 10-bit packed
PixelFormat_SCF1WGWB12	Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked
PixelFormat_SCF1WGWB12p	Sparse Color Filter #1 White-Green-White-Blue 12-bit packed
PixelFormat_SCF1WGWB14	Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked
PixelFormat_SCF1WGWB16	Sparse Color Filter #1 White-Green-White-Blue 16-bit
PixelFormat_SCF1WGWR8	Sparse Color Filter #1 White-Green-White-Red 8-bit
PixelFormat_SCF1WGWR10	Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked
PixelFormat_SCF1WGWR10p	Sparse Color Filter #1 White-Green-White-Red 10-bit packed
PixelFormat_SCF1WGWR12	Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked
PixelFormat_SCF1WGWR12p	Sparse Color Filter #1 White-Green-White-Red 12-bit packed
PixelFormat_SCF1WGWR14	Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked
PixelFormat_SCF1WGWR16	Sparse Color Filter #1 White-Green-White-Red 16-bit
PixelFormat_SCF1WRWG8	Sparse Color Filter #1 White-Red-White-Green 8-bit
PixelFormat_SCF1WRWG10	Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked
PixelFormat_SCF1WRWG10p	Sparse Color Filter #1 White-Red-White-Green 10-bit packed
PixelFormat_SCF1WRWG12	Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked
PixelFormat_SCF1WRWG12p	Sparse Color Filter #1 White-Red-White-Green 12-bit packed
PixelFormat_SCF1WRWG14	Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked
PixelFormat_SCF1WRWG16	Sparse Color Filter #1 White-Red-White-Green 16-bit
PixelFormat_YCbCr8_CbYCr	YCbCr 4:4:4 8-bit
PixelFormat_YCbCr10_CbYCr	YCbCr 4:4:4 10-bit unpacked
PixelFormat_YCbCr10p_CbYCr	YCbCr 4:4:4 10-bit packed
PixelFormat_YCbCr12_CbYCr	YCbCr 4:4:4 12-bit unpacked
PixelFormat_YCbCr12p_CbYCr	YCbCr 4:4:4 12-bit packed
PixelFormat_YCbCr411_8_CbYYCrYY	YCbCr 4:1:1 8-bit
PixelFormat_YCbCr422_8_CbYCrY	YCbCr 4:2:2 8-bit
PixelFormat_YCbCr422_10	YCbCr 4:2:2 10-bit unpacked
PixelFormat_YCbCr422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked
PixelFormat_YCbCr422_10p	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed
PixelFormat_YCbCr422_12	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked
PixelFormat_YCbCr422_12p	YCbCr 4:2:2 12-bit packed

Enumerator

PixelFormat_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormat_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormat_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormat_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormat_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormat_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormat_YCbCr601_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormat_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormat_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601
PixelFormat_YCbCr601_422_10	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormat_YCbCr601_422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormat_YCbCr601_422_10p	YCbCr 4:2:2 10-bit packed BT.601
PixelFormat_YCbCr601_422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed BT.601
PixelFormat_YCbCr601_422_12	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormat_YCbCr601_422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormat_YCbCr601_422_12p	YCbCr 4:2:2 12-bit packed BT.601
PixelFormat_YCbCr601_422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed BT.601
PixelFormat_YCbCr709_8_CbYCr	YCbCr 4:4:4 8-bit BT.709
PixelFormat_YCbCr709_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.709
PixelFormat_YCbCr709_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.709
PixelFormat_YCbCr709_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.709
PixelFormat_YCbCr709_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.709
PixelFormat_YCbCr709_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.709
PixelFormat_YCbCr709_422_8	YCbCr 4:2:2 8-bit BT.709
PixelFormat_YCbCr709_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.709
PixelFormat_YCbCr709_422_10	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormat_YCbCr709_422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormat_YCbCr709_422_10p	YCbCr 4:2:2 10-bit packed BT.709
PixelFormat_YCbCr709_422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed BT.709
PixelFormat_YCbCr709_422_12	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormat_YCbCr709_422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormat_YCbCr709_422_12p	YCbCr 4:2:2 12-bit packed BT.709
PixelFormat_YCbCr709_422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed BT.709
PixelFormat_YUV8_UYV	YUV 4:4:4 8-bit
PixelFormat_YUV411_8_UYYVYY	YUV 4:1:1 8-bit
PixelFormat_YUV422_8	YUV 4:2:2 8-bit
PixelFormat_YUV422_8_UYVY	YUV 4:2:2 8-bit
PixelFormat_Polarized8	Monochrome Polarized 8-bit
PixelFormat_Polarized10p	Monochrome Polarized 10-bit packed
PixelFormat_Polarized12p	Monochrome Polarized 12-bit packed
PixelFormat_Polarized16	Monochrome Polarized 16-bit
PixelFormat_BayerRGPolarized8	Polarized Bayer Red Green filter 8-bit
PixelFormat_BayerRGPolarized10p	Polarized Bayer Red Green filter 10-bit packed
PixelFormat_BayerRGPolarized12p	Polarized Bayer Red Green filter 12-bit packed
PixelFormat_BayerRGPolarized16	Polarized Bayer Red Green filter 16-bit
PixelFormat_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormat_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit

Enumerator

PixelFormat_JPEGMono8	JPEG Monochrome 8-bit
PixelFormat_JPEGColor8	JPEG Color 8-bit
PixelFormat_Raw16	Raw 16 bit.
PixelFormat_Raw8	Raw bit.
PixelFormat_R12_Jpeg	Red 12-bit JPEG.
PixelFormat_GR12_Jpeg	Green Red 12-bit JPEG.
PixelFormat_GB12_Jpeg	Green Blue 12-bit JPEG.
PixelFormat_B12_Jpeg	Blue 12-bit packed JPEG.
PixelFormat_GR12	Green-Red (single) channel from Bayer pattern 12-bit.
PixelFormat_GB12	Green-Blue (single) channel from Bayer pattern 12-bit.
UNKNOWN_PIXELFORMAT	
NUM_PIXELFORMAT	

14.5.1.153 PixelFormatInfoSelectorEnums

```
enum PixelFormatInfoSelectorEnums
```

< Select the pixel format for which the information will be returned.

Enumerator

PixelFormatInfoSelector_Mono1p	Monochrome 1-bit packed
PixelFormatInfoSelector_Mono2p	Monochrome 2-bit packed
PixelFormatInfoSelector_Mono4p	Monochrome 4-bit packed
PixelFormatInfoSelector_Mono8	Monochrome 8-bit
PixelFormatInfoSelector_Mono8s	Monochrome 8-bit signed
PixelFormatInfoSelector_Mono10	Monochrome 10-bit unpacked
PixelFormatInfoSelector_Mono10p	Monochrome 10-bit packed
PixelFormatInfoSelector_Mono12	Monochrome 12-bit unpacked
PixelFormatInfoSelector_Mono12p	Monochrome 12-bit packed
PixelFormatInfoSelector_Mono14	Monochrome 14-bit unpacked
PixelFormatInfoSelector_Mono16	Monochrome 16-bit
PixelFormatInfoSelector_Mono16s	Monochrome 16-bit signed
PixelFormatInfoSelector_Mono32f	Monochrome 32-bit float
PixelFormatInfoSelector_BayerBG8	Bayer Blue-Green 8-bit
PixelFormatInfoSelector_BayerBG10	Bayer Blue-Green 10-bit unpacked
PixelFormatInfoSelector_BayerBG10p	Bayer Blue-Green 10-bit packed
PixelFormatInfoSelector_BayerBG12	Bayer Blue-Green 12-bit unpacked
PixelFormatInfoSelector_BayerBG12p	Bayer Blue-Green 12-bit packed
PixelFormatInfoSelector_BayerBG16	Bayer Blue-Green 16-bit
PixelFormatInfoSelector_BayerGB8	Bayer Green-Blue 8-bit
PixelFormatInfoSelector_BayerGB10	Bayer Green-Blue 10-bit unpacked
PixelFormatInfoSelector_BayerGB10p	Bayer Green-Blue 10-bit packed
PixelFormatInfoSelector_BayerGB12	Bayer Green-Blue 12-bit unpacked
PixelFormatInfoSelector_BayerGB12p	Bayer Green-Blue 12-bit packed

Enumerator

PixelFormatInfoSelector_BayerGB16	Bayer Green-Blue 16-bit
PixelFormatInfoSelector_BayerGR8	Bayer Green-Red 8-bit
PixelFormatInfoSelector_BayerGR10	Bayer Green-Red 10-bit unpacked
PixelFormatInfoSelector_BayerGR10p	Bayer Green-Red 10-bit packed
PixelFormatInfoSelector_BayerGR12	Bayer Green-Red 12-bit unpacked
PixelFormatInfoSelector_BayerGR12p	Bayer Green-Red 12-bit packed
PixelFormatInfoSelector_BayerGR16	Bayer Green-Red 16-bit
PixelFormatInfoSelector_BayerRG8	Bayer Red-Green 8-bit
PixelFormatInfoSelector_BayerRG10	Bayer Red-Green 10-bit unpacked
PixelFormatInfoSelector_BayerRG10p	Bayer Red-Green 10-bit packed
PixelFormatInfoSelector_BayerRG12	Bayer Red-Green 12-bit unpacked
PixelFormatInfoSelector_BayerRG12p	Bayer Red-Green 12-bit packed
PixelFormatInfoSelector_BayerRG16	Bayer Red-Green 16-bit
PixelFormatInfoSelector_RGBa8	Red-Green-Blue-alpha 8-bit
PixelFormatInfoSelector_RGBa10	Red-Green-Blue-alpha 10-bit unpacked
PixelFormatInfoSelector_RGBa10p	Red-Green-Blue-alpha 10-bit packed
PixelFormatInfoSelector_RGBa12	Red-Green-Blue-alpha 12-bit unpacked
PixelFormatInfoSelector_RGBa12p	Red-Green-Blue-alpha 12-bit packed
PixelFormatInfoSelector_RGBa14	Red-Green-Blue-alpha 14-bit unpacked
PixelFormatInfoSelector_RGBa16	Red-Green-Blue-alpha 16-bit
PixelFormatInfoSelector_RGB8	Red-Green-Blue 8-bit
PixelFormatInfoSelector_RGB8_Planar	Red-Green-Blue 8-bit planar
PixelFormatInfoSelector_RGB10	Red-Green-Blue 10-bit unpacked
PixelFormatInfoSelector_RGB10_Planar	Red-Green-Blue 10-bit unpacked planar
PixelFormatInfoSelector_RGB10p	Red-Green-Blue 10-bit packed
PixelFormatInfoSelector_RGB10p32	Red-Green-Blue 10-bit packed into 32-bit
PixelFormatInfoSelector_RGB12	Red-Green-Blue 12-bit unpacked
PixelFormatInfoSelector_RGB12_Planar	Red-Green-Blue 12-bit unpacked planar
PixelFormatInfoSelector_RGB12p	Red-Green-Blue 12-bit packed
PixelFormatInfoSelector_RGB14	Red-Green-Blue 14-bit unpacked
PixelFormatInfoSelector_RGB16	Red-Green-Blue 16-bit
PixelFormatInfoSelector_RGB16s	Red-Green-Blue 16-bit signed
PixelFormatInfoSelector_RGB32f	Red-Green-Blue 32-bit float
PixelFormatInfoSelector_RGB16_Planar	Red-Green-Blue 16-bit planar
PixelFormatInfoSelector_RGB565p	Red-Green-Blue 5/6/5-bit packed
PixelFormatInfoSelector_BGRa8	Blue-Green-Red-alpha 8-bit
PixelFormatInfoSelector_BGRa10	Blue-Green-Red-alpha 10-bit unpacked
PixelFormatInfoSelector_BGRa10p	Blue-Green-Red-alpha 10-bit packed
PixelFormatInfoSelector_BGRa12	Blue-Green-Red-alpha 12-bit unpacked
PixelFormatInfoSelector_BGRa12p	Blue-Green-Red-alpha 12-bit packed
PixelFormatInfoSelector_BGRa14	Blue-Green-Red-alpha 14-bit unpacked
PixelFormatInfoSelector_BGRa16	Blue-Green-Red-alpha 16-bit
PixelFormatInfoSelector_RGBa32f	Red-Green-Blue-alpha 32-bit float
PixelFormatInfoSelector_BGR8	Blue-Green-Red 8-bit
PixelFormatInfoSelector_BGR10	Blue-Green-Red 10-bit unpacked

Enumerator

PixelFormatInfoSelector_BGR10p	Blue-Green-Red 10-bit packed
PixelFormatInfoSelector_BGR12	Blue-Green-Red 12-bit unpacked
PixelFormatInfoSelector_BGR12p	Blue-Green-Red 12-bit packed
PixelFormatInfoSelector_BGR14	Blue-Green-Red 14-bit unpacked
PixelFormatInfoSelector_BGR16	Blue-Green-Red 16-bit
PixelFormatInfoSelector_BGR565p	Blue-Green-Red 5/6/5-bit packed
PixelFormatInfoSelector_R8	Red 8-bit
PixelFormatInfoSelector_R10	Red 10-bit
PixelFormatInfoSelector_R12	Red 12-bit
PixelFormatInfoSelector_R16	Red 16-bit
PixelFormatInfoSelector_G8	Green 8-bit
PixelFormatInfoSelector_G10	Green 10-bit
PixelFormatInfoSelector_G12	Green 12-bit
PixelFormatInfoSelector_G16	Green 16-bit
PixelFormatInfoSelector_B8	Blue 8-bit
PixelFormatInfoSelector_B10	Blue 10-bit
PixelFormatInfoSelector_B12	Blue 12-bit
PixelFormatInfoSelector_B16	Blue 16-bit
PixelFormatInfoSelector_Coord3D_ABC8	3D coordinate A-B-C 8-bit
PixelFormatInfoSelector_Coord3D_ABC8_Planar	3D coordinate A-B-C 8-bit planar
PixelFormatInfoSelector_Coord3D_ABC10p	3D coordinate A-B-C 10-bit packed
PixelFormatInfoSelector_Coord3D_ABC10p_Planar	3D coordinate A-B-C 10-bit packed planar
PixelFormatInfoSelector_Coord3D_ABC12p	3D coordinate A-B-C 12-bit packed
PixelFormatInfoSelector_Coord3D_ABC12p_Planar	3D coordinate A-B-C 12-bit packed planar
PixelFormatInfoSelector_Coord3D_ABC16	3D coordinate A-B-C 16-bit
PixelFormatInfoSelector_Coord3D_ABC16_Planar	3D coordinate A-B-C 16-bit planar
PixelFormatInfoSelector_Coord3D_ABC32f	3D coordinate A-B-C 32-bit floating point
PixelFormatInfoSelector_Coord3D_ABC32f_Planar	3D coordinate A-B-C 32-bit floating point planar
PixelFormatInfoSelector_Coord3D_AC8	3D coordinate A-C 8-bit
PixelFormatInfoSelector_Coord3D_AC8_Planar	3D coordinate A-C 8-bit planar
PixelFormatInfoSelector_Coord3D_AC10p	3D coordinate A-C 10-bit packed
PixelFormatInfoSelector_Coord3D_AC10p_Planar	3D coordinate A-C 10-bit packed planar
PixelFormatInfoSelector_Coord3D_AC12p	3D coordinate A-C 12-bit packed
PixelFormatInfoSelector_Coord3D_AC12p_Planar	3D coordinate A-C 12-bit packed planar
PixelFormatInfoSelector_Coord3D_AC16	3D coordinate A-C 16-bit
PixelFormatInfoSelector_Coord3D_AC16_Planar	3D coordinate A-C 16-bit planar
PixelFormatInfoSelector_Coord3D_AC32f	3D coordinate A-C 32-bit floating point
PixelFormatInfoSelector_Coord3D_AC32f_Planar	3D coordinate A-C 32-bit floating point planar
PixelFormatInfoSelector_Coord3D_A8	3D coordinate A 8-bit
PixelFormatInfoSelector_Coord3D_A10p	3D coordinate A 10-bit packed
PixelFormatInfoSelector_Coord3D_A12p	3D coordinate A 12-bit packed
PixelFormatInfoSelector_Coord3D_A16	3D coordinate A 16-bit
PixelFormatInfoSelector_Coord3D_A32f	3D coordinate A 32-bit floating point
PixelFormatInfoSelector_Coord3D_B8	3D coordinate B 8-bit
PixelFormatInfoSelector_Coord3D_B10p	3D coordinate B 10-bit packed
PixelFormatInfoSelector_Coord3D_B12p	3D coordinate B 12-bit packed
PixelFormatInfoSelector_Coord3D_B16	3D coordinate B 16-bit
PixelFormatInfoSelector_Coord3D_B32f	3D coordinate B 32-bit floating point

Enumerator

PixelFormatInfoSelector_Coord3D_C8	3D coordinate C 8-bit
PixelFormatInfoSelector_Coord3D_C10p	3D coordinate C 10-bit packed
PixelFormatInfoSelector_Coord3D_C12p	3D coordinate C 12-bit packed
PixelFormatInfoSelector_Coord3D_C16	3D coordinate C 16-bit
PixelFormatInfoSelector_Coord3D_C32f	3D coordinate C 32-bit floating point
PixelFormatInfoSelector_Confidence1	Confidence 1-bit unpacked
PixelFormatInfoSelector_Confidence1p	Confidence 1-bit packed
PixelFormatInfoSelector_Confidence8	Confidence 8-bit
PixelFormatInfoSelector_Confidence16	Confidence 16-bit
PixelFormatInfoSelector_Confidence32f	Confidence 32-bit floating point
PixelFormatInfoSelector_BiColorBGRG8	Bi-color Blue/Green - Red/Green 8-bit
PixelFormatInfoSelector_BiColorBGRG10	Bi-color Blue/Green - Red/Green 10-bit unpacked
PixelFormatInfoSelector_BiColorBGRG10p	Bi-color Blue/Green - Red/Green 10-bit packed
PixelFormatInfoSelector_BiColorBGRG12	Bi-color Blue/Green - Red/Green 12-bit unpacked
PixelFormatInfoSelector_BiColorBGRG12p	Bi-color Blue/Green - Red/Green 12-bit packed
PixelFormatInfoSelector_BiColorRGBG8	Bi-color Red/Green - Blue/Green 8-bit
PixelFormatInfoSelector_BiColorRGBG10	Bi-color Red/Green - Blue/Green 10-bit unpacked
PixelFormatInfoSelector_BiColorRGBG10p	Bi-color Red/Green - Blue/Green 10-bit packed
PixelFormatInfoSelector_BiColorRGBG12	Bi-color Red/Green - Blue/Green 12-bit unpacked
PixelFormatInfoSelector_BiColorRGBG12p	Bi-color Red/Green - Blue/Green 12-bit packed
PixelFormatInfoSelector_SCF1WBWG8	Sparse Color Filter #1 White-Blue-White-Green 8-bit
PixelFormatInfoSelector_SCF1WBWG10	Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked
PixelFormatInfoSelector_SCF1WBWG10p	Sparse Color Filter #1 White-Blue-White-Green 10-bit packed
PixelFormatInfoSelector_SCF1WBWG12	Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked
PixelFormatInfoSelector_SCF1WBWG12p	Sparse Color Filter #1 White-Blue-White-Green 12-bit packed
PixelFormatInfoSelector_SCF1WBWG14	Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked
PixelFormatInfoSelector_SCF1WBWG16	Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked
PixelFormatInfoSelector_SCF1WGWB8	Sparse Color Filter #1 White-Green-White-Blue 8-bit
PixelFormatInfoSelector_SCF1WGWB10	Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked
PixelFormatInfoSelector_SCF1WGWB10p	Sparse Color Filter #1 White-Green-White-Blue 10-bit packed
PixelFormatInfoSelector_SCF1WGWB12	Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked
PixelFormatInfoSelector_SCF1WGWB12p	Sparse Color Filter #1 White-Green-White-Blue 12-bit packed
PixelFormatInfoSelector_SCF1WGWB14	Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked
PixelFormatInfoSelector_SCF1WGWB16	Sparse Color Filter #1 White-Green-White-Blue 16-bit
PixelFormatInfoSelector_SCF1WGWR8	Sparse Color Filter #1 White-Green-White-Red 8-bit
PixelFormatInfoSelector_SCF1WGWR10	Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked

Enumerator

PixelFormatInfoSelector_SCF1WGWR10p	Sparse Color Filter #1 White-Green-White-Red 10-bit packed
PixelFormatInfoSelector_SCF1WGWR12	Sparse Color Filter #1 White-Green-White-Red 12-bit unpacked
PixelFormatInfoSelector_SCF1WGWR12p	Sparse Color Filter #1 White-Green-White-Red 12-bit packed
PixelFormatInfoSelector_SCF1WGWR14	Sparse Color Filter #1 White-Green-White-Red 14-bit unpacked
PixelFormatInfoSelector_SCF1WGWR16	Sparse Color Filter #1 White-Green-White-Red 16-bit
PixelFormatInfoSelector_SCF1WRWG8	Sparse Color Filter #1 White-Red-White-Green 8-bit
PixelFormatInfoSelector_SCF1WRWG10	Sparse Color Filter #1 White-Red-White-Green 10-bit unpacked
PixelFormatInfoSelector_SCF1WRWG10p	Sparse Color Filter #1 White-Red-White-Green 10-bit packed
PixelFormatInfoSelector_SCF1WRWG12	Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked
PixelFormatInfoSelector_SCF1WRWG12p	Sparse Color Filter #1 White-Red-White-Green 12-bit packed
PixelFormatInfoSelector_SCF1WRWG14	Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked
PixelFormatInfoSelector_SCF1WRWG16	Sparse Color Filter #1 White-Red-White-Green 16-bit
PixelFormatInfoSelector_YCbCr8	YCbCr 4:4:4 8-bit
PixelFormatInfoSelector_YCbCr8_CbYCr	YCbCr 4:4:4 8-bit
PixelFormatInfoSelector_YCbCr10_CbYCr	YCbCr 4:4:4 10-bit unpacked
PixelFormatInfoSelector_YCbCr10p_CbYCr	YCbCr 4:4:4 10-bit packed
PixelFormatInfoSelector_YCbCr12_CbYCr	YCbCr 4:4:4 12-bit unpacked
PixelFormatInfoSelector_YCbCr12p_CbYCr	YCbCr 4:4:4 12-bit packed
PixelFormatInfoSelector_YCbCr411_8	YCbCr 4:1:1 8-bit
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY	YCbCr 4:1:1 8-bit
PixelFormatInfoSelector_YCbCr422_8	YCbCr 4:2:2 8-bit
PixelFormatInfoSelector_YCbCr422_8_CbYCrY	YCbCr 4:2:2 8-bit
PixelFormatInfoSelector_YCbCr422_10	YCbCr 4:2:2 10-bit unpacked
PixelFormatInfoSelector_YCbCr422_10_CbYCrY	YCbCr 4:2:2 10-bit unpacked
PixelFormatInfoSelector_YCbCr422_10p	YCbCr 4:2:2 10-bit packed
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY	YCbCr 4:2:2 10-bit packed
PixelFormatInfoSelector_YCbCr422_12	YCbCr 4:2:2 12-bit unpacked
PixelFormatInfoSelector_YCbCr422_12_CbYCrY	YCbCr 4:2:2 12-bit unpacked
PixelFormatInfoSelector_YCbCr422_12p	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY	YCbCr 4:2:2 12-bit packed
PixelFormatInfoSelector_YCbCr601_8_CbYCr	YCbCr 4:4:4 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY	YCbCr 4:1:1 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.601
PixelFormatInfoSelector_YCbCr601_422_10	YCbCr 4:2:2 10-bit unpacked BT.601

Enumerator

PixelFormatInfoSelector_YCbCr601_422_10_CbY↔CrY	YCbCr 4:2:2 10-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_10p	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_10p_Cb↔YCrY	YCbCr 4:2:2 10-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12_CbY↔CrY	YCbCr 4:2:2 12-bit unpacked BT.601
PixelFormatInfoSelector_YCbCr601_422_12p	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr601_422_12p_Cb↔YCrY	YCbCr 4:2:2 12-bit packed BT.601
PixelFormatInfoSelector_YCbCr709_8_CbYCr	YCbCr 4:4:4 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_10_CbYCr	YCbCr 4:4:4 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_10p_CbYCr	YCbCr 4:4:4 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_12_CbYCr	YCbCr 4:4:4 12-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_12p_CbYCr	YCbCr 4:4:4 12-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_411_8_CbYY↔CrYY	YCbCr 4:1:1 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_8	YCbCr 4:2:2 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY	YCbCr 4:2:2 8-bit BT.709
PixelFormatInfoSelector_YCbCr709_422_10	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_10_CbY↔CrY	YCbCr 4:2:2 10-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_10p	YCbCr 4:2:2 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_10p_Cb↔YCrY	YCbCr 4:2:2 10-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_12	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_12_CbY↔CrY	YCbCr 4:2:2 12-bit unpacked BT.709
PixelFormatInfoSelector_YCbCr709_422_12p	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YCbCr709_422_12p_Cb↔YCrY	YCbCr 4:2:2 12-bit packed BT.709
PixelFormatInfoSelector_YUV8_UYV	YUV 4:4:4 8-bit
PixelFormatInfoSelector_YUV411_8_UYYVYY	YUV 4:1:1 8-bit
PixelFormatInfoSelector_YUV422_8	YUV 4:2:2 8-bit
PixelFormatInfoSelector_YUV422_8_UYVY	YUV 4:2:2 8-bit
PixelFormatInfoSelector_Polarized8	Monochrome Polarized 8-bit
PixelFormatInfoSelector_Polarized10p	Monochrome Polarized 10-bit packed
PixelFormatInfoSelector_Polarized12p	Monochrome Polarized 12-bit packed
PixelFormatInfoSelector_Polarized16	Monochrome Polarized 16-bit
PixelFormatInfoSelector_BayerRGPolarized8	Polarized Bayer Red Green filter 8-bit
PixelFormatInfoSelector_BayerRGPolarized10p	Polarized Bayer Red Green filter 10-bit packed
PixelFormatInfoSelector_BayerRGPolarized12p	Polarized Bayer Red Green filter 12-bit packed
PixelFormatInfoSelector_BayerRGPolarized16	Polarized Bayer Red Green filter 16-bit
PixelFormatInfoSelector_LLCMono8	Lossless Compression Monochrome 8-bit
PixelFormatInfoSelector_LLCBayerRG8	Lossless Compression Bayer Red Green filter 8-bit
PixelFormatInfoSelector_JPEGMono8	JPEG Monochrome 8-bit
PixelFormatInfoSelector_JPEGColor8	JPEG Color 8-bit
NUM_PIXELFORMATINFOSELECTOR	

14.5.1.154 PixelFormatIntType

enum `PixelFormatIntType`

Possible integer types and packing used in a pixel format.

Enumerator

SPINNAKER_INT_TYPE_UINT8	
SPINNAKER_INT_TYPE_INT8	
SPINNAKER_INT_TYPE_UINT10	
SPINNAKER_INT_TYPE_UINT10p	
SPINNAKER_INT_TYPE_UINT10P	
SPINNAKER_INT_TYPE_UINT12	
SPINNAKER_INT_TYPE_UINT12p	
SPINNAKER_INT_TYPE_UINT12P	
SPINNAKER_INT_TYPE_UINT14	
SPINNAKER_INT_TYPE_UINT16	
SPINNAKER_INT_TYPE_INT16	
SPINNAKER_INT_TYPE_FLOAT32	
SPINNAKER_INT_TYPE_UNKNOWN	

14.5.1.155 PixelSizeEnums

enum `PixelSizeEnums`

< Total size in bits of a pixel of the image.

Enumerator

PixelSize_Bpp1	1 bit per pixel.
PixelSize_Bpp2	2 bits per pixel.
PixelSize_Bpp4	4 bits per pixel.
PixelSize_Bpp8	8 bits per pixel.
PixelSize_Bpp10	10 bits per pixel.
PixelSize_Bpp12	12 bits per pixel.
PixelSize_Bpp14	14 bits per pixel.
PixelSize_Bpp16	16 bits per pixel.
PixelSize_Bpp20	20 bits per pixel.
PixelSize_Bpp24	24 bits per pixel.
PixelSize_Bpp30	30 bits per pixel.
PixelSize_Bpp32	32 bits per pixel.
PixelSize_Bpp36	36 bits per pixel.
PixelSize_Bpp48	48 bits per pixel.
PixelSize_Bpp64	64 bits per pixel.
PixelSize_Bpp96	96 bits per pixel.
NUM_PIXELSIZE	

14.5.1.156 POEStatusEnum

enum [POEStatusEnum](#)

< Reports and controls the interface's power over Ethernet status.

Enumerator

POEStatus_NotSupported	Not Supported
POEStatus_PowerOff	Power is Off
POEStatus_PowerOn	Power is On
NUMPOESTATUS	

14.5.1.157 PolarizationQuadrant

enum [PolarizationQuadrant](#)

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

Enumerator

SPINNAKER_POLARIZATION_QUADRANT_I0	The 0 degree of polarization.
SPINNAKER_POLARIZATION_QUADRANT_I45	The 45 degree of polarization.
SPINNAKER_POLARIZATION_QUADRANT_I90	The 90 degree of polarization.
SPINNAKER_POLARIZATION_QUADRANT_I135	The 135 degree of polarization.

14.5.1.158 RegionDestinationEnums

enum [RegionDestinationEnums](#)

< Control the destination of the selected region.

Enumerator

RegionDestination_Stream0	The destination of the region is the data stream 0.
RegionDestination_Stream1	The destination of the region is the data stream 1.
RegionDestination_Stream2	The destination of the region is the data stream 2.
NUM_REGIONDESTINATION	

14.5.1.159 RegionModeEnums

enum [RegionModeEnums](#)

< Controls if the selected Region of interest is active and streaming.

Enumerator

RegionMode_Off	Disable the usage of the Region.
RegionMode_On	Enable the usage of the Region.
NUM_REGIONMODE	

14.5.1.160 RegionSelectorEnums

enum [RegionSelectorEnums](#)

< Selects the Region of interest to control. The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently.

Enumerator

RegionSelector_Region0	Selected feature will control the region 0.
RegionSelector_Region1	Selected feature will control the region 1.
RegionSelector_Region2	Selected feature will control the region 2.
RegionSelector_All	Selected features will control all the regions at the same time.
NUM_REGIONSELECTOR	

14.5.1.161 RgbTransformLightSourceEnums

enum [RgbTransformLightSourceEnums](#)

< Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources. Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

Enumerator

RgbTransformLightSource_General	Uses a matrix calibrated for a wide range of light sources.
RgbTransformLightSource_Tungsten2800K	Uses a matrix optimized for tungsten/incandescent light with color temperature 2800K.
RgbTransformLightSource_WarmFluorescent3000K	Uses a matrix optimized for a typical warm fluorescecent light with color temperature 3000K.
RgbTransformLightSource_CoolFluorescent4000K	Uses a matrix optimized for a typical cool fluorescecent light with color temperature 4000K.

Enumerator

RgbTransformLightSource_Daylight5000K	Uses a matrix optimized for noon Daylight with color temperature 5000K.
RgbTransformLightSource_Cloudy6500K	Uses a matrix optimized for a cloudy sky with color temperature 6500K.
RgbTransformLightSource_Shade8000K	Uses a matrix optimized for shade with color temperature 8000K.
RgbTransformLightSource_Custom	Uses a custom matrix set by the user through the ColorTransformationValueSelector and ColorTransformationValue controls.
NUM_RGBTRANSFORMLIGHTSOURCE	

14.5.1.162 Scan3dCoordinateReferenceSelectorEnums

enum [Scan3dCoordinateReferenceSelectorEnums](#)

< Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

Enumerator

Scan3dCoordinateReferenceSelector_RotationX	Rotation around X axis.
Scan3dCoordinateReferenceSelector_RotationY	Rotation around Y axis.
Scan3dCoordinateReferenceSelector_RotationZ	Rotation around Z axis.
Scan3dCoordinateReferenceSelector_TranslationX	X axis translation.
Scan3dCoordinateReferenceSelector_TranslationY	Y axis translation.
Scan3dCoordinateReferenceSelector_TranslationZ	Z axis translation.
NUM_SCAN3DCOORDINATEREFERENCESELECTOR	

14.5.1.163 Scan3dCoordinateSelectorEnums

enum [Scan3dCoordinateSelectorEnums](#)

< Selects the individual coordinates in the vectors for 3D information/transformation.

Enumerator

Scan3dCoordinateSelector_CoordinateA	The first (X or Theta) coordinate
Scan3dCoordinateSelector_CoordinateB	The second (Y or Phi) coordinate
Scan3dCoordinateSelector_CoordinateC	The third (Z or Rho) coordinate.
NUM_SCAN3DCOORDINATESELECTOR	

14.5.1.164 Scan3dCoordinateSystemEnums

enum [Scan3dCoordinateSystemEnums](#)

< Specifies the Coordinate system to use for the device.

Enumerator

Scan3dCoordinateSystem_Cartesian	Default value. 3-axis orthogonal, right-hand X-Y-Z.
Scan3dCoordinateSystem_Spherical	A Theta-Phi-Rho coordinate system.
Scan3dCoordinateSystem_Cylindrical	A Theta-Y-Rho coordinate system.
NUM_SCAN3DCOORDINATESYSTEM	

14.5.1.165 Scan3dCoordinateSystemReferenceEnums

enum [Scan3dCoordinateSystemReferenceEnums](#)

< Defines coordinate system reference location.

Enumerator

Scan3dCoordinateSystemReference_Anchor	Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used.
Scan3dCoordinateSystemReference_Transformed	Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices.
NUM_SCAN3DCOORDINATESYSTEMREFERENCE	

14.5.1.166 Scan3dCoordinateTransformSelectorEnums

enum [Scan3dCoordinateTransformSelectorEnums](#)

< Sets the index to read/write a coordinate transform value.

Enumerator

Scan3dCoordinateTransformSelector_RotationX	Rotation around X axis.
Scan3dCoordinateTransformSelector_RotationY	Rotation around Y axis.
Scan3dCoordinateTransformSelector_RotationZ	Rotation around Z axis.
Scan3dCoordinateTransformSelector_TranslationX	Translation along X axis.
Scan3dCoordinateTransformSelector_TranslationY	Translation along Y axis.
Scan3dCoordinateTransformSelector_TranslationZ	Translation along Z axis.
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR	

14.5.1.167 Scan3dDistanceUnitEnums

enum [Scan3dDistanceUnitEnums](#)

< Specifies the unit used when delivering calibrated distance data.

Enumerator

Scan3dDistanceUnit_Millimeter	Distance values are in millimeter units (default).
Scan3dDistanceUnit_Inch	Distance values are in inch units.
NUM_SCAN3DDISTANCEUNIT	

14.5.1.168 Scan3dOutputModeEnums

enum [Scan3dOutputModeEnums](#)

< Controls the Calibration and data organization of the device, naming the coordinates transmitted.

Enumerator

Scan3dOutputMode_UncalibratedC	Uncalibrated 2.5D Depth map. The distance data does not represent a physical unit and may be non-linear. The data is a 2.5D range map only.
Scan3dOutputMode_CalibratedABC_Grid	3 Coordinates in grid organization. The full 3 coordinate data with the grid array organization from the sensor kept.
Scan3dOutputMode_CalibratedABC_PointCloud	3 Coordinates without organization. The full 3 coordinate data without any organization of data points. Typically only valid points transmitted giving varying image size.
Scan3dOutputMode_CalibratedAC	2 Coordinates with fixed B sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis uses the scale and offset parameters for the B axis.
Scan3dOutputMode_CalibratedAC_Linescan	2 Coordinates with varying sampling. The data is sent as a A and C coordinates (X,Z or Theta,Rho). The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_CalibratedC	Calibrated 2.5D Depth map. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. No information on X-Y axes available.
Scan3dOutputMode_CalibratedC_Linescan	Depth Map with varying B sampling. The distance data is expressed in the chosen distance unit. The data is a 2.5D range map. The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_RectifiedC	Rectified 2.5D Depth map. The distance data has been rectified to a uniform sampling pattern in the X and Y direction. The data is a 2.5D range map only. If a complete 3D point cloud is rectified but transmitted as explicit coordinates it should be transmitted as one of the "CalibratedABC" formats.

Enumerator

Scan3dOutputMode_RectifiedC_Linescan	Rectified 2.5D Depth map with varying B sampling. The data is sent as rectified 1D profiles using Coord3D_C pixels. The B (Y) axis comes from the encoder chunk value.
Scan3dOutputMode_DisparityC	Disparity 2.5D Depth map. The distance is inversely proportional to the pixel (disparity) value.
Scan3dOutputMode_DisparityC_Linescan	Disparity 2.5D Depth map with varying B sampling. The distance is inversely proportional to the pixel (disparity) value. The B (Y) axis comes from the encoder chunk value.
NUM_SCAN3DOUTPUTMODE	

14.5.1.169 SensorDigitizationTapsEnums

enum [SensorDigitizationTapsEnums](#)

< Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Enumerator

SensorDigitizationTaps_One	1 tap.
SensorDigitizationTaps_Two	2 taps.
SensorDigitizationTaps_Three	3 taps.
SensorDigitizationTaps_Four	4 taps.
SensorDigitizationTaps_Eight	8 taps.
SensorDigitizationTaps_Ten	10 taps.
NUM_SENSORDIGITIZATIONTAPS	

14.5.1.170 SensorShutterModeEnums

enum [SensorShutterModeEnums](#)

< Sets the shutter mode of the device.

Enumerator

SensorShutterMode_Global	The shutter opens and closes at the same time for all pixels. All the pixels are exposed for the same length of time at the same time.
SensorShutterMode_Rolling	The shutter opens and closes sequentially for groups (typically lines) of pixels. All the pixels are exposed for the same length of time but not at the same time.
SensorShutterMode_GlobalReset	The shutter opens at the same time for all pixels but ends in a sequential manner. The pixels are exposed for different lengths of time.
NUM_SENSORSHUTTERMODE	

14.5.1.171 SensorTapsEnums

enum [SensorTapsEnums](#)

< Number of taps of the camera sensor.

Enumerator

SensorTaps_One	1 tap.
SensorTaps_Two	2 taps.
SensorTaps_Three	3 taps.
SensorTaps_Four	4 taps.
SensorTaps_Eight	8 taps.
SensorTaps_Ten	10 taps.
NUM_SENSORTAPS	

14.5.1.172 SequencerConfigurationModeEnums

enum [SequencerConfigurationModeEnums](#)

< Controls whether or not a sequencer is in configuration mode.

Enumerator

SequencerConfigurationMode_Off	
SequencerConfigurationMode_On	
NUM_SEQUENCERCONFIGURATIONMODE	

14.5.1.173 SequencerConfigurationValidEnums

enum [SequencerConfigurationValidEnums](#)

< Display whether the current sequencer configuration is valid to run.

Enumerator

SequencerConfigurationValid_No	
SequencerConfigurationValid_Yes	
NUM_SEQUENCERCONFIGURATIONVALID	

14.5.1.174 SequencerModeEnums

enum [SequencerModeEnums](#)

< Controls whether or not a sequencer is active.

Enumerator

SequencerMode_Off	
SequencerMode_On	
NUM_SEQUENCERMODE	

14.5.1.175 SequencerSetValidEnums

enum [SequencerSetValidEnums](#)

< Displays whether the currently selected sequencer set's register contents are valid to use.

Enumerator

SequencerSetValid_No	
SequencerSetValid_Yes	
NUM_SEQUENCERSETVALID	

14.5.1.176 SequencerTriggerActivationEnums

enum [SequencerTriggerActivationEnums](#)

< Specifies the activation mode of the sequencer trigger.

Enumerator

SequencerTriggerActivation_RisingEdge	
SequencerTriggerActivation_FallingEdge	
SequencerTriggerActivation_AnyEdge	
SequencerTriggerActivation_LevelHigh	
SequencerTriggerActivation_LevelLow	
NUM_SEQUENCERTRIGGERACTIVATION	

14.5.1.177 SequencerTriggerSourceEnums

enum [SequencerTriggerSourceEnums](#)

< Specifies the internal signal or physical input line to use as the sequencer trigger source.

Enumerator

SequencerTriggerSource_Off	
SequencerTriggerSource_FrameStart	
NUM_SEQUENCERTRIGGERSOURCE	

14.5.1.178 SerialPortBaudRateEnums

enum [SerialPortBaudRateEnums](#)

< This feature controls the baud rate used by the selected serial port.

Enumerator

SerialPortBaudRate_Baud300	
SerialPortBaudRate_Baud600	
SerialPortBaudRate_Baud1200	
SerialPortBaudRate_Baud2400	
SerialPortBaudRate_Baud4800	
SerialPortBaudRate_Baud9600	
SerialPortBaudRate_Baud14400	
SerialPortBaudRate_Baud19200	
SerialPortBaudRate_Baud38400	
SerialPortBaudRate_Baud57600	
SerialPortBaudRate_Baud115200	
SerialPortBaudRate_Baud230400	
SerialPortBaudRate_Baud460800	
SerialPortBaudRate_Baud921600	
NUM_SERIALPORTBAUDRATE	

14.5.1.179 SerialPortParityEnums

enum [SerialPortParityEnums](#)

< This feature controls the parity used by the selected serial port.

Enumerator

SerialPortParity_None	
SerialPortParity_Odd	
SerialPortParity_Even	
SerialPortParity_Mark	
SerialPortParity_Space	
NUM_SERIALPORTPARITY	

14.5.1.180 SerialPortSelectorEnums

enum [SerialPortSelectorEnums](#)

< Selects which serial port of the device to control.

Enumerator

SerialPortSelector_SerialPort0	
NUM_SERIALPORTSELECTOR	

14.5.1.181 SerialPortSourceEnums

enum [SerialPortSourceEnums](#)

< Specifies the physical input Line on which to receive serial data.

Enumerator

SerialPortSource_Line0	
SerialPortSource_Line1	
SerialPortSource_Line2	
SerialPortSource_Line3	
SerialPortSource_Off	
NUM_SERIALPORTSOURCE	

14.5.1.182 SerialPortStopBitsEnums

enum [SerialPortStopBitsEnums](#)

< This feature controls the number of stop bits used by the selected serial port.

Enumerator

SerialPortStopBits_Bits1	
SerialPortStopBits_Bits1AndAHalf	
SerialPortStopBits_Bits2	
NUM_SERIALPORTSTOPBITS	

14.5.1.183 SoftwareSignalSelectorEnums

enum [SoftwareSignalSelectorEnums](#)

< Selects which Software Signal features to control.

Enumerator

SoftwareSignalSelector_SoftwareSignal0	Selects the software generated signal to control.
SoftwareSignalSelector_SoftwareSignal1	Selects the software generated signal to control.
SoftwareSignalSelector_SoftwareSignal2	Selects the software generated signal to control.
NUM_SOFTWARESIGNALSELECTOR	

14.5.1.184 SourceDataRange

enum [SourceDataRange](#)

[Image](#) normalization source data options.

Options to normalize the source data based on the max and min values present in the specific image (image data) or the theoretical absolute max and min image data values for the image type (absolute data). By default the absolute max and min values for an image are the max and min values allowable for the image's pixel format. An exception to this is for some computed image data formats such as AoLP, DoLP and Stokes, where the absolute max and min are dependant on the algorithm used.

For a given pixel, normalization is done by: $\text{NormalizedValue} = ((\text{maxDest} - \text{minDest}) * (\text{PixelValue} - \text{minSrc}) / (\text{maxSrc} - \text{minSrc})) + \text{minDest}$

Enumerator

SPINNAKER_SOURCE_DATA_RANGE_IMAGE_↔ DATA_RANGE	Normalize based on the actual max and min values for the source image.
SPINNAKER_SOURCE_DATA_RANGE_ABSOLU↔ TE_DATA_RANGE	Normalize based on the theoretical max and min values for the source image.
SPINNAKER_SOURCE_DATA_RANGE_IMAGE_↔ MIN_ABSOLUTE_MAX	Normalize based on the actual min and theoretical max values for the source image.
SPINNAKER_SOURCE_DATA_RANGE_ABSOLU↔ TE_MIN_IMAGE_MAX	Normalize based on the theoretical min and actual max values for the source image.

14.5.1.185 SourceSelectorEnums

enum [SourceSelectorEnums](#)

< Selects the source to control.

Enumerator

SourceSelector_Source0	Selects the data source 0.
SourceSelector_Source1	Selects the data source 1.
SourceSelector_Source2	Selects the data source 2.
SourceSelector_All	Selects all the data sources.
NUM_SOURCESELECTOR	

14.5.1.186 SpinnakerLogLevel

enum [SpinnakerLogLevel](#)

Log levels.

Enumerator

SPINNAKER_LOG_LEVEL_OFF	
SPINNAKER_LOG_LEVEL_FATAL	
SPINNAKER_LOG_LEVEL_ALERT	
SPINNAKER_LOG_LEVEL_CRIT	
SPINNAKER_LOG_LEVEL_ERROR	
SPINNAKER_LOG_LEVEL_WARN	
SPINNAKER_LOG_LEVEL_NOTICE	
SPINNAKER_LOG_LEVEL_INFO	
SPINNAKER_LOG_LEVEL_DEBUG	
SPINNAKER_LOG_LEVEL_NOTSET	

14.5.1.187 StatisticsChannel

enum [StatisticsChannel](#)

Channels that allow statistics to be calculated.

Enumerator

SPINNAKER_STATISTICS_CHANNEL_GREY	
SPINNAKER_STATISTICS_CHANNEL_RED	
SPINNAKER_STATISTICS_CHANNEL_GREEN	
SPINNAKER_STATISTICS_CHANNEL_BLUE	
SPINNAKER_STATISTICS_CHANNEL_HUE	
SPINNAKER_STATISTICS_CHANNEL_SATURATION	
SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS	
SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS	

14.5.1.188 StreamBufferCountModeEnum

enum [StreamBufferCountModeEnum](#)

< Controls access to setting the number of buffers used for the stream.

Enumerator

StreamBufferCountMode_Manual	The number of buffers used for the stream is set by the user.
NUMSTREAMBUFFERCOUNTMODE	

14.5.1.189 StreamBufferHandlingModeEnum

enum [StreamBufferHandlingModeEnum](#)

< Available buffer handling modes of this data stream:

Enumerator

StreamBufferHandlingMode_OldestFirst	The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.
StreamBufferHandlingMode_OldestFirstOverwrite	The application always gets the buffer from the head of the output buffer queue (thus, the oldest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires. If a new buffer arrives it will overwrite the existing buffer from the head of the queue (behaves like a circular buffer).
StreamBufferHandlingMode_NewestOnly	The application always gets the latest completed buffer (the newest one). If the Output Buffer Queue is empty, the application waits for a newly acquired buffer until the timeout expires. This buffer handling mode is typically used in a live display GUI where it is important that there is no lag between camera and display.
StreamBufferHandlingMode_NewestFirst	The application always gets the buffer from the tail of the output buffer queue (thus, the newest available one). If the output buffer queue is empty, the application waits for a newly acquired buffer until the timeout expires.
NUMSTREAMBUFFERHANDLINGMODE	

14.5.1.190 StreamModeEnum

enum [StreamModeEnum](#)

< Stream mode of the device.

Enumerator

StreamMode_Socket	Socket
StreamMode_LWF	Light Weight Filter Driver
StreamMode_MVA	Machine Vision Accelerator Driver
NUMSTREAMMODE	

14.5.1.191 StreamTypeEnum

enum [StreamTypeEnum](#)

The enum definitions for TL Device nodes from the transport layer .xml files.

< Stream type of the device.

Enumerator

StreamType_GigEVision	GigE Vision
StreamType_CameraLink	Camera Link
StreamType_CameraLinkHS	Camera Link High Speed
StreamType_CoaXPress	CoaXPress
StreamType_USB3Vision	USB3 Vision
StreamType_Custom	Custom transport layer
NUMSTREAMTYPE	

14.5.1.192 TestPatternEnums

enum [TestPatternEnums](#)

< Selects the type of test pattern that is generated by the device as image source.

Enumerator

TestPattern_Off	Test pattern is disabled.
TestPattern_Increment	Pixel value increments by 1 for each pixel.
TestPattern_SensorTestPattern	A test pattern generated by the image sensor. The pattern varies for different sensor models.
NUM_TESTPATTERN	

14.5.1.193 TestPatternGeneratorSelectorEnums

```
enum TestPatternGeneratorSelectorEnums
```

< Selects which test pattern generator is controlled by the TestPattern feature.

Enumerator

TestPatternGeneratorSelector_Sensor	TestPattern feature controls the sensor's test pattern generator.
TestPatternGeneratorSelector_PipelineStart	TestPattern feature controls the test pattern inserted at the start of the image pipeline.
NUM_TESTPATTERNGENERATORSELECTOR	

14.5.1.194 TIFFCompressionMethod

```
enum TIFFCompressionMethod
```

Compression method to use for encoding TIFF images.

Enumerator

SPINNAKER_TIFF_COMPRESS_METHOD_NONE	Save without any compression.
SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS	Save using PACKBITS compression.
SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE	Save using DEFLATE compression (ZLIB compression).
SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE	Save using ADOBE DEFLATE compression.
SPINNAKER_TIFF_COMPRESS_METHOD_CCITT_FAX3	Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
SPINNAKER_TIFF_COMPRESS_METHOD_CCITT_FAX4	Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.
SPINNAKER_TIFF_COMPRESS_METHOD_LZW	Save using LZW compression.
SPINNAKER_TIFF_COMPRESS_METHOD_JPEG	Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths.

14.5.1.195 TimerSelectorEnums

```
enum TimerSelectorEnums
```

< Selects which Timer to configure.

Enumerator

TimerSelector_Timer0	Selects the Timer 0.
TimerSelector_Timer1	Selects the Timer 1.
TimerSelector_Timer2	Selects the Timer 2.
NUM_TIMERSELECTOR	

14.5.1.196 TimerStatusEnums

enum `TimerStatusEnums`

< Returns the current status of the Timer.

Enumerator

TimerStatus_TimerIdle	The Timer is idle.
TimerStatus_TimerTriggerWait	The Timer is waiting for a start trigger.
TimerStatus_TimerActive	The Timer is counting for the specified duration.
TimerStatus_TimerCompleted	The Timer reached the TimerDuration count.
NUM_TIMERSTATUS	

14.5.1.197 TimerTriggerActivationEnums

enum `TimerTriggerActivationEnums`

< Selects the activation mode of the trigger to start the Timer.

Enumerator

TimerTriggerActivation_RisingEdge	Starts counting on the Rising Edge of the selected trigger signal.
TimerTriggerActivation_FallingEdge	Starts counting on the Falling Edge of the selected trigger signal.
TimerTriggerActivation_AnyEdge	Starts counting on the Falling or Rising Edge of the selected trigger signal.
TimerTriggerActivation_LevelHigh	Counts as long as the selected trigger signal level is High.
TimerTriggerActivation_LevelLow	Counts as long as the selected trigger signal level is Low.
NUM_TIMERTRIGGERACTIVATION	

14.5.1.198 TimerTriggerSourceEnums

enum `TimerTriggerSourceEnums`

< Selects the source of the trigger to start the Timer.

Enumerator

TimerTriggerSource_Off	Disables the Timer trigger.
TimerTriggerSource_AcquisitionTrigger	Starts with the reception of the Acquisition Trigger.
TimerTriggerSource_AcquisitionStart	Starts with the reception of the Acquisition Start.
TimerTriggerSource_AcquisitionEnd	Starts with the reception of the Acquisition End.
TimerTriggerSource_FrameTrigger	Starts with the reception of the Frame Start Trigger.
TimerTriggerSource_FrameStart	Starts with the reception of the Frame Start.
TimerTriggerSource_FrameEnd	Starts with the reception of the Frame End.
TimerTriggerSource_FrameBurstStart	Starts with the reception of the Frame Burst Start.
TimerTriggerSource_FrameBurstEnd	Starts with the reception of the Frame Burst End.
TimerTriggerSource_LineTrigger	Starts with the reception of the Line Start Trigger.
TimerTriggerSource_LineStart	Starts with the reception of the Line Start.
TimerTriggerSource_LineEnd	Starts with the reception of the Line End.
TimerTriggerSource_ExposureStart	Starts with the reception of the Exposure Start.
TimerTriggerSource_ExposureEnd	Starts with the reception of the Exposure End.
TimerTriggerSource_Line0	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line1	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_Line2	Starts when the specified TimerTriggerActivation condition is met on the chosen I/O Line.
TimerTriggerSource_UserOutput0	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_UserOutput1	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_UserOutput2	Specifies which User Output bit signal to use as internal source for the trigger.
TimerTriggerSource_Counter0Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter1Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter2Start	Starts with the reception of the Counter Start.
TimerTriggerSource_Counter0End	Starts with the reception of the Counter End.
TimerTriggerSource_Counter1End	Starts with the reception of the Counter End.
TimerTriggerSource_Counter2End	Starts with the reception of the Counter End.
TimerTriggerSource_Timer0Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer1Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer2Start	Starts with the reception of the Timer Start.
TimerTriggerSource_Timer0End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Timer1End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Timer2End	Starts with the reception of the Timer End. Note that a timer can retrigger itself to achieve a free running Timer.
TimerTriggerSource_Encoder0	Starts with the reception of the Encoder output signal.
TimerTriggerSource_Encoder1	Starts with the reception of the Encoder output signal.
TimerTriggerSource_Encoder2	Starts with the reception of the Encoder output signal.
TimerTriggerSource_SoftwareSignal0	Starts on the reception of the Software Signal.
TimerTriggerSource_SoftwareSignal1	Starts on the reception of the Software Signal.
TimerTriggerSource_SoftwareSignal2	Starts on the reception of the Software Signal.
TimerTriggerSource_Action0	Starts with the assertion of the chosen action signal.

Enumerator

TimerTriggerSource_Action1	Starts with the assertion of the chosen action signal.
TimerTriggerSource_Action2	Starts with the assertion of the chosen action signal.
TimerTriggerSource_LinkTrigger0	Starts with the reception of the chosen Link Trigger.
TimerTriggerSource_LinkTrigger1	Starts with the reception of the chosen Link Trigger.
TimerTriggerSource_LinkTrigger2	Starts with the reception of the chosen Link Trigger.
NUM_TIMERTRIGGERSOURCE	

14.5.1.199 TLPayloadType

```
enum TLPayloadType
```

Enumerator

SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN	
SPINNAKER_TLPAYLOAD_TYPE_IMAGE	
SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA	
SPINNAKER_TLPAYLOAD_TYPE_FILE	
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA	
SPINNAKER_TLPAYLOAD_TYPE_JPEG	
SPINNAKER_TLPAYLOAD_TYPE_JPEG2000	
SPINNAKER_TLPAYLOAD_TYPE_H264	
SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY	
SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC	
SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART	
SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID	
SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED	
SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED	
SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED	

14.5.1.200 TLPixelFormatNamespace

```
enum TLPixelFormatNamespace
```

This enum represents the namespace in which the TL specific pixel format resides.

This enum is returned from a captured image when calling [Image::GetTLPixelFormatNamespace\(\)](#). It can be used to interpret the raw pixel format returned from [Image::GetTLPixelFormat\(\)](#).

See also

[Image::GetTLPixelFormat\(\)](#)

[Image::GetTLPixelFormatNamespace\(\)](#)

Enumerator

SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT	
SPINNAKER_TLPIXELFORMAT_NAMESPACE_CUSTOM_ID	

14.5.1.201 TLTypeEnum

enum [TLTypeEnum](#)

< Transport layer type of the GenTL Producer implementation.

Enumerator

TLType_GigEVision	GigE Vision
TLType_CameraLink	Camera Link
TLType_CameraLinkHS	Camera Link High Speed
TLType_CoaXPRESS	CoaXPRESS
TLType_USB3Vision	USB3 Vision
TLType_Mixed	Different Interface modules of the GenTL Producer are of different types
TLType_Custom	Custom transport layer
NUMTLTYPE	

14.5.1.202 TransferComponentSelectorEnums

enum [TransferComponentSelectorEnums](#)

< Selects the color component for the control of the TransferStreamChannel feature.

Enumerator

TransferComponentSelector_Red	The TransferStreamChannel feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats.
TransferComponentSelector_Green	The TransferStreamChannel feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats.
TransferComponentSelector_Blue	The TransferStreamChannel feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats.

Enumerator

TransferComponentSelector_All	The TransferStreamChannel feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats.
NUM_TRANSFERCOMPONENTSELECTOR	

14.5.1.203 TransferControlModeEnums

enum [TransferControlModeEnums](#)

< Selects the control method for the transfers. Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks.

Enumerator

TransferControlMode_Basic	Basic
TransferControlMode_Automatic	Automatic
TransferControlMode_UserControlled	User Controlled
NUM_TRANSFERCONTROLMODE	

14.5.1.204 TransferOperationModeEnums

enum [TransferOperationModeEnums](#)

< Selects the operation mode of the transfer. Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops.

Enumerator

TransferOperationMode_Continuous	Continuous
TransferOperationMode_MultiBlock	Multi Block
NUM_TRANSFEROPERATIONMODE	

14.5.1.205 TransferQueueModeEnums

enum [TransferQueueModeEnums](#)

< Specifies the operation mode of the transfer queue.

Enumerator

TransferQueueMode_FirstInFirstOut	Blocks first In are transferred Out first.
NUM_TRANSFERQUEUEMODE	

14.5.1.206 TransferSelectorEnums

enum [TransferSelectorEnums](#)

< Selects which stream transfers are currently controlled by the selected Transfer features.

Enumerator

TransferSelector_Stream0	The transfer features control the data stream 0.
TransferSelector_Stream1	The transfer features control the data stream 1.
TransferSelector_Stream2	The transfer features control the data stream 2.
TransferSelector_All	The transfer features control all the data streams simulateneously.
NUM_TRANSFERSELECTOR	

14.5.1.207 TransferStatusSelectorEnums

enum [TransferStatusSelectorEnums](#)

< Selects which status of the transfer module to read.

Enumerator

TransferStatusSelector_Streaming	Data blocks are transmitted when enough data is available.
TransferStatusSelector_Paused	Data blocks transmission is suspended immediately.
TransferStatusSelector_Stopping	Data blocks transmission is stopping. The current block transmission will be completed and the transfer state will stop.
TransferStatusSelector_Stopped	Data blocks transmission is stopped.
TransferStatusSelector_QueueOverflow	Data blocks queue is in overflow state.
NUM_TRANSFERSTATUSSELECTOR	

14.5.1.208 TransferTriggerActivationEnums

enum [TransferTriggerActivationEnums](#)

< Specifies the activation mode of the transfer control trigger.

Enumerator

TransferTriggerActivation_RisingEdge	Specifies that the trigger is considered valid on the rising edge of the source signal.
TransferTriggerActivation_FallingEdge	Specifies that the trigger is considered valid on the falling edge of the source signal.
TransferTriggerActivation_AnyEdge	Specifies that the trigger is considered valid on the falling or rising edge of the source signal.
TransferTriggerActivation_LevelHigh	Specifies that the trigger is considered valid as long as the level of the source signal is high. This can apply to TransferActive and TransferPause trigger.
TransferTriggerActivation_LevelLow	Specifies that the trigger is considered valid as long as the level of the source signal is low. This can apply to TransferActive and TransferPause trigger.
NUM_TRANSFERTRIGGERACTIVATION	

14.5.1.209 TransferTriggerModeEnums

```
enum TransferTriggerModeEnums
```

< Controls if the selected trigger is active.

Enumerator

TransferTriggerMode_Off	Disables the selected trigger.
TransferTriggerMode_On	Enable the selected trigger.
NUM_TRANSFERTRIGGERMODE	

14.5.1.210 TransferTriggerSelectorEnums

```
enum TransferTriggerSelectorEnums
```

< Selects the type of transfer trigger to configure.

Enumerator

TransferTriggerSelector_TransferStart	Selects a trigger to start the transfers.
TransferTriggerSelector_TransferStop	Selects a trigger to stop the transfers.
TransferTriggerSelector_TransferAbort	Selects a trigger to abort the transfers.
TransferTriggerSelector_TransferPause	Selects a trigger to pause the transfers.
TransferTriggerSelector_TransferResume	Selects a trigger to Resume the transfers.
TransferTriggerSelector_TransferActive	Selects a trigger to Activate the transfers. This trigger type is used when TriggerActivation is set LevelHigh or levelLow.
TransferTriggerSelector_TransferBurstStart	Selects a trigger to start the transfer of a burst of frames specified by TransferBurstCount.
TransferTriggerSelector_TransferBurstStop	Selects a trigger to end the transfer of a burst of frames.
NUM_TRANSFERTRIGGERSELECTOR	

14.5.1.211 TransferTriggerSourceEnums

enum [TransferTriggerSourceEnums](#)

< Specifies the signal to use as the trigger source for transfers.

Enumerator

TransferTriggerSource_Line0	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Line1	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Line2	Specifies which physical line (or pin) and associated I/O control block to use as external source for the transfer control trigger signal.
TransferTriggerSource_Counter0Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter1Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter2Start	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter0End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter1End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Counter2End	Specifies which of the Counter signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer0Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer1Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer2Start	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer0End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer1End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Timer2End	Specifies which Timer signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal0	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal1	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_SoftwareSignal2	Specifies which Software Signal to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action0	Specifies which Action command to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action1	Specifies which Action command to use as internal source for the transfer control trigger signal.
TransferTriggerSource_Action2	Specifies which Action command to use as internal source for the transfer control trigger signal.
NUM_TRANSFERTRIGGERSOURCE	

14.5.1.212 TriggerActivationEnums

enum `TriggerActivationEnums`

< Specifies the activation mode of the trigger.

Enumerator

TriggerActivation_LevelLow	
TriggerActivation_LevelHigh	
TriggerActivation_FallingEdge	
TriggerActivation_RisingEdge	
TriggerActivation_AnyEdge	
NUM_TRIGGERACTIVATION	

14.5.1.213 TriggerModeEnums

enum `TriggerModeEnums`

< Controls whether or not trigger is active.

Enumerator

TriggerMode_Off	
TriggerMode_On	
NUM_TRIGGERMODE	

14.5.1.214 TriggerOverlapEnums

enum `TriggerOverlapEnums`

< Specifies the overlap mode of the trigger.

Enumerator

TriggerOverlap_Off	
TriggerOverlap_ReadOut	
TriggerOverlap_PreviousFrame	
NUM_TRIGGEROVERLAP	

14.5.1.215 TriggerSelectorEnums

enum [TriggerSelectorEnums](#)

< Selects the type of trigger to configure.

Enumerator

TriggerSelector_AcquisitionStart	
TriggerSelector_FrameStart	
TriggerSelector_FrameBurstStart	
NUM_TRIGGERSELECTOR	

14.5.1.216 TriggerSourceEnums

enum [TriggerSourceEnums](#)

< Specifies the internal signal or physical input line to use as the trigger source.

Enumerator

TriggerSource_Software	
TriggerSource_Line0	
TriggerSource_Line1	
TriggerSource_Line2	
TriggerSource_Line3	
TriggerSource_UserOutput0	
TriggerSource_UserOutput1	
TriggerSource_UserOutput2	
TriggerSource_UserOutput3	
TriggerSource_Counter0Start	
TriggerSource_Counter1Start	
TriggerSource_Counter0End	
TriggerSource_Counter1End	
TriggerSource_LogicBlock0	
TriggerSource_LogicBlock1	
TriggerSource_Action0	
NUM_TRIGGERSOURCE	

14.5.1.217 UserOutputSelectorEnums

enum [UserOutputSelectorEnums](#)

< Selects which bit of the User Output register is set by UserOutputValue.

Enumerator

UserOutputSelector_UserOutput0	
UserOutputSelector_UserOutput1	
UserOutputSelector_UserOutput2	
UserOutputSelector_UserOutput3	
NUM_USEROUTPUTSELECTOR	

14.5.1.218 UserSetDefaultEnums

```
enum UserSetDefaultEnums
```

< Selects the feature User Set to load and make active by default when the device is restarted.

Enumerator

UserSetDefault_Default	Factory default set.
UserSetDefault_UserSet0	User configurable set 0.
UserSetDefault_UserSet1	User configurable set 1.
NUM_USERSETDEFAULT	

14.5.1.219 UserSetSelectorEnums

```
enum UserSetSelectorEnums
```

< Selects the feature User Set to load, save or configure.

Enumerator

UserSetSelector_Default	Factory default set.
UserSetSelector_UserSet0	User configurable set 0.
UserSetSelector_UserSet1	User configurable set 1.
NUM_USERSETSELECTOR	

14.5.1.220 WhiteClipSelectorEnums

```
enum WhiteClipSelectorEnums
```

< Selects which White Clip to control.

Enumerator

WhiteClipSelector_All	White Clip will be applied to all channels or taps.
WhiteClipSelector_Red	White Clip will be applied to the red channel.
WhiteClipSelector_Green	White Clip will be applied to the green channel.
WhiteClipSelector_Blue	White Clip will be applied to the blue channel.
WhiteClipSelector_Y	White Clip will be applied to Y channel.
WhiteClipSelector_U	White Clip will be applied to U channel.
WhiteClipSelector_V	White Clip will be applied to V channel.
WhiteClipSelector_Tap1	White Clip will be applied to Tap 1.
WhiteClipSelector_Tap2	White Clip will be applied to Tap 2.
NUM_WHITECLIPSELECTOR	

14.5.2 Function Documentation

14.5.2.1 DEPRECATED_CLASS()

```
class Spinnaker::DEPRECATED_CLASS (
    "AVIRecorder is deprecated,
    use SpinVideo instead." )
```

Provides the functionality for the user to record images to an AVI file.

NOTE: This class is deprecated and replaced by SpinVideo. Refer to [SpinVideo.h](#) instead. Default constructor.

Default destructor.

Open an AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	Options to apply to the AVI file.

See also

AVIClose()

Open an MJPEG AVI file in preparation for writing Images to disk. The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the AVI file.
<i>pOption</i>	MJPEG options to apply to the AVI file.

See also

AVIClose()
MJPGOption

Open an H264 MP4 file in preparation for writing Images to disk. The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the MP4 file.
<i>pOption</i>	H264 options to apply to the MP4 file.

See also

AVIClose()
H264Option

Append an image to the AVI/MP4 file.

Parameters

<i>pImage</i>	The image to append.

Close the AVI/MP4 file.

See also

AVIOpen()

Set the maximum file size (in megabytes) of a AVI/MP4 file. A new AVI/MP4 file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

Parameters

<i>size</i>	The maximum AVI file size in MB.

See also

AVIAppend(ImagePtr pImage)

14.5.2.2 operator==()

```
bool Spinnaker::operator==(
    const std::nullptr_t ,
    const BasePtr< T, B > & rhs ) [inline]
```

Pointer equal.

14.5.3 Variable Documentation

14.5.3.1 EVENT_TIMEOUT_INFINITE

```
const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFFFF
```

14.5.3.2 EVENT_TIMEOUT_NONE

```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

Timeout values for getting next image, device, or interface event.

14.6 Spinnaker::GenApi Namespace Reference

Classes

- struct [AttachStatistics_t](#)
Delivers information about the attached chunks and nodes.
- class [AutoLock](#)
- class [BooleanNode](#)
Interface for string properties.
- class [CategoryNode](#)
Interface for string properties.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.
- class [CChunkAdapterDcam](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.
- class [CEnumerationTRef](#)
Interface for string properties.
- class [CEventAdapter](#)
Delivers Events to ports.
- class [CEventAdapter1394](#)
Distribute the events to the node map.
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.
- class [CEventAdapterGEV](#)

- Connects a GigE Event to a node map.*
- class [CEventAdapterU3V](#)
 - Connects a U3V Event to a node map.*
- class [CEventPort](#)
 - Port attachable to an event.*
- class [CFeatureBag](#)
 - Bag holding streamable features of a nodetree.*
- class [CFloatPtr](#)
 - SmartPointer for IFloat interface pointer.*
- class [CGeneric_XMLLoaderParams](#)
 - Empty base class used by class [CNodeMapRef](#) as generic template argument.*
- class [CLock](#)
 - A lock class.*
- class [CLockEx](#)
 - This class is for testing purposes only.*
- class [CNodeCallback](#)
 - callback body instance for INode pointers*
- class [CNodeMapFactory](#)
 - The node map factory is used for creating node maps from camera description files.*
- class [CNodeMapRef](#)
 - Smartpointer for NodeMaps with create function.*
- class [CNodeMapRefT](#)
 - Smartpointer template for NodeMaps with create function.*
- class [CommandNode](#)
 - [Interface](#) for string properties.*
- class [Counter](#)
 - Definition of a simple [Counter](#) class.*
- class [CPointer](#)
 - Encapsulates a [GenApi](#) pointer dealing with the dynamic_cast automatically.*
- class [CPortImpl](#)
 - Standard implementation for a port.*
- class [CPortWriteList](#)
 - Container holding a list of port write commands.*
- class [CRegisterPortImpl](#)
 - Standard implementation for a port using a register based transport layer.*
- class [CSelectorSet](#)
 - The set of selectors selecting a given node.*
- class [CTestPortStruct](#)
 - Implements a register spaces based on a C++ struct.*
- struct [DCAM_CHECKSUM](#)
- struct [DCAM_CHUNK_TRAILER](#)
- class [double_autovector_t](#)
 - Vector of doubles with reference counting.*
- class [EAccessModeClass](#)
 - Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)
 - Holds conversion methods for the caching mode enumeration.*
- class [EDisplayNotationClass](#)
 - Holds conversion methods for the notation type of floats.*
- class [EEndianessClass](#)
 - Holds conversion methods for the endianess enumeration.*

- class [EGenApiSchemaVersionClass](#)
helper class converting EGenApiSchemaVersion from and to string
- class [EInputDirectionClass](#)
Holds conversion methods for the notation type of floats.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EnumEntryNode](#)
Interface for string properties.
- class [EnumNode](#)
Interface for string properties.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [ESlopeClass](#)
Holds conversion methods for the converter formulas.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [FileProtocolAdapter](#)
Adapter between the std::iostreambuf and the SFNC Features representing the device file system.
- class [FloatNode](#)
Interface for string properties.
- class [FloatRegNode](#)
Interface for string properties.
- class [Function_NodeCallback](#)
Container for a function pointer.
- struct [GVCP_CHUNK_TRAILER](#)
header of a GVCP request packet
- struct [GVCP_EVENT_ITEM](#)
layout of a GVCP event item (Extended ID flag not set)
- struct [GVCP_EVENT_ITEM_BASIC](#)
layout of a GVCP event item (common to all types)
- struct [GVCP_EVENT_ITEM_EXTENDED_ID](#)
layout of a GVCP event item (Extended ID flag set)
- struct [GVCP_EVENT_REQUEST](#)
Layout of a GVCP event request packet (Extended ID flag not set)
- struct [GVCP_EVENT_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event request packet (Extended ID flag set)
- struct [GVCP_EVENTDATA_REQUEST](#)
Layout of a GVCP event data request packet (Extended ID flag not set)
- struct [GVCP_EVENTDATA_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event data request packet (Extended ID flag set)
- struct [GVCP_REQUEST_HEADER](#)
header of a GVCP request packet
- class [IDevFileStreamBase](#)
- class [IDevFileStreamBuf](#)
- class [int64_autovector_t](#)

- Vector of integers with reference counting.*
- class [IntegerNode](#)
 - Interface for string properties.*
- class [IntRegNode](#)
 - Interface for string properties.*
- class [Member_NodeCallback](#)
 - Container for a member function pointer.*
- class [Node](#)
 - class common to all nodes*
- class [NodeMap](#)
 - Smart pointer template for NodeMaps with create function.*
- class [ODevFileStreamBase](#)
- class [ODevFileStreamBuf](#)
- class [PortNode](#)
 - Interface for value properties.*
- class [PortRecorder](#)
 - Interface for recording write commands on a port.*
- class [PortReplay](#)
 - Interface for replaying write commands on a port.*
- class [RegisterNode](#)
 - Interface for string properties.*
- struct [SingleChunkData_t](#)
- struct [SingleChunkDataStr_t](#)
- class [SpinTestCamera](#)
- class [StringNode](#)
 - Interface for string properties.*
- class [StringRegNode](#)
 - Interface for string properties.*
- struct [U3V_CHUNK_TRAILER](#)
 - header of a GVCP request packet*
- struct [U3V_COMMAND_HEADER](#)
 - U3V/GenCP command header.*
- struct [U3V_EVENT_DATA](#)
 - U3V/GenCP EVENT_CMD specific command data.*
- struct [U3V_EVENT_MESSAGE](#)
 - Entire event data message (without the variable-sized data field)*
- class [ValueNode](#)
 - Interface for value properties.*

Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)
- typedef [CategoryNode](#) [CCategoryRef](#)
- typedef [CommandNode](#) [CCommandRef](#)
- typedef [EnumEntryNode](#) [CEnumEntryRef](#)
- typedef [EnumNode](#) [CEnumerationRef](#)
- typedef [ODevFileStreamBase](#)< char, std::char_traits< char > > [ODevFileStream](#)
- typedef [IDevFileStreamBase](#)< char, std::char_traits< char > > [IDevFileStream](#)
- typedef [FloatNode](#) [CFloatRef](#)
- typedef node_vector [NodeList_t](#)
 - a list of node references*

- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes
- typedef [IntegerNode](#) [CIntegerRef](#)
- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)
- typedef [CPointer](#)< [IBase](#) > [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef [CPointer](#)< [INode](#), [IBase](#) > [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef [CPointer](#)< [IValue](#) > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef [CPointer](#)< [ICategory](#) > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef [CPointer](#)< [IBoolean](#) > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef [CPointer](#)< [IInteger](#) > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef [CPointer](#)< [IString](#) > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef [CPointer](#)< [IRegister](#) > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef [CPointer](#)< [IEnumeration](#) > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef [CPointer](#)< [IEnumEntry](#) > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef [CPointer](#)< [IPort](#) > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef [CPointer](#)< [IPortReplay](#) > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef [CPointer](#)< [IPortRecorder](#) > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.
- typedef [PortNode](#) [CPortRef](#)
- typedef [PortRecorder](#) [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.

- typedef [RegisterNode](#) [CRegisterRef](#)
- typedef [StringNode](#) [CStringRef](#)
- typedef [GenICam::gcstring_vector](#) [StringList_t](#)
A list of strings.
- typedef [ValueNode](#) [CValueRef](#)

Enumerations

- enum [GVCP_MESSAGE_TAGS](#) {
[TAG_EVENT_CMD](#) = 0xc0,
[TAG_EVENTDATA_CMD](#) = 0xc2 }
- enum [_ECallbackType](#) {
[cbPostInsideLock](#) = 1,
[cbPostOutsideLock](#) = 2 }
the type of callback
- enum [ECacheUsage_t](#) {
[CacheUsage_Automatic](#),
[CacheUsage_ForceWrite](#),
[CacheUsage_ForceRead](#),
[CacheUsage_Ignore](#) }
Lists the cache usage strategies.
- enum [EContentType_t](#) {
[ContentType_Xml](#),
[ContentType_ZippedXml](#) }
Lists the processable file types.
- enum [_ESign](#) {
[Signed](#),
[Unsigned](#),
[_UndefinedSign](#) }
signed or unsigned integers
- enum [_EAccessMode](#) {
[NI](#),
[NA](#),
[WO](#),
[RO](#),
[RW](#),
[_UndefinedAccesMode](#),
[_CycleDetectAccesMode](#) }
access mode of a node
- enum [_EVisibility](#) {
[Beginner](#) = 0,
[Expert](#) = 1,
[Guru](#) = 2,
[Invisible](#) = 3,
[_UndefinedVisibility](#) = 99 }
recommended visibility of a node
- enum [_ECachingMode](#) {
[NoCache](#),
[WriteThrough](#),
[WriteAround](#),
[_UndefinedCachingMode](#) }
caching mode of a register

- enum [_ERepresentation](#) {
[Linear](#),
[Logarithmic](#),
[Boolean](#),
[PureNumber](#),
[HexNumber](#),
[IPV4Address](#),
[MACAddress](#),
[_UndefinedRepresentation](#) }
recommended representation of a node value
- enum [_EEndianness](#) {
[BigEndian](#),
[LittleEndian](#),
[_UndefinedEndian](#) }
Endianness of a value in a register.
- enum [_ENamespace](#) {
[Custom](#),
[Standard](#),
[_UndefinedNameSpace](#) }
Defines if a node name is standard or custom.
- enum [_EStandardNameSpace](#) {
[None](#),
[GEV](#),
[IIDC](#),
[CL](#),
[USB](#),
[_UndefinedStandardNameSpace](#) }
Defines from which standard namespace a node name comes from.
- enum [_EYesNo](#) {
[Yes](#) = 1,
[No](#) = 0,
[_UndefinedYesNo](#) = 2 }
Defines the choices of a Yes/No alternative.
- enum [_ESlope](#) {
[Increasing](#),
[Decreasing](#),
[Varying](#),
[Automatic](#),
[_UndefinedESlope](#) }
typedef for formula type
- enum [_EXMLValidation](#) {
[xvLoad](#) = 0x00000001L,
[xvCycles](#) = 0x00000002L,
[xvSFNC](#) = 0x00000004L,
[xvDefault](#) = 0x00000000L,
[xvAll](#) = 0xffffffffL,
[_UndefinedEXMLValidation](#) = 0x80000000L }
typedef describing the different validity checks which can be performed on an XML file
- enum [_EDisplayNotation](#) {
[fnAutomatic](#),
[fnFixed](#),
[fnScientific](#),
[_UndefinedEDisplayNotation](#) }
typedef for float notation
- enum [_EInterfaceType](#) {
[intfIValue](#),


```

intfIBase,
intfInteger,
intfBoolean,
intfCommand,
intfFloat,
intfString,
intfRegister,
intfCategory,
intfEnumeration,
intfEnumEntry,
intfIPort }

    typedef for interface type
• enum _ELinkType {
    ctParentNodes,
    ctReadingChildren,
    ctWritingChildren,
    ctInvalidatingChildren,
    ctDependingNodes,
    ctTerminalNodes }

    typedef for link type
• enum _EIncMode {
    noIncrement,
    fixedIncrement,
    listIncrement }

    typedef for increment mode
• enum _EInputDirection {
    idFrom,
    idTo,
    idNone }

    typedef for link type
• enum _EGenApiSchemaVersion {
    v1_0 = 1,
    v1_1 = 2,
    _Undefined = -1 }

    GenApi schema version.

```

Functions

- void [SPINNAKER_API SET_GUID](#) (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)
- virtual void [operator=](#) (bool Value)
 - Set node value.*
- virtual bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const =0
 - Get node value.*
- virtual bool [operator\(\)](#) () const
 - Get node value.*
- virtual EYesNo [CacheChunkData](#) () const =0
 - Indicates if the chunk a adapter must hold a cached version of the chunk data.*
- virtual bool [IsDone](#) (bool [Verify](#)=true)=0
 - Query whether the command is executed.*
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()=0
 - Get the vendor name.*
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()=0

- Get tool tip.*

 - virtual [GenICam::gcstring GetStandardNameSpace](#) ()=0

Get the standard name space.
- virtual void [GetGenApiVersion](#) ([GenICam::Version_t](#) &Version, [uint16_t](#) &Build)=0

Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) ([GenICam::Version_t](#) &Version)=0

Get the schema version number.
- virtual void [GetDeviceVersion](#) ([GenICam::Version_t](#) &Version)=0

Get the version of the device description file.
- virtual [GenICam::gcstring GetProductGuid](#) ()=0

Get the Guid describing the product.
- virtual [GenICam::gcstring GetVersionGuid](#) ()=0

Get the Guid describing the product version.
- virtual [GenICam::gcstring GetSymbolic](#) () const =0

Get symbolic enum value.
- virtual double [GetNumericValue](#) ()=0

Get double number associated with the entry.
- virtual bool [IsSelfClearing](#) ()=0

Indicates if the corresponding [EnumEntry](#) is self clearing.
- virtual void [GetEntries](#) ([NodeList_t](#) &Entries)=0

Get list of entry nodes.
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)=0

Set string node value.
- virtual void [SetIntValue](#) ([int64_t](#) Value, bool [Verify](#)=true)=0

Set integer node value.
- virtual [GenICam::gcstring operator*](#) ()=0

Get string node value.
- virtual [int64_t](#) [GetIntValue](#) (bool [Verify](#)=false, bool [IgnoreCache](#)=false)=0

Get integer node value.
- virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)=0

Get an entry node by name.
- virtual [IEnumEntry](#) * [GetEntry](#) (const [int64_t](#) IntValue)=0

Get an entry node by its [IntValue](#).
- virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool [IgnoreCache](#)=false)=0

Get the current entry.
- virtual [IEnumeration](#) & [operator=](#) ([EnumT](#) Value)=0

Set node value.
- virtual [IEnumEntry](#) * [GetEntry](#) (const [EnumT](#) Value)=0

returns the [EnumEntry](#) object belonging to the [Value](#)
- virtual [IFloat](#) & [operator=](#) (double Value)=0

Set node value.
- virtual double [GetMin](#) ()=0

Get minimum value allowed.
- virtual double [GetMax](#) ()=0

Get maximum value allowed.
- virtual bool [HasInc](#) ()=0

True if the float has a constant increment.
- virtual [EIncMode](#) [GetIncMode](#) ()=0

Get increment mode.
- virtual double [GetInc](#) ()=0

Get the constant increment if there is any.

- virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual [ERepresentation](#) [GetRepresentation](#) ()=0
Get recommended representation.
- virtual [GenICam::gcstring](#) [GetUnit](#) () const =0
Get the physical unit name.
- virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const =0
Get the way the float should be converted to a string.
- virtual [int64_t](#) [GetDisplayPrecision](#) () const =0
Get the precision to be used when converting the float to a string.
- virtual void [ImposeMin](#) (double Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)=0
Restrict maximum value.
- virtual [Integer](#) & [operator=](#) (int64_t Value)=0
Set node value.
- virtual void [ImposeMin](#) (int64_t Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)=0
Restrict maximum value.
- virtual [GenApi::ENamespace](#) [GetNameSpace](#) () const =0
Get name space.
- virtual [EVisibility](#) [GetVisibility](#) () const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()=0
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const =0
Is the node value cacheable.
- virtual [EYesNo](#) [IsAccessModeCacheable](#) () const =0
True if the AccessMode can be cached.
- virtual [ECachingMode](#) [GetCachingMode](#) () const =0
Get Caching Mode.
- virtual [int64_t](#) [GetPollingTime](#) () const =0
recommended polling time (for non-cacheable nodes)
- virtual [GenICam::gcstring](#) [GetDescription](#) () const =0
Get a long description of the node.
- virtual [GenICam::gcstring](#) [GetDisplayName](#) () const =0
Get a name string for display.
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const =0
Get a name of the device.
- virtual void [GetChildren](#) ([GenApi::NodeList_t](#) &Children, [ELinkType](#) LinkType=[ctReadingChildren](#)) const =0
Get all nodes this node directly depends on.
- virtual void [GetParents](#) ([GenApi::NodeList_t](#) &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType](#) [RegisterCallback](#) ([CNodeCallback](#) *pCallback)=0
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)=0
De register change callback Destroys CNodeCallback object.
- virtual [INodeMap](#) * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual [GenICam::gcstring](#) [GetEventID](#) () const =0

- Get the EventId of the node.*

 - virtual bool `IsStreamable` () const =0

True if the node is streamable.
- virtual void `GetPropertyNames` (GenICam::gcstring_vector &PropertyNames) const =0

Returns a list of the names all properties set during initialization.
- virtual bool `GetProperty` (const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)=0

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void `ImposeAccessMode` (EAccessMode ImposedAccessMode)=0

Imposes an access mode to the natural access mode of the node.
- virtual void `ImposeVisibility` (EVisibility ImposedVisibility)=0

Imposes a visibility to the natural visibility of the node.
- virtual INode * `GetAlias` () const =0

Retrieves the a node which describes the same feature in a different way.
- virtual INode * `GetCastAlias` () const =0

Retrieves the a node which describes the same feature so that it can be casted.
- virtual GenICam::gcstring `GetDocuURL` () const =0

Gets a URL pointing to the documentation of that feature.
- virtual bool `IsDeprecated` () const =0

True if the node should not be used any more.
- virtual EInterfaceType `GetPrincipalInterfaceType` () const =0

Get the type of the main interface of a node.
- virtual bool `IsFeature` () const =0

True if the node can be reached via category nodes from a category node named "Root".
- virtual bool `operator==` (int nullPtr) const =0
- virtual bool `operator!=` (int nullPtr) const =0
- bool `IsReadable` (EAccessMode AccessMode)

Tests if readable.
- bool `IsReadable` (const IBase *p)

Checks if a node is readable.
- bool `IsReadable` (const IBase &r)

Checks if a node is readable.
- bool `IsWritable` (EAccessMode AccessMode)

Tests if writable.
- bool `IsWritable` (const IBase *p)

Checks if a node is writable.
- bool `IsWritable` (const IBase &r)

Checks if a node is writable.
- bool `IsImplemented` (EAccessMode AccessMode)

Tests if implemented.
- bool `IsImplemented` (const IBase *p)

Checks if a node is implemented.
- bool `IsImplemented` (const IBase &r)

Checks if a node is implemented.
- bool `IsAvailable` (EAccessMode AccessMode)

Tests if available.
- bool `IsAvailable` (const IBase *p)

Checks if a node is available.
- bool `IsAvailable` (const IBase &r)

Checks if a node is available.

- EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)
Computes which access mode the two guards allow together.
- bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)
Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)
Computes which visibility the two guards allow together.
- bool [IsCacheable](#) (ECachingMode CachingMode)
Tests Cacheability.
- ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)
Computes which CachingMode results from a combination.
- virtual [INode * GetNode](#) (const [GenICam::gcstring](#) &Name) const =0
Retrieves the node from the central map by Name.
- virtual void [InvalidateNodes](#) () const =0
Invalidates all nodes.
- virtual bool [Connect](#) ([IPort](#) *pPort, const [GenICam::gcstring](#) &PortName) const =0
Connects a port to a port node with given name.
- virtual bool [Connect](#) ([IPort](#) *pPort) const =0
Connects a port to the standard port "Device".
- virtual void [Poll](#) (int64_t ElapsedTime)=0
Fires nodes which have a polling time.
- virtual [CLock](#) & [GetLock](#) () const =0
Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const =0
Get the number of nodes in the map.
- virtual void [LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)=0
Loads an XML from a file.
- virtual void [LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)=0
Loads an XML from a file with injection.
- virtual void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)=0
Loads an XML from a string.
- virtual void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)=0
Loads an XML from a string with injection.
- virtual void [PreprocessXMLFromFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xvDefault](#))=0
Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void [MergeXMLFiles](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectedFileName, const [GenICam::gcstring](#) &OutputFileName)=0
Injects an XML file into a target file.
- virtual void [ExtractIndependentSubtree](#) (const [GenICam::gcstring](#) &XMLData, const [GenICam::gcstring](#) &InjectXMLData, const [GenICam::gcstring](#) &SubTreeRootNodeName, [GenICam::gcstring](#) &ExtractedSubtree)=0
Extract independent subtree.
- virtual void [GetSupportedSchemaVersions](#) ([GenICam::gcstring_vector](#) &SchemaVersions)=0
Gets a list of supported schema versions.
- virtual void [LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)=0
Loads an XML from a ZIP file.
- virtual void [LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)=0
Loads an XML from a ZIP data buffer.

- virtual void [PreprocessXMLFromZIPFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xvDefault](#))=0
Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Writes a chunk of bytes to the port.
- virtual EYesNo [GetSwapEndianness](#) ()=0
Determines if the port adapter must perform an endianness swap.
- virtual void [Replay](#) (IPort *pPort)=0
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)=0
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()=0
Gets the cookie a port implementation may have set for caching a command list.
- virtual void [StopRecording](#) ()=0
Stops recording.
- virtual void [Get](#) (uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=false, bool IgnoreCache=false)=0
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()=0
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()=0
Retrieves the Address of the register.
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node
- virtual bool [SetNext](#) (bool Tick=true)=0
Sets digit to next value.
- virtual void [Restore](#) ()=0
Restores the selectors' values found at creation.
- virtual [GenICam::gcstring](#) [ToString](#) ()=0
Returns a string representation of the digit.
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)=0
Retrieves an ordered list of selectors.
- virtual int64_t [GetMaxLength](#) ()=0
Retrieves the maximum length of the string in bytes.
- virtual [GenICam::gcstring](#) [ToString](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get content of the node as string.
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool [Verify](#)=true)=0
Set content of the node as string.
- virtual bool [IsValueCacheValid](#) () const =0
Checks if the value comes from cache or is requested from another node.
- template<class Function >
[CNodeCallback](#) * [make_NodeCallback](#) (INode *pNode, Function function, ECallbackType CallbackType)
make a new callback object for C functions
- template<class Function >
intptr_t [Register](#) (INode *pNode, Function f, ECallbackType CallbackType=[cbPostInsideLock](#))
Register a C-function as a callback.
- template<class Client , class Member >
[CNodeCallback](#) * [make_NodeCallback](#) (INode *pNode, Client &client, Member member, ECallbackType CallbackType)

- make a new callback object for member functions*
- `template<class Client , class Member >`
`intptr_t Register (INode *pNode, Client &c, Member m, ECallbackType CallbackType=cbPostInsideLock)`
Register a C++-member function a callback.
- [SPINNAKER_API](#) `void Deregister (GenApi::CallbackHandleType pCallbackInfo)`
Unregistering callback by handle.
- [SPINNAKER_API](#) `IDestroy * CastToIDestroy (INodeMap *pNodeMap)`
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- `template<class TCameraParams >`
`void _LoadXMLFromFile (const GenICam::gcstring &FileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromZIPFile (const GenICam::gcstring &ZipFileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromFileInject (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromString (const GenICam::gcstring &XMLData)`
- `template<class TCameraParams >`
`void _LoadXMLFromZIPData (const void *zipData, size_t zipSize)`
- `template<class TCameraParams >`
`void _LoadXMLFromStringInject (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)`
- `template<class TCameraParams >`
`void _GetSupportedSchemaVersions (GenICam::gcstring\_vector &SchemaVersions)`
- `template<class TCameraParams >`
`GenICam::gcstring _GetDeviceName ()`
- `template<class TCameraParams >`
`void _Poll (int64_t ElapsedTime)`
- `template<class TCameraParams >`
`void _GetNodes (NodeList\_t &Nodes)`
- `template<class TCameraParams >`
`INode * _GetNode (const GenICam::gcstring &key)`
- `template<class TCameraParams >`
`void _InvalidateNodes ()`
- `template<class TCameraParams >`
`bool _Connect (IPort *pPort, const GenICam::gcstring &PortName)`
- `template<class TCameraParams >`
`bool _Connect (IPort *pPort)`
- `template<class TCameraParams >`
`bool _ClearXMLCache ()`
- `virtual void PersistFeature (IValue &item)=0`
Stores a feature.
- [SPINNAKER_API](#) `std::istream & EatComments (std::istream &is)`
Helper function ignoring lines starting with comment character '#'.
- [SPINNAKER_API](#) `std::istream & operator>> (std::istream &is, CFeatureBag &FeatureBag)`
Reads in persistent data from a stream.
- [SPINNAKER_API](#) `std::ostream & operator<< (std::ostream &os, const CFeatureBag &FeatureBag)`
writes out persistent data to a stream
- `template<class T , class B >`
`bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is readable.
- `template<class T , class B >`
`bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Writable.

- `template<class T, class B >`
`bool IsImplemented (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Implemented.
- `template<class T, class B >`
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Available.
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`
Returns the name of the main interface as string DEPRICATED, use `IBase::GetPrincipalInterfaceType()` instead.
- `virtual void SetNumEnums (int NumEnums)=0`
sets the number of enum values

Variables

- `interface SPINNAKER_API_ABSTRACT IBase`
Base interface common to all nodes.
- `const uint8_t COMMAND_MAGIC = 0x42`
- `const uint32_t U3V_EVENT_PREFIX = 0x45563355`
- `const uint16_t GENCP_EVENT_CMD_ID = 0x0C00`
- `const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)`
- `const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)`
- `interface SPINNAKER_API_ABSTRACT IBoolean`
Interface for Boolean properties.
- `interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0`
- `interface SPINNAKER_API_ABSTRACT ICategory`
Gives access to a category node.
- `interface SPINNAKER_API_ABSTRACT IChunkPort`
Interface for ports attached to a chunk.
- `interface SPINNAKER_API_ABSTRACT ICommand`
Interface for command like properties.
- `interface SPINNAKER_API_ABSTRACT IDestroy`
Interface to destroy an object.
- `interface SPINNAKER_API_ABSTRACT IDeviceInfo`
Interface to get information about the device (= nodemap)
- `interface SPINNAKER_API_ABSTRACT IEnumEntry`
Interface of single enum value.
- `interface SPINNAKER_API_ABSTRACT IEnumeration`
Interface for enumeration properties.
- `template<typename EnumT >`
`interface SPINNAKER_API_ABSTRACT IEnumerationT`
Interface for enumeration properties.
- `interface SPINNAKER_API_ABSTRACT virtual public IEnumReference`
Interface to construct an enum reference.
- `interface SPINNAKER_API_ABSTRACT IFloat`
Interface for float properties.
- `interface SPINNAKER_API_ABSTRACT IInteger`
Interface for integer properties.
- `interface SPINNAKER_API_ABSTRACT INode`
Interface common to all nodes.
- `interface SPINNAKER_API_ABSTRACT virtual public IReference`
Interface to construct a reference.
- `interface SPINNAKER_API_ABSTRACT INodeMap`

- *Interface to access the node map.*
- [interface SPINNAKER_API_ABSTRACT INodeMapDyn](#)
- *Interface to access the node map.*
- [interface SPINNAKER_API_ABSTRACT IPort](#)
- *Interface for ports.*
- [interface SPINNAKER_API_ABSTRACT int64_t Address](#)
- [interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0](#)
- [interface SPINNAKER_API IPortConstruct](#)
- *Interface for ports.*
- [interface SPINNAKER_API_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER_API_ABSTRACT IPortReplay](#)
- *Interface for replaying write commands on a port.*
- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)
- *Interface for recording write commands on a port.*
- [interface SPINNAKER_API_ABSTRACT IRegister](#)
- *Interface for registers.*
- [interface SPINNAKER_API_ABSTRACT ISelector](#)
- *Interface for groups of features selected by a single one.*
- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)
- *Interface of a "digit" of the "counter" formed by the selector set.*
- [interface SPINNAKER_API_ABSTRACT IString](#)
- *Interface for string properties.*
- [interface SPINNAKER_API_ABSTRACT IValue](#)
- *Interface for value properties.*
- [interface SPINNAKER_API_ABSTRACT IPersistScript](#)
- *Basic interface to persist values to.*

14.6.1 Typedef Documentation

14.6.1.1 CallbackHandleType

```
typedef intptr_t CallbackHandleType
```

the callback handle for nodes

14.6.1.2 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

14.6.1.3 CCategoryRef

typedef [CategoryNode](#) CCategoryRef

14.6.1.4 CCommandRef

typedef [CommandNode](#) CCommandRef

14.6.1.5 CEnumEntryRef

typedef [EnumEntryNode](#) CEnumEntryRef

14.6.1.6 CEnumerationRef

typedef [EnumNode](#) CEnumerationRef

14.6.1.7 CFloatRef

typedef [FloatNode](#) CFloatRef

14.6.1.8 CIntegerRef

typedef [IntegerNode](#) CIntegerRef

14.6.1.9 CNodeMapRef

typedef [NodeMap](#) CNodeMapRef

14.6.1.10 CNodeRef

typedef [Node](#) CNodeRef

14.6.1.11 CPortRecorderRef

```
typedef PortRecorder CPortRecorderRef
```

Reference to an IPortRecorder pointer.

14.6.1.12 CPortRef

```
typedef PortNode CPortRef
```

14.6.1.13 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

14.6.1.14 CSelectorRef

```
typedef Node CSelectorRef
```

14.6.1.15 CStringRef

```
typedef StringNode CStringRef
```

14.6.1.16 CValueRef

```
typedef ValueNode CValueRef
```

14.6.1.17 IDevFileStream

```
typedef IDevFileStreamBase<char, std::char_traits<char> > IDevFileStream
```

14.6.1.18 NodeList_t

```
typedef node_vector NodeList\_t
```

a list of node references

14.6.1.19 ODevFileStream

```
typedef ODevFileStreamBase<char, std::char_traits<char> > ODevFileStream
```

14.6.1.20 StringList_t

```
typedef GenICam::gcstring_vector StringList\_t
```

A list of strings.

14.6.2 Enumeration Type Documentation

14.6.2.1 _EAccessMode

```
enum \_EAccessMode
```

access mode of a node

Enumerator

NI	
NA	Not implemented.
WO	Not available.
RO	Write Only.
RW	Read Only.
_UndefinedAccesMode	Read and Write.
_CycleDetectAccesMode	Object is not yet initialized. used internally for AccessMode cycle detection

14.6.2.2 _ECachingMode

```
enum \_ECachingMode
```

caching mode of a register

Enumerator

NoCache	
WriteThrough	Do not use cache.
WriteAround	Write to cache and register.
_UndefinedCachingMode	Write to register, write to cache on read. Not yet initialized

14.6.2.3 _ECallbackType

```
enum _ECallbackType
```

the type of callback

Enumerator

cbPostInsideLock	
cbPostOutsideLock	callback is fired on leaving the tree inside the lock-guarded area

14.6.2.4 _EDisplayNotation

```
enum _EDisplayNotation
```

typedef for float notation

Enumerator

fnAutomatic	
fnFixed	the notation if either scientific or fixed depending on what is shorter
fnScientific	the notation is fixed, e.g. 123.4
_UndefinedEDisplayNotation	the notation is scientific, e.g. 1.234e2 Object is not yet initialized

14.6.2.5 _EEndianness

```
enum _EEndianness
```

Endianness of a value in a register.

Enumerator

BigEndian	
LittleEndian	Register is big endian.
_UndefinedEndian	Register is little endian. Object is not yet initialized

14.6.2.6 _EGenApiSchemaVersion

```
enum _EGenApiSchemaVersion
```

GenApi schema version.

Enumerator

v1_0	
v1_1	
_Undefined	

14.6.2.7 _EIncMode

```
enum _EIncMode
```

typedef for increment mode

Enumerator

noIncrement	
fixedIncrement	The feature has no increment.
listIncrement	The feature has a fix increment.

14.6.2.8 _EInputDirection

```
enum _EInputDirection
```

typedef for link type

Enumerator

idFrom	
idTo	Indicates a swiss knife that it is used as worker for a converter computing FROM.
idNone	Indicates a swiss knife that it is used as worker for a converter computing TO. SwissKnife is not used within a converter

14.6.2.9 _EInterfaceType

enum [_EInterfaceType](#)

typedef for interface type

Enumerator

intfIValue	
intfIBase	IValue interface.
intfIInteger	IBase interface.
intfIBoolean	IInteger interface.
intfICommand	IBoolean interface.
intfIFloat	ICommand interface.
intfIString	IFloat interface.
intfIRegister	IString interface.
intfICategory	IRegister interface.
intfIEnumeration	ICategory interface.
intfIEnumEntry	IEnumeration interface.
intfIPort	IEnumEntry interface. IPort interface

14.6.2.10 _ELinkType

enum [_ELinkType](#)

typedef for link type

Enumerator

ctParentNodes	
ctReadingChildren	All nodes for which this node is at least an invalidating child.
ctWritingChildren	All nodes which can be read from.
ctInvalidatingChildren	All nodes which can write a value further down the node stack.
ctDependingNodes	All directly connected nodes which invalidate this node.
ctTerminalNodes	All directly or indirectly connected nodes which are invalidated by this nodes (i.e. which are dependent on this node) All indirectly connected terminal nodes

14.6.2.11 _ENamespace

enum [_ENamespace](#)

Defines if a node name is standard or custom.

Enumerator

Custom	
Standard	name resides in custom namespace
_UndefinedNameSpace	name resides in one of the standard namespaces Object is not yet initialized

14.6.2.12 _ERepresentation

enum `_ERepresentation`

recommended representation of a node value

Enumerator

Linear	
Logarithmic	Slider with linear behavior.
Boolean	Slider with logarithmic behavior.
PureNumber	Check box.
HexNumber	Decimal number in an edit control.
IPV4Address	Hex number in an edit control.
MACAddress	IP-Address.
_UndefinedRepresentation	MAC-Address.

14.6.2.13 _ESign

enum `_ESign`

signed or unsigned integers

Enumerator

Signed	
Unsigned	Integer is signed.
_UndefinedSign	Integer is unsigned. Object is not yet initialized

14.6.2.14 _ESlope

enum `_ESlope`

typedef for formula type

Enumerator

Increasing	
Decreasing	strictly monotonous increasing
Varying	strictly monotonous decreasing
Automatic	slope changes, e.g. at run-time
_UndefinedESlope	slope is determined automatically by probing the function Object is not yet initialized

14.6.2.15 _EStandardNameSpace

enum [_EStandardNameSpace](#)

Defines from which standard namespace a node name comes from.

Enumerator

None	
GEV	name resides in custom namespace
IIDC	name resides in GigE Vision namespace
CL	name resides in 1394 IIDC namespace
USB	name resides in camera link namespace
_UndefinedStandardNameSpace	name resides in USB namespace Object is not yet initialized

14.6.2.16 _EVisibility

enum [_EVisibility](#)

recommended visibility of a node

Enumerator

Beginner	
Expert	Always visible.
Guru	Visible for experts or Gurus.
Invisible	Visible for Gurus.
_UndefinedVisibility	Not Visible.

14.6.2.17 _EXMLValidation

enum [_EXMLValidation](#)

typedef describing the different validity checks which can be performed on an XML file

The enum values for a bit field of length uint32_t

Enumerator

xvLoad	
xvCycles	Creates a dummy node map.
xvSFNC	checks for write and dependency cycles (implies xvLoad)
xvDefault	checks for conformance with the standard feature naming convention (SFNC)
xvAll	checks performed if nothing else is said
_UndefinedEXMLValidation	all possible checks

14.6.2.18 _EYesNo

enum [_EYesNo](#)

Defines the choices of a Yes/No alternative.

Enumerator

Yes	
No	yes
_UndefinedYesNo	no

14.6.2.19 ECacheUsage_t

enum [ECacheUsage_t](#)

Lists the cache usage strategies.

The cache stores preprocessed camera description xml files providing faster access or smaller footprint. note The environment variable GENICAM_CACHE_VERSION, e.g. GENICAM_CACHE_V3_0, must contain the path to cache directory for using the cache.

Enumerator

CacheUsage_Automatic	The use of cache files is determined automatically.
CacheUsage_ForceWrite	Forces the loading and preprocessing of the camera description xml file. If a cache directory is available the result of preprocessing is written to the cache.
CacheUsage_ForceRead	Suppresses loading and preprocessing of the camera description xml file and. forces reading a cache file from cache directory. Fails if no matching cache file is available.
CacheUsage_Ignore	Forces the loading and preprocessing of the camera description xml file. No cache file is written.

14.6.2.20 EContentType_t

```
enum EContentType_t
```

Lists the processable file types.

Enumerator

ContentType_Xml	XML camera description file text.
ContentType_ZippedXml	Zipped XML camera description file text.

14.6.2.21 GVCP_MESSAGE_TAGS

```
enum GVCP_MESSAGE_TAGS
```

Enumerator

TAG_EVENT_CMD	
TAG_EVENTDATA_CMD	

14.6.3 Function Documentation

14.6.3.1 _ClearXMLCache()

```
bool Spinnaker::GenApi::_ClearXMLCache ( ) [inline]
```

14.6.3.2 _Connect() [1/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort ) [inline]
```

14.6.3.3 _Connect() [2/2]

```
bool Spinnaker::GenApi::_Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [inline]
```

14.6.3.4 `_GetDeviceName()`

```
GenICam::gcstring Spinnaker::GenApi::_GetDeviceName ( ) [inline]
```

14.6.3.5 `_GetNode()`

```
INode* Spinnaker::GenApi::_GetNode (
    const GenICam::gcstring & key ) [inline]
```

14.6.3.6 `_GetNodes()`

```
void Spinnaker::GenApi::_GetNodes (
    NodeList_t & Nodes ) [inline]
```

14.6.3.7 `_GetSupportedSchemaVersions()`

```
void Spinnaker::GenApi::_GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) [inline]
```

14.6.3.8 `_InvalidateNodes()`

```
void Spinnaker::GenApi::_InvalidateNodes ( ) [inline]
```

14.6.3.9 `_LoadXMLFromFile()`

```
void Spinnaker::GenApi::_LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [inline]
```

14.6.3.10 `_LoadXMLFromFileInject()`

```
void Spinnaker::GenApi::_LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [inline]
```

14.6.3.11 _LoadXMLFromString()

```
void Spinnaker::GenApi::_LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [inline]
```

14.6.3.12 _LoadXMLFromStringInject()

```
void Spinnaker::GenApi::_LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [inline]
```

14.6.3.13 _LoadXMLFromZIPData()

```
void Spinnaker::GenApi::_LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [inline]
```

14.6.3.14 _LoadXMLFromZIPFile()

```
void Spinnaker::GenApi::_LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [inline]
```

14.6.3.15 _Poll()

```
void Spinnaker::GenApi::_Poll (
    int64_t ElapsedTime ) [inline]
```

14.6.3.16 CacheChunkData()

```
virtual EYesNo Spinnaker::GenApi::CacheChunkData ( ) const [pure virtual]
```

Indicates if the chunk a adapter must hold a cached version of the chunk data.

14.6.3.17 CastToIDestroy()

```
SPINNAKER_API IDestroy* Spinnaker::GenApi::CastToIDestroy (
    INodeMap * pNodeMap )
```

makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)

14.6.3.18 Connect() [1/2]

```
virtual bool Spinnaker::GenApi::Connect (
    IPort * pPort ) const [pure virtual]
```

Connects a port to the standard port "Device".

14.6.3.19 Connect() [2/2]

```
virtual bool Spinnaker::GenApi::Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) const [pure virtual]
```

Connects a port to a port node with given name.

14.6.3.20 Deregister()

```
SPINNAKER_API void Spinnaker::GenApi::Deregister (
    GenApi::CallbackHandleType pCallbackInfo )
```

Unregistering callback by handle.

14.6.3.21 DeregisterCallback()

```
virtual bool Spinnaker::GenApi::DeregisterCallback (
    CallbackHandleType hCallback ) [pure virtual]
```

De register change callback Destroys [CNodeCallback](#) object.

Returns

true if the callback handle was valid

14.6.3.22 EatComments()

```
SPINNAKER_API std::istream& Spinnaker::GenApi::EatComments (
    std::istream & is )
```

Helper function ignoring lines starting with comment character '#'.

14.6.3.23 ExtractIndependentSubtree()

```
virtual void Spinnaker::GenApi::ExtractIndependentSubtree (
    const GenICam::gcstring & XMLData,
    const GenICam::gcstring & InjectXMLData,
    const GenICam::gcstring & SubTreeRootNodeName,
    GenICam::gcstring & ExtractedSubtree ) [pure virtual]
```

Extract independent subtree.

Parameters

<i>InjectXMLData</i>	> The XML data the subtree is extracted from.
<i>SubTreeRootNodeName</i>	> Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed.
<i>ExtractedSubtree</i>	> The name of the node that represents the root of the subtree that shall be extracted.> The returned extracted subtree as string.

14.6.3.24 FromString()

```
virtual void Spinnaker::GenApi::FromString (
    const GenICam::gcstring & ValueStr,
    bool Verify = true ) [pure virtual]
```

Set content of the node as string.

Parameters

<i>ValueStr</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

14.6.3.25 Get()

```
virtual void Spinnaker::GenApi::Get (
    uint8_t * pBuffer,
```

```

    int64_t Length,
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]

```

Fills a buffer with the register's contents.

Parameters

<i>pBuffer</i>	The buffer receiving the data to read
<i>Length</i>	The number of bytes to retrieve
<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

14.6.3.26 GetAddress()

```
virtual int64_t Spinnaker::GenApi::GetAddress ( ) [pure virtual]
```

Retrieves the Address of the register.

14.6.3.27 GetAlias()

```
virtual INode* Spinnaker::GenApi::GetAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature in a different way.

14.6.3.28 GetCachingMode()

```
virtual ECachingMode Spinnaker::GenApi::GetCachingMode ( ) const [pure virtual]
```

Get Caching Mode.

14.6.3.29 GetCastAlias()

```
virtual INode* Spinnaker::GenApi::GetCastAlias ( ) const [pure virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

14.6.3.30 GetChildren()

```

virtual void Spinnaker::GenApi::GetChildren (
    GenApi::NodeList_t & Children,
    ELinkType LinkType = ctReadingChildren ) const [pure virtual]

```

Get all nodes this node directly depends on.

Parameters

out	<i>Children</i>	List of children nodes
	<i>LinkType</i>	The link type

14.6.3.31 GetCookie()

```
virtual int64_t Spinnaker::GenApi::GetCookie ( ) [pure virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

14.6.3.32 GetCurrentEntry()

```
IEnumEntry * GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get the current entry.

14.6.3.33 GetDescription()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDescription ( ) const [pure virtual]
```

Get a long description of the node.

14.6.3.34 GetDeviceName()

```
GenICam::gcstring GetDeviceName ( ) [pure virtual]
```

Get a name of the device.

Get device name The device name identifies a device instance, e.g.

for debugging purposes. The default is "Device".

14.6.3.35 GetDeviceVersion()

```
virtual void Spinnaker::GenApi::GetDeviceVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the version of the device description file.

14.6.3.36 GetDisplayName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDisplayName ( ) const [pure virtual]
```

Get a name string for display.

14.6.3.37 GetDisplayNotation()

```
virtual EDisplayNotation Spinnaker::GenApi::GetDisplayNotation ( ) const [pure virtual]
```

Get the way the float should be converted to a string.

14.6.3.38 GetDisplayPrecision()

```
virtual int64\_t Spinnaker::GenApi::GetDisplayPrecision ( ) const [pure virtual]
```

Get the precision to be used when converting the float to a string.

14.6.3.39 GetDocuURL()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetDocuURL ( ) const [pure virtual]
```

Gets a URL pointing to the documentation of that feature.

14.6.3.40 GetEntries()

```
virtual void Spinnaker::GenApi::GetEntries (
    NodeList\_t & Entries ) [pure virtual]
```

Get list of entry nodes.

14.6.3.41 GetEntry() [1/2]

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntry (
    const EnumT Value ) [pure virtual]
```

returns the EnumEntry object belonging to the Value

14.6.3.42 GetEntry() [2/2]

```
IEnumEntry * GetEntry (
    const int64_t IntValue ) [pure virtual]
```

Get an entry node by its IntValue.

14.6.3.43 GetEntryByName()

```
virtual IEnumEntry* Spinnaker::GenApi::GetEntryByName (
    const GenICam::gcstring & Symbolic ) [pure virtual]
```

Get an entry node by name.

14.6.3.44 GetEventID()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetEventID ( ) const [pure virtual]
```

Get the EventId of the node.

14.6.3.45 GetGenApiVersion()

```
virtual void Spinnaker::GenApi::GetGenApiVersion (
    GenICam::Version_t & Version,
    uint16_t & Build ) [pure virtual]
```

Get the version of the DLL's GenApi implementation.

14.6.3.46 GetInc()

```
int64_t GetInc ( ) [pure virtual]
```

Get the constant increment if there is any.

Get increment.

14.6.3.47 GetIncMode()

```
EIncMode GetIncMode ( ) [pure virtual]
```

Get increment mode.

14.6.3.48 GetIntValue()

```
virtual int64_t Spinnaker::GenApi::GetIntValue (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get integer node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

14.6.3.49 GetLength()

```
virtual int64_t Spinnaker::GenApi::GetLength ( ) [pure virtual]
```

Retrieves the Length of the register [Bytes].

14.6.3.50 GetListOfValidValues()

```
int64_autovector_t GetListOfValidValues (
    bool bounded = true ) [pure virtual]
```

Get list of valid value.

14.6.3.51 GetLock()

```
virtual CLock& Spinnaker::GenApi::GetLock ( ) const [pure virtual]
```

Returns the lock which guards the node map.

14.6.3.52 GetMax()

```
int64_t GetMax ( ) [pure virtual]
```

Get maximum value allowed.

14.6.3.53 GetMaxLength()

```
virtual int64_t Spinnaker::GenApi::GetMaxLength ( ) [pure virtual]
```

Retrieves the maximum length of the string in bytes.

14.6.3.54 GetMin()

```
int64_t GetMin ( ) [pure virtual]
```

Get minimum value allowed.

14.6.3.55 GetNameSpace()

```
virtual GenApi::ENamespace Spinnaker::GenApi::GetNameSpace ( ) const [pure virtual]
```

Get name space.

14.6.3.56 GetNode()

```
virtual INode* Spinnaker::GenApi::GetNode (
    const GenICam::gcstring & Name ) const [pure virtual]
```

Retrieves the node from the central map by Name.

14.6.3.57 GetNodeMap()

```
virtual INodeMap* Spinnaker::GenApi::GetNodeMap ( ) const [pure virtual]
```

Retrieves the central node map.

14.6.3.58 GetNumericValue()

```
virtual double Spinnaker::GenApi::GetNumericValue ( ) [pure virtual]
```

Get double number associated with the entry.

14.6.3.59 GetNumNodes()

```
virtual uint64_t Spinnaker::GenApi::GetNumNodes ( ) const [pure virtual]
```

Get the number of nodes in the map.

14.6.3.60 GetParents()

```
virtual void Spinnaker::GenApi::GetParents (
    GenApi::NodeList_t & Parents ) const [pure virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	<i>Parents</i>	List of parent nodes

14.6.3.61 GetPollingTime()

```
virtual int64_t Spinnaker::GenApi::GetPollingTime ( ) const [pure virtual]
```

recommended polling time (for non-cacheable nodes)

14.6.3.62 GetPrincipalInterfaceType()

```
virtual EInterfaceType Spinnaker::GenApi::GetPrincipalInterfaceType ( ) const [pure virtual]
```

Get the type of the main interface of a node.

14.6.3.63 GetProductGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetProductGuid ( ) [pure virtual]
```

Get the Guid describing the product.

14.6.3.64 GetProperty()

```
virtual bool Spinnaker::GenApi::GetProperty (
    const GenICam::gcstring & PropertyName,
    GenICam::gcstring & ValueStr,
    GenICam::gcstring & AttributeStr ) [pure virtual]
```

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

14.6.3.65 GetPropertyNames()

```
virtual void Spinnaker::GenApi::GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [pure virtual]
```

Returns a list of the names all properties set during initialization.

14.6.3.66 GetRepresentation()

```
ERepresentation GetRepresentation ( ) [pure virtual]
```

Get recommended representation.

14.6.3.67 GetSchemaVersion()

```
virtual void Spinnaker::GenApi::GetSchemaVersion (
    GenICam::Version_t & Version ) [pure virtual]
```

Get the schema version number.

14.6.3.68 GetSelectedFeatures()

```
virtual void Spinnaker::GenApi::GetSelectedFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of selected features

14.6.3.69 GetSelectingFeatures()

```
virtual void Spinnaker::GenApi::GetSelectingFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of features selecting this node

14.6.3.70 GetSelectorList()

```
virtual void Spinnaker::GenApi::GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [pure virtual]
```

Retrieves an ordered list of selectors.

Parameters

<i>Incremental</i>	> List to contain the selector pointer> if true only selector changed since the last GetNext are contained

14.6.3.71 GetStandardNameSpace()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetStandardNameSpace ( ) [pure virtual]
```

Get the standard name space.

14.6.3.72 GetSupportedSchemaVersions()

```
virtual void Spinnaker::GenApi::GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [pure virtual]
```

Gets a list of supported schema versions.

Each list entry is a string with the format "<Major>.<Minor>" where <Major> and <Minor> are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

14.6.3.73 GetSwapEndianess()

```
virtual EYesNo Spinnaker::GenApi::GetSwapEndianess ( ) [pure virtual]
```

Determines if the port adapter must perform an endianess swap.

14.6.3.74 GetSymbolic()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetSymbolic ( ) const [pure virtual]
```

Get symbolic enum value.

14.6.3.75 GetToolTip()

```
GenICam::gcstring GetToolTip ( ) [pure virtual]
```

Get tool tip.

Get a short description of the node.

14.6.3.76 GetUnit()

```
GenICam::gcstring GetUnit ( ) const [pure virtual]
```

Get the physical unit name.

14.6.3.77 GetValue()

```
GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) const [pure virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

14.6.3.78 GetVendorName()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVendorName ( ) [pure virtual]
```

Get the vendor name.

14.6.3.79 GetVersionGuid()

```
virtual GenICam::gcstring Spinnaker::GenApi::GetVersionGuid ( ) [pure virtual]
```

Get the Guid describing the product version.

14.6.3.80 GetVisibility()

```
virtual EVisibility Spinnaker::GenApi::GetVisibility ( ) const [pure virtual]
```

Get the recommended visibility of the node.

14.6.3.81 HasInc()

```
virtual bool Spinnaker::GenApi::HasInc ( ) [pure virtual]
```

True if the float has a constant increment.

14.6.3.82 ImposeAccessMode()

```
virtual void Spinnaker::GenApi::ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [pure virtual]
```

Imposes an access mode to the natural access mode of the node.

14.6.3.83 ImposeMax() [1/2]

```
virtual void Spinnaker::GenApi::ImposeMax (
    double Value ) [pure virtual]
```

Restrict maximum value.

14.6.3.84 ImposeMax() [2/2]

```
virtual void Spinnaker::GenApi::ImposeMax (
    int64_t Value ) [pure virtual]
```

Restrict maximum value.

14.6.3.85 ImposeMin() [1/2]

```
virtual void Spinnaker::GenApi::ImposeMin (
    double Value ) [pure virtual]
```

Restrict minimum value.

14.6.3.86 ImposeMin() [2/2]

```
virtual void Spinnaker::GenApi::ImposeMin (
    int64_t Value ) [pure virtual]
```

Restrict minimum value.

14.6.3.87 ImposeVisibility()

```
virtual void Spinnaker::GenApi::ImposeVisibility (
    EVisibility ImposedVisibility ) [pure virtual]
```

Imposes a visibility to the natural visibility of the node.

14.6.3.88 InvalidateNode()

```
virtual void Spinnaker::GenApi::InvalidateNode ( ) [pure virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

14.6.3.89 InvalidateNodes()

```
virtual void Spinnaker::GenApi::InvalidateNodes ( ) const [pure virtual]
```

Invalidates all nodes.

14.6.3.90 IsAccessModeCacheable()

```
virtual EYesNo Spinnaker::GenApi::IsAccessModeCacheable ( ) const [pure virtual]
```

True if the AccessMode can be cached.

14.6.3.91 IsCachable()

```
virtual bool Spinnaker::GenApi::IsCachable ( ) const [pure virtual]
```

Is the node value cacheable.

14.6.3.92 IsDeprecated()

```
virtual bool Spinnaker::GenApi::IsDeprecated ( ) const [pure virtual]
```

True if the node should not be used any more.

14.6.3.93 IsDone()

```
virtual bool Spinnaker::GenApi::IsDone (
    bool Verify = true ) [pure virtual]
```

Query whether the command is executed.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

14.6.3.94 IsFeature()

```
virtual bool Spinnaker::GenApi::IsFeature ( ) const [pure virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

14.6.3.95 IsSelfClearing()

```
virtual bool Spinnaker::GenApi::IsSelfClearing ( ) [pure virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

14.6.3.96 IsStreamable()

```
virtual bool Spinnaker::GenApi::IsStreamable ( ) const [pure virtual]
```

True if the node is streamable.

14.6.3.97 IsValueCacheValid()

```
virtual bool Spinnaker::GenApi::IsValueCacheValid ( ) const [pure virtual]
```

Checks if the value comes from cache or is requested from another node.

14.6.3.98 LoadXMLFromFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [pure virtual]
```

Loads an XML from a file.

14.6.3.99 LoadXMLFromFileInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [pure virtual]
```

Loads an XML from a file with injection.

14.6.3.100 LoadXMLFromString()

```
virtual void Spinnaker::GenApi::LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [pure virtual]
```

Loads an XML from a string.

14.6.3.101 LoadXMLFromStringInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [pure virtual]
```

Loads an XML from a string with injection.

14.6.3.102 LoadXMLFromZIPData()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [pure virtual]
```

Loads an XML from a ZIP data buffer.

14.6.3.103 LoadXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [pure virtual]
```

Loads an XML from a ZIP file.

14.6.3.104 make_NodeCallback() [1/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType )
```

make a new callback object for member functions

14.6.3.105 make_NodeCallback() [2/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
    INode * pNode,
    Function function,
    ECallbackType CallbackType )
```

make a new callback object for C functions

14.6.3.106 MergeXMLFiles()

```
virtual void Spinnaker::GenApi::MergeXMLFiles (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectedFileName,
    const GenICam::gcstring & OutputFileName ) [pure virtual]
```

Injects an XML file into a target file.

Parameters

<i>InjectedFileName</i>	> Name of the target XML file to process
<i>OutputFileName</i>	> Name of the Injected XML file to process > Name of the output file

14.6.3.107 operator"!=()

```
virtual bool Spinnaker::GenApi::operator!=(
    int nullptr ) const [pure virtual]
```

14.6.3.108 operator()()

```
GenICam::gcstring operator() ( ) const [virtual]
```

Get node value.

Execute the command.

14.6.3.109 operator*()

```
GenICam::gcstring operator* ( ) [pure virtual]
```

Get string node value.

Get node value.

14.6.3.110 operator<<()

```
SPINNAKER_API std::ostream& Spinnaker::GenApi::operator<< (
    std::ostream & os,
    const CFeatureBag & FeatureBag )
```

writes out persistent data to a stream

14.6.3.111 operator=() [1/5]

```
virtual void Spinnaker::GenApi::operator= (
    bool Value ) [virtual]
```

Set node value.

14.6.3.112 operator=() [2/5]

```
IString & operator= (
    const GenICam::gcstring & ValueStr ) [pure virtual]
```

Set string node value.

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

14.6.3.113 operator=() [3/5]

```
virtual IFloat& Spinnaker::GenApi::operator= (
    double Value ) [pure virtual]
```

Set node value.

14.6.3.114 operator=() [4/5]

```
virtual IEnumeration& Spinnaker::GenApi::operator= (
    EnumT Value ) [pure virtual]
```

Set node value.

14.6.3.115 operator=() [5/5]

```
virtual Integer& Spinnaker::GenApi::operator= (
    int64_t Value ) [pure virtual]
```

Set node value.

14.6.3.116 operator==()

```
virtual bool Spinnaker::GenApi::operator== (
    int nullptr ) const [pure virtual]
```

14.6.3.117 operator>>()

```
SPINNAKER_API std::istream& Spinnaker::GenApi::operator>> (
    std::istream & is,
    CFeatureBag & FeatureBag )
```

Reads in persistent data from a stream.

14.6.3.118 PersistFeature()

```
virtual void Spinnaker::GenApi::PersistFeature (
    IValue & item ) [pure virtual]
```

Stores a feature.

14.6.3.119 Poll()

```
virtual void Spinnaker::GenApi::Poll (
    int64_t ElapsedTime ) [pure virtual]
```

Fires nodes which have a polling time.

14.6.3.120 PreprocessXMLFromFile()

```
virtual void Spinnaker::GenApi::PreprocessXMLFromFile (
    const GenICam::gcstring & XMLFileName,
    const GenICam::gcstring & StyleSheetFileName,
    const GenICam::gcstring & OutputFileName,
    const uint32_t XMLValidation = xvDefault ) [pure virtual]
```

Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Parameters

<i>StyleSheetFileName</i>	> The name of the XML file to process
<i>OutputFileName</i>	> Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the OutputFileName is an empty string
<i>XMLValidation</i>	> Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file

14.6.3.121 PreprocessXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::PreprocessXMLFromZIPFile (
    const GenICam::gcstring & XMLFileName,
    const GenICam::gcstring & StyleSheetFileName,
    const GenICam::gcstring & OutputFileName,
    const uint32_t XMLValidation = xvDefault ) [pure virtual]
```

Loads a Zipped XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.

Parameters

<i>StyleSheetFileName</i>	> The name of the XML file to process
<i>OutputFileName</i>	> Optional name of a style sheet which is applied after the pre-processor (can be empty string)> This has no effect if the OutputFileName is an empty string
<i>XMLValidation</i>	> Optional name of an output file into which the processed data is written (can be empty string)> Optional bit mask formed from EXMLValidation enumeration indicating which tests should be performed on the XML file

14.6.3.122 Register() [1/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Client & c,
    Member m,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C++-member function a callback.

14.6.3.123 Register() [2/2]

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Function f,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C-function as a callback.

14.6.3.124 RegisterCallback()

```
virtual CallbackHandleType Spinnaker::GenApi::RegisterCallback (
    CNodeCallback * pCallback ) [pure virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

14.6.3.125 Replay()

```
virtual void Spinnaker::GenApi::Replay (
    IPort * pPort ) [pure virtual]
```

Replays the write command to the given port interface.

14.6.3.126 Restore()

```
virtual void Spinnaker::GenApi::Restore ( ) [pure virtual]
```

Restores the selectors' values found at creation.

14.6.3.127 SET_GUID()

```
void SPINNAKER_API Spinnaker::GenApi::SET_GUID (
    SPIN_GUID & name,
    uint32_t l,
    uint16_t w1,
    uint16_t w2,
    uint8_t b1,
    uint8_t b2,
    uint8_t b3,
    uint8_t b4,
    uint8_t b5,
    uint8_t b6,
    uint8_t b7,
    uint8_t b8 )
```

14.6.3.128 SetCookie()

```
virtual void Spinnaker::GenApi::SetCookie (
    const int64_t Value ) [pure virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

14.6.3.129 SetIntValue()

```
virtual void Spinnaker::GenApi::SetIntValue (
    int64_t Value,
    bool Verify = true ) [pure virtual]
```

Set integer node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

14.6.3.130 SetNext()

```
virtual bool Spinnaker::GenApi::SetNext (
    bool Tick = true ) [pure virtual]
```

Sets digit to next value.

Parameters

<i>Tick</i>	if false the counter does not tick (but realize it could have)

Returns

true if the resulting value is valid

14.6.3.131 SetNumEnums()

```
virtual void Spinnaker::GenApi::SetNumEnums (
    int NumEnums ) [pure virtual]
```

sets the number of enum values

14.6.3.132 StopRecording()

```
virtual void Spinnaker::GenApi::StopRecording ( ) [pure virtual]
```

Stops recording.

14.6.3.133 ToString() [1/2]

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString ( ) [pure virtual]
```

Returns a string representation of the digit.

14.6.3.134 ToString() [2/2]

```
virtual GenICam::gcstring Spinnaker::GenApi::ToString (
    bool Verify = false,
    bool IgnoreCache = false ) [pure virtual]
```

Get content of the node as string.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

14.6.3.135 Write()

```
virtual void Spinnaker::GenApi::Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes a chunk of bytes to the port.

14.6.4 Variable Documentation**14.6.4.1 Address**

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

14.6.4.2 COMMAND_MAGIC

```
const uint8_t COMMAND_MAGIC = 0x42
```

14.6.4.3 GENCP_COMMAND_HEADER_SIZE

```
const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)
```

14.6.4.4 GENCP_EVENT_BASIC_SIZE

```
const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)
```

14.6.4.5 GENCP_EVENT_CMD_ID

```
const uint16_t GENCP_EVENT_CMD_ID = 0x0C00
```

14.6.4.6 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

Initial value:

```
{  
    virtual EAccessMode GetAccessMode() const = 0
```

Base interface common to all nodes.

14.6.4.7 IBoolean

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

[Interface](#) for Boolean properties.

14.6.4.8 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

14.6.4.9 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

[Interface](#) for ports attached to a chunk.

14.6.4.10 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

[Interface](#) for command like properties.

14.6.4.11 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

Initial value:

```
{  
  
    virtual void Destroy() = 0
```

Interface to destroy an object.

14.6.4.12 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

Initial value:

```
{  
  
    virtual GenICam::gcstring GetModelName() = 0
```

Interface to get information about the device (= nodemap)

14.6.4.13 IEnumEntry

```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

Interface of single enum value.

Maps of Enum Values to symbolic values

14.6.4.14 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

Interface for enumeration properties.

14.6.4.15 IEnumerationT

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

Interface for enumeration properties.

14.6.4.16 IEnumReference

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

Initial value:

```
{  
  
    virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

Interface to construct an enum reference.

14.6.4.17 IFloat

```
interface SPINNAKER_API_ABSTRACT IFloat
```

Interface for float properties.

14.6.4.18 IInteger

```
interface SPINNAKER_API_ABSTRACT IInteger
```

Interface for integer properties.

14.6.4.19 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

Interface common to all nodes.

14.6.4.20 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

Initial value:

```
{  
  
    virtual void GetNodes(NodeList_t & Nodes) const = 0
```

Interface to access the node map.

14.6.4.21 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

[Interface](#) to access the node map.

14.6.4.22 Invalidate

```
interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0
```

14.6.4.23 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

Initial value:

```
{  
  
    virtual void SetInfo(GenICam::gcstring & Info) = 0
```

Basic interface to persist values to.

14.6.4.24 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

[Interface](#) for ports.

14.6.4.25 IPortConstruct

```
interface SPINNAKER_API IPortConstruct
```

[Interface](#) for ports.

14.6.4.26 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

[Interface](#) for recording write commands on a port.

14.6.4.27 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

Interface for replaying write commands on a port.

14.6.4.28 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

Initial value:

```
{  
    virtual void Write(const void* pBuffer, int64_t Address, int64_t Length) = 0
```

14.6.4.29 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

Initial value:

```
{  
    virtual GenICam::gcstring GetName(bool FullQualified = false) const = 0
```

Interface to construct a reference.

14.6.4.30 IRegister

```
interface SPINNAKER_API_ABSTRACT IRegister
```

Interface for registers.

14.6.4.31 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
```

Interface for groups of features selected by a single one.

14.6.4.32 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

Initial value:

```
{  
  
    virtual bool SetFirst() = 0
```

Interface of a "digit" of the "counter" formed by the selector set.

14.6.4.33 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

Interface for string properties.

14.6.4.34 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

Interface for value properties.

14.6.4.35 Length

```
interface SPINNAKER_API_ABSTRACT int64_t Length = 0
```

14.6.4.36 U3V_EVENT_PREFIX

```
const uint32_t U3V_EVENT_PREFIX = 0x45563355
```

14.6.4.37 Verify

```
interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0
```

14.7 Spinnaker::GenICam Namespace Reference

Classes

- class [AutoLock](#)
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [gcstring](#)
- class [LockableObject](#)
Instance-Lock for an object.
- struct [Version_t](#)
Version.

Functions

- [SPINNAKER_API](#) void [ThrowBadAlloc](#) ()
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str)
STL getline.
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str, char delim)
STL getline.
- template<typename Td, typename Ts >
Td [INTEGRAL_CAST2](#) (Ts s)
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- template<typename T >
T [INTEGRAL_CAST](#) (int64_t ll)
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- [SPINNAKER_API](#) bool [DoesEnvironmentVariableExist](#) (const [Spinnaker::GenICam::gcstring](#) &VariableName)
Returns true if an environment variable exists.
- [SPINNAKER_API](#) [gcstring](#) [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) bool [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName, [gcstring](#) &VariableContent)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) [gcstring](#) [UrlEncode](#) (const [gcstring](#) &Input)
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- [SPINNAKER_API](#) [gcstring](#) [UrlDecode](#) (const [gcstring](#) &Input)
Replaces xx escapes by their char equivalent.
- [SPINNAKER_API](#) void [ReplaceEnvironmentVariables](#) ([gcstring](#) &Buffer, bool ReplaceBlankBy20=false)
Replaces in a string and replace ' ' with %20.
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamCacheFolder](#) (void)
Retrieve the path of the [GenICam](#) cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamLogConfig](#) (void)

- Retrieve the path of the *GenICam* logging properties file.
- **SPINNAKER_API** `gcstring GetGenICamCLProtocolFolder` (void)
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling *SetGenICamCLProtocolFolder()*.
- **SPINNAKER_API** `void SetGenICamCacheFolder` (const `gcstring` &path)
Stores the path of the *GenICam* cache folder.
- **SPINNAKER_API** `void SetGenICamLogConfig` (const `gcstring` &path)
Stores the path of the *GenICam* logging properties file.
- **SPINNAKER_API** `void SetGenICamCLProtocolFolder` (const `gcstring` &path)
Stores the path of the CLProtocol folder.
- **SPINNAKER_API** `void Tokenize` (const `gcstring` &str, `gcstring_vector` &tokens, const `gcstring` &delimiters=" ")
splits str input string into a list of tokens using the delimiter
- **SPINNAKER_API** `void GetFiles` (const `gcstring` &FileTemplate, `gcstring_vector` &FileNames, const bool DirectoriesOnly=false)
Gets a list of files or directories matching a given FileTemplate.
- **SPINNAKER_API** `gcstring GetModulePathFromFunction` (void *pFunction)
Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

14.7.1 Function Documentation

14.7.1.1 DoesEnvironmentVariableExist()

```
SPINNAKER_API bool Spinnaker::GenICam::DoesEnvironmentVariableExist (
    const Spinnaker::GenICam::gcstring & VariableName )
```

Returns true if an environment variable exists.

14.7.1.2 GetFiles()

```
SPINNAKER_API void Spinnaker::GenICam::GetFiles (
    const gcstring & FileTemplate,
    gcstring_vector & FileNames,
    const bool DirectoriesOnly = false )
```

Gets a list of files or directories matching a given FileTemplate.

Parameters

<i>FileNames</i>	> The file template. Can contain environment variables.
<i>DirectoriesOnly</i>	> A list of files matching the file template

14.7.1.3 GetGenICamCacheFolder()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCacheFolder (
    void )
```

Retrieve the path of the [GenICam](#) cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).

If [GetGenICamCacheFolder\(\)](#) is called before [SetGenICamCacheFolder\(\)](#), it will return the value of environment variable GENICAM_CACHE_Vx_y. If this environment variable does not exist, an exception will be thrown.

14.7.1.4 GetGenICamCLProtocolFolder()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamCLProtocolFolder (
    void )
```

Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling [SetGenICamCLProtocolFolder\(\)](#).

If [GetGenICamCLProtocolFolder\(\)](#) is called before [SetGenICamCLProtocolFolder\(\)](#), it will return the value of environment variable GENICAM_CLPROTOCOL. If this environment variable does not exist, an exception will be thrown.

14.7.1.5 GetGenICamLogConfig()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetGenICamLogConfig (
    void )
```

Retrieve the path of the [GenICam](#) logging properties file.

The path to the logging properties file can be stored by calling [SetGenICamLogConfig\(\)](#). If [GetGenICamLogConfig\(\)](#) is called before [SetGenICamLogConfig\(\)](#), it will return the value of environment variable GENICAM_LOG_CONFIG_Vx_y. If this environment variable does not exist, an exception will be thrown.

14.7.1.6 getline() [1/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

STL getline.

14.7.1.7 getline() [2/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str,
    char delim ) [inline]
```

STL getline.

14.7.1.8 GetModulePathFromFunction()

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetModulePathFromFunction (
    void * pFunction )
```

Gets the full path to the module (DLL/SO) containing the given *pFunction*; empty string if not found.

true = only subdirectories (ex . and ..) are retrieved; false = only files are retrieved

14.7.1.9 GetValueOfEnvironmentVariable() [1/2]

```
SPINNAKER_API gcstring Spinnaker::GenICam::GetValueOfEnvironmentVariable (
    const gcstring & VariableName )
```

Retrieve the value of an environment variable.

Exceptions

<i>runtime_exception</i>	if not found

14.7.1.10 GetValueOfEnvironmentVariable() [2/2]

```
SPINNAKER_API bool Spinnaker::GenICam::GetValueOfEnvironmentVariable (
    const gcstring & VariableName,
    gcstring & VariableContent )
```

Retrieve the value of an environment variable.

Returns

true if environment variable was found, otherwise false

14.7.1.11 INTEGRAL_CAST()

```
T Spinnaker::GenICam::INTEGRAL_CAST (
    int64_t ll ) [inline]
```

This verifies at runtime if there was no loss of data if an `int64_t` was downcast to type `T` (e.g.

`int32_t`)

14.7.1.12 INTEGRAL_CAST2()

```
Td Spinnaker::GenICam::INTEGRAL_CAST2 (
    Ts s ) [inline]
```

This verifies at runtime if there was no loss of data if an type Ts (e.g. int64t) was downcast to type Td (e.g. int32_t)

14.7.1.13 ReplaceEnvironmentVariables()

```
SPINNAKER_API void Spinnaker::GenICam::ReplaceEnvironmentVariables (
    gcstring & Buffer,
    bool ReplaceBlankBy20 = false )
```

Replaces in a string and replace ' ' with %20.

14.7.1.14 SetGenICamCacheFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCacheFolder (
    const gcstring & path )
```

Stores the path of the [GenICam](#) cache folder.

14.7.1.15 SetGenICamCLProtocolFolder()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamCLProtocolFolder (
    const gcstring & path )
```

Stores the path of the CLProtocol folder.

14.7.1.16 SetGenICamLogConfig()

```
SPINNAKER_API void Spinnaker::GenICam::SetGenICamLogConfig (
    const gcstring & path )
```

Stores the path of the [GenICam](#) logging properties file.

14.7.1.17 ThrowBadAlloc()

```
SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc ( )
```

14.7.1.18 Tokenize()

```
SPINNAKER_API void Spinnaker::GenICam::Tokenize (
    const gcstring & str,
    gcstring_vector & tokens,
    const gcstring & delimiters = " ")
```

splits str input string into a list of tokens using the delimiter

Parameters

<i>str</i>	string to be split
<i>tokens</i>	result of the splitting operation
<i>delimiters</i>	delimiters for the splitting

14.7.1.19 UrlDecode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlDecode (
    const gcstring & Input )
```

Replaces xx escapes by their char equivalent.

14.7.1.20 UrlEncode()

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlEncode (
    const gcstring & Input )
```

Converts \ to / and replaces all unsafe characters by their xx equivalent.

14.8 Spinnaker::Video Namespace Reference**Classes**

- struct [AVIOption](#)
Options for saving AVI files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [MJPGOption](#)
Options for saving MJPG files.
- class [SpinVideo](#)
Provides the functionality for the user to record images to an AVI/MP4 file.

Chapter 15

Class Documentation

15.1 ActionCommandResult Struct Reference

Action Command Result.

Public Attributes

- unsigned int [DeviceAddress](#)
- [ActionCommandStatus](#) Status

15.1.1 Detailed Description

Action Command Result.

15.1.2 Member Data Documentation

15.1.2.1 DeviceAddress

`unsigned int DeviceAddress`

15.1.2.2 Status

`ActionCommandStatus Status`

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

- `include/SpinnakerDefs.h`

15.2 AttachStatistics_t Struct Reference

Delivers information about the attached chunks and nodes.

Public Attributes

- int [NumChunkPorts](#)
Number of chunk ports found in the node map.
- int [NumChunks](#)
Number of chunks found in the buffer.
- int [NumAttachedChunks](#)
Number of chunks from the buffer attached to a chunk port.

15.2.1 Detailed Description

Delivers information about the attached chunks and nodes.

15.2.2 Member Data Documentation

15.2.2.1 NumAttachedChunks

`int NumAttachedChunks`

Number of chunks from the buffer attached to a chunk port.

15.2.2.2 NumChunkPorts

`int NumChunkPorts`

Number of chunk ports found in the node map.

15.2.2.3 NumChunks

`int NumChunks`

Number of chunks found in the buffer.

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

- `include/SpinGenApi/ChunkAdapter.h`

15.3 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLOCK &lock\)](#)
- [~AutoLock \(\)](#)

15.3.1 Constructor & Destructor Documentation

15.3.1.1 AutoLock()

```
AutoLock (
    CLOCK & lock ) [inline]
```

15.3.1.2 ~AutoLock()

```
~AutoLock ( ) [inline]
```

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

- include/SpinGenApi/[GCSynch.h](#)

15.4 AutoLock Class Reference

Public Member Functions

- [AutoLock \(CLOCK &lock\)](#)
- [~AutoLock \(\)](#)

15.4.1 Constructor & Destructor Documentation

15.4.1.1 AutoLock()

```
AutoLock (
    CLOCK & lock ) [inline]
```

15.4.1.2 ~AutoLock()

`~AutoLock ( ) [inline]`

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

- include/SpinGenApi/[Synch.h](#)

15.5 AVIOption Struct Reference

Options for saving AVI files.

Public Member Functions

- [AVIOption \(\)](#)

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [reserved](#) [192]

15.5.1 Detailed Description

Options for saving AVI files.

15.5.2 Constructor & Destructor Documentation

15.5.2.1 AVIOption()

`AVIOption ( ) [inline]`

15.5.3 Member Data Documentation

15.5.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

15.5.3.2 height

```
unsigned int height
```

Height of source image.

15.5.3.3 reserved

```
unsigned int reserved[192]
```

15.5.3.4 width

```
unsigned int width
```

Width of source image.

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

- [include/SpinVideoDefs.h](#)

15.6 BasePtr< T, B > Class Template Reference

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Public Member Functions

- [BasePtr](#) (void) throw ()
Default constructor.
- virtual [~BasePtr](#) (void)
- [BasePtr](#) (const [BasePtr](#) &other) throw ()
Copy constructor.
- virtual [BasePtr](#) & [operator=](#) (const [BasePtr](#) &rhs)
Assign INode Pointer.
- virtual [BasePtr](#) & [operator=](#) (const int nMustBeNull)
- virtual [BasePtr](#) & [operator=](#) (const long nMustBeNull)
- virtual [BasePtr](#) & [operator=](#) (const std::nullptr_t nullptr)
- virtual [operator T*](#) (void) const
Dereferencing.
- virtual T * [operator->](#) (void) const
Dereferencing.
- virtual bool [IsValid](#) () const throw ()
True if the pointer is valid.
- virtual [operator bool](#) (void) const throw ()
True if the pointer is valid.
- virtual bool [operator==](#) (const [BasePtr](#) &rT) const
Pointer equal.
- virtual bool [operator==](#) (std::nullptr_t) const
Pointer equal.
- virtual bool [operator==](#) (int nMustBeNull) const
Pointer equal.
- virtual bool [operator==](#) (long nMustBeNull) const
Pointer equal.
- virtual bool [operator!=](#) (const [BasePtr](#) &rT) const
Pointer not equal.
- virtual bool [operator!=](#) (std::nullptr_t) const
Pointer not equal.
- virtual bool [operator!=](#) (int nMustBeNull) const
Pointer not equal.
- virtual bool [operator!=](#) (long nMustBeNull) const
Pointer not equal.
- virtual T * [get](#) () const
[get\(\)](#)

Protected Attributes

- PointerData * [m_pT](#)
Underlying raw pointer.

Friends

- class [CameraListImpl](#)
- class [InterfaceListImpl](#)

15.6.1 Detailed Description

```
template<class T, class B = T>
class Spinnaker::BasePtr< T, B >
```

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

15.6.2 Constructor & Destructor Documentation

15.6.2.1 BasePtr() [1/2]

```
BasePtr (
    void ) throw ( )
```

Default constructor.

15.6.2.2 ~BasePtr()

```
virtual ~BasePtr (
    void ) [virtual]
```

15.6.2.3 BasePtr() [2/2]

```
BasePtr (
    const BasePtr< T, B > & other ) throw ( )
```

Copy constructor.

15.6.3 Member Function Documentation

15.6.3.1 get()

```
virtual T* get ( ) const [virtual]
```

[get\(\)](#)

15.6.3.2 IsValid()

```
virtual bool IsValid ( ) const throw ( )    [virtual]
```

True if the pointer is valid.

15.6.3.3 operator bool()

```
virtual operator bool (
    void ) const throw ( )    [virtual]
```

True if the pointer is valid.

15.6.3.4 operator T*()

```
virtual operator T* (
    void ) const    [virtual]
```

Dereferencing.

15.6.3.5 operator"!=()" [1/4]

```
virtual bool operator!= (
    const BasePtr< T, B > & rT ) const    [virtual]
```

Pointer not equal.

15.6.3.6 operator"!=()" [2/4]

```
virtual bool operator!= (
    int nMustBeNull ) const    [virtual]
```

Pointer not equal.

15.6.3.7 operator"!=()" [3/4]

```
virtual bool operator!= (
    long nMustBeNull ) const    [virtual]
```

Pointer not equal.

15.6.3.8 operator!=(()) [4/4]

```
virtual bool operator!=(  
    std::nullptr_t ) const [virtual]
```

Pointer not equal.

15.6.3.9 operator->()

```
virtual T* operator->(  
    void ) const [virtual]
```

Dereferencing.

15.6.3.10 operator=() [1/4]

```
virtual BasePtr& operator=(  
    const BasePtr< T, B > & rhs ) [virtual]
```

Assign INode Pointer.

15.6.3.11 operator=() [2/4]

```
virtual BasePtr& operator=(  
    const int nMustBeNull ) [virtual]
```

15.6.3.12 operator=() [3/4]

```
virtual BasePtr& operator=(  
    const long nMustBeNull ) [virtual]
```

15.6.3.13 operator=() [4/4]

```
virtual BasePtr& operator=(  
    const std::nullptr_t nullptr ) [virtual]
```

15.6.3.14 operator==() [1/4]

```
virtual bool operator==(
    const BasePtr< T, B > & rT ) const [virtual]
```

Pointer equal.

15.6.3.15 operator==() [2/4]

```
virtual bool operator==(
    int nMustBeNull ) const [virtual]
```

Pointer equal.

15.6.3.16 operator==() [3/4]

```
virtual bool operator==(
    long nMustBeNull ) const [virtual]
```

Pointer equal.

15.6.3.17 operator==() [4/4]

```
virtual bool operator==(
    std::nullptr_t ) const [virtual]
```

Pointer equal.

15.6.4 Friends And Related Function Documentation**15.6.4.1 CameraListImpl**

```
friend class CameraListImpl [friend]
```

15.6.4.2 InterfaceListImpl

```
friend class InterfaceListImpl [friend]
```

15.6.5 Member Data Documentation

15.6.5.1 m_pT

PointerData* m_pT [protected]

Underlying raw pointer.

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

- include/[BasePtr.h](#)

15.7 BMPOption Struct Reference

Options for saving Bitmap image.

Public Member Functions

- [BMPOption](#) ()

Public Attributes

- bool [indexedColor_8bit](#)
- unsigned int [reserved](#) [16]

Reserved for future use.

15.7.1 Detailed Description

Options for saving Bitmap image.

15.7.2 Constructor & Destructor Documentation

15.7.2.1 BMPOption()

[BMPOption](#) () [inline]

15.7.3 Member Data Documentation

15.7.3.1 indexedColor_8bit

```
bool indexedColor_8bit
```

15.7.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

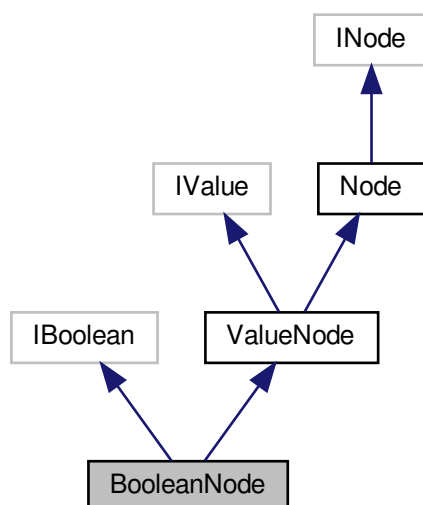
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

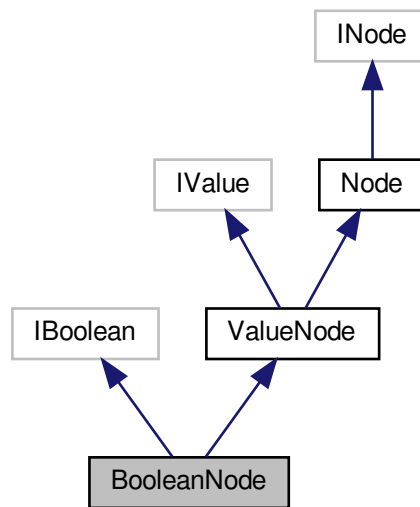
15.8 BooleanNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for BooleanNode:



Collaboration diagram for BooleanNode:



Public Member Functions

- [BooleanNode](#) ()
- [BooleanNode](#) (std::shared_ptr< Node::NodeImpl > pBoolean)
- virtual [~BooleanNode](#) ()
- void [SetValue](#) (bool Value, bool [Verify](#)=true)
Set node value.
- virtual void [operator=](#) (bool Value)
Set node value.
- bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const
Get node value.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.8.1 Detailed Description

[Interface](#) for string properties.

15.8.2 Constructor & Destructor Documentation

15.8.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

15.8.2.2 BooleanNode() [2/2]

```
BooleanNode (
    std::shared_ptr< Node::NodeImpl > pBoolean )
```

15.8.2.3 ~BooleanNode()

```
virtual ~BooleanNode ( ) [virtual]
```

15.8.3 Member Function Documentation

15.8.3.1 GetValue()

```
bool GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) const
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked.
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false).

Returns

The value read.

15.8.3.2 operator=()

```
virtual void operator= (
    bool Value ) [virtual]
```

Set node value.

15.8.3.3 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

15.8.3.4 SetValue()

```
void SetValue (
    bool Value,
    bool Verify = true )
```

Set node value.

Parameters

<i>Value</i>	The value to set.
<i>Verify</i>	Enables AccessMode and Range verification (default = true).

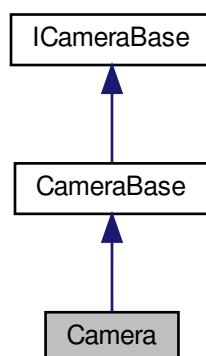
The documentation for this class was generated from the following file:

- include/SpinGenApi/[BooleanNode.h](#)

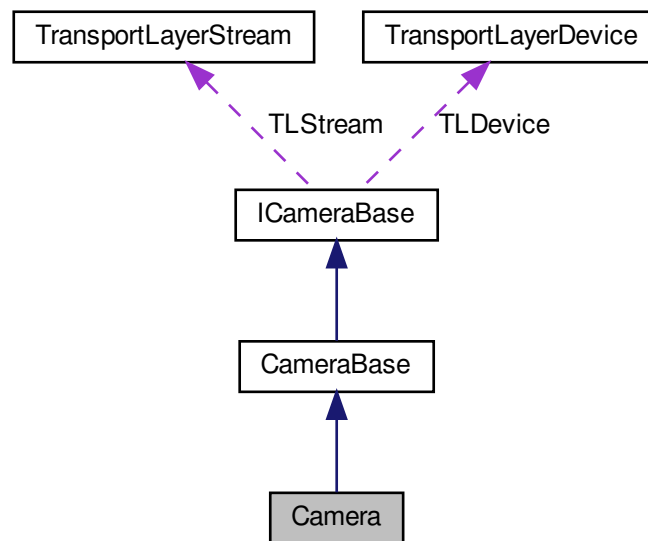
15.9 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- [~Camera \(\)](#)
- void [Init \(\)](#)

Public Attributes

- [GenApi::Integer](#) & [LUTIndex](#)
- [GenApi::Boolean](#) & [LUTEnable](#)
- [GenApi::Integer](#) & [LUTValue](#)
- [GenApi::EnumerationT< LUTSelectorEnums >](#) & [LUTSelector](#)
- [GenApi::Float](#) & [ExposureTime](#)
- [GenApi::Command](#) & [AcquisitionStop](#)
Description: This command stops the acquisition of images.
- [GenApi::Float](#) & [AcquisitionResultingFrameRate](#)
Description: Resulting frame rate in Hertz.
- [GenApi::Float](#) & [AcquisitionLineRate](#)
Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.
- [GenApi::Command](#) & [AcquisitionStart](#)
Description: This command starts the acquisition of images.
- [GenApi::Command](#) & [TriggerSoftware](#)
- [GenApi::EnumerationT< ExposureModeEnums >](#) & [ExposureMode](#)
- [GenApi::EnumerationT< AcquisitionModeEnums >](#) & [AcquisitionMode](#)
Description: Sets the acquisition mode of the device.
- [GenApi::Integer](#) & [AcquisitionFrameCount](#)
- [GenApi::EnumerationT< TriggerSourceEnums >](#) & [TriggerSource](#)

- [GenApi::IEnumerationT< TriggerActivationEnums > & TriggerActivation](#)
Description: Specifies the activation mode of the trigger.
- [GenApi::IEnumerationT< SensorShutterModeEnums > & SensorShutterMode](#)
Description: Sets the shutter mode of the device.
- [GenApi::IFloat & TriggerDelay](#)
- [GenApi::IEnumerationT< TriggerModeEnums > & TriggerMode](#)
- [GenApi::IFloat & AcquisitionFrameRate](#)
Description: User controlled acquisition frame rate in Hertz Visibility:
- [GenApi::IEnumerationT< TriggerOverlapEnums > & TriggerOverlap](#)
Description: Specifies the overlap mode of the trigger.
- [GenApi::IEnumerationT< TriggerSelectorEnums > & TriggerSelector](#)
Description: Selects the type of trigger to configure.
- [GenApi::IBoolean & AcquisitionFrameRateEnable](#)
Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.
- [GenApi::IEnumerationT< ExposureAutoEnums > & ExposureAuto](#)
Description: Sets the automatic exposure mode Visibility:
- [GenApi::Integer & AcquisitionBurstFrameCount](#)
- [GenApi::Integer & EventTest](#)
Description: Returns the unique identifier of the Test type of Event.
- [GenApi::Integer & EventTestTimestamp](#)
Description: Returns the Timestamp of the Test Event.
- [GenApi::Integer & EventExposureEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.
- [GenApi::Integer & EventExposureEnd](#)
Description: Returns the unique identifier of the Exposure End type of Event.
- [GenApi::Integer & EventExposureEndTimestamp](#)
Description: Returns the Timestamp of the Exposure End Event.
- [GenApi::Integer & EventError](#)
Description: Returns the unique identifier of the Error type of Event.
- [GenApi::Integer & EventErrorTimestamp](#)
Description: Returns the Timestamp of the Error Event.
- [GenApi::Integer & EventErrorCode](#)
Description: Returns the error code for the error that happened Visibility:
- [GenApi::Integer & EventErrorFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.
- [GenApi::IEnumerationT< EventSelectorEnums > & EventSelector](#)
Description: Selects which Event to enable or disable.
- [GenApi::IBoolean & EventSerialReceiveOverflow](#)
Description: Returns the status of the event serial receive overflow.
- [GenApi::Integer & EventSerialPortReceive](#)
Description: Returns the unique identifier of the Serial Port Receive type of Event.
- [GenApi::Integer & EventSerialPortReceiveTimestamp](#)
Description: Returns the Timestamp of the Serial Port Receive Event.
- [GenApi::IString & EventSerialData](#)
Description: Returns the serial data that was received.
- [GenApi::Integer & EventSerialDataLength](#)
Description: Returns the length of the received serial data that was included in the event payload.
- [GenApi::IEnumerationT< EventNotificationEnums > & EventNotification](#)
Description: Enables/Disables the selected event.
- [GenApi::Integer & LogicBlockLUTRowIndex](#)
Description: Controls the row of the truth table to access in the selected LUT.

- [GenApi::IEnumerationT< LogicBlockSelectorEnums > & LogicBlockSelector](#)
Description: Selects which LogicBlock to configure Visibility:
- [GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums > & LogicBlockLUTInputActivation](#)
Description: Selects the activation mode of the Logic Input Source signal.
- [GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums > & LogicBlockLUTInputSelector](#)
Description: Controls which LogicBlockLUT Input Source & Activation to access.
- [GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums > & LogicBlockLUTInputSource](#)
Description: Selects the source for the input into the Logic LUT.
- [GenApi::IBoolean & LogicBlockLUTOutputValue](#)
Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.
- [GenApi::IInteger & LogicBlockLUTOutputValueAll](#)
Description: Sets the value of all the output bits in the selected LUT.
- [GenApi::IEnumerationT< LogicBlockLUTSelectorEnums > & LogicBlockLUTSelector](#)
Description: Selects which LogicBlock LUT to configure Visibility:
- [GenApi::IFloat & ColorTransformationValue](#)
- [GenApi::IBoolean & ColorTransformationEnable](#)
- [GenApi::IEnumerationT< ColorTransformationSelectorEnums > & ColorTransformationSelector](#)
Description: Selects which Color Transformation module is controlled by the various Color Transformation features.
- [GenApi::IEnumerationT< RgbTransformLightSourceEnums > & RgbTransformLightSource](#)
- [GenApi::IFloat & Saturation](#)
Description: Controls the color saturation.
- [GenApi::IBoolean & SaturationEnable](#)
Description: Enables/disables Saturation adjustment.
- [GenApi::IEnumerationT< ColorTransformationValueSelectorEnums > & ColorTransformationValueSelector](#)
- [GenApi::IInteger & TimestampLatchValue](#)
Description: Returns the latched value of the timestamp counter.
- [GenApi::ICommand & TimestampReset](#)
Description: Resets the current value of the device timestamp counter.
- [GenApi::IString & DeviceUserID](#)
Description: User-programmable device identifier.
- [GenApi::IFloat & DeviceTemperature](#)
Description: Device temperature in degrees Celsius (C).
- [GenApi::IInteger & MaxDeviceResetTime](#)
Description: Time to wait until device reset complete (ms).
- [GenApi::IInteger & DeviceTLVersionMinor](#)
- [GenApi::IString & DeviceSerialNumber](#)
- [GenApi::IString & DeviceVendorName](#)
Description: Name of the manufacturer of the device.
- [GenApi::IEnumerationT< DeviceRegistersEndiannessEnums > & DeviceRegistersEndianness](#)
Description: Endianness of the registers of the device.
- [GenApi::IString & DeviceManufacturerInfo](#)
Description: Manufacturer information about the device.
- [GenApi::IInteger & DeviceLinkSpeed](#)
- [GenApi::IInteger & LinkUptime](#)
Description: Time since the last phy negotiation (enumeration).
- [GenApi::IInteger & DeviceEventChannelCount](#)
- [GenApi::ICommand & TimestampLatch](#)
Description: Latches the current timestamp counter into TimestampLatchValue.
- [GenApi::IEnumerationT< DeviceScanTypeEnums > & DeviceScanType](#)
Description: Scan type of the sensor of the device.
- [GenApi::ICommand & DeviceReset](#)

Description: This is a command that immediately resets and reboots the device.

- [GenApi::IEnumerationT< DeviceCharacterSetEnums > & DeviceCharacterSet](#)
- [GenApi::Integer & DeviceLinkThroughputLimit](#)
- [GenApi::IString & DeviceFirmwareVersion](#)

Description: Version of the firmware on the device.

- [GenApi::Integer & DeviceStreamChannelCount](#)
- [GenApi::IEnumerationT< DeviceTLTypeEnums > & DeviceTLType](#)

Description: Transport Layer type of the device.

- [GenApi::IString & DeviceVersion](#)

Description: Version of the device.

- [GenApi::IEnumerationT< DevicePowerSupplySelectorEnums > & DevicePowerSupplySelector](#)
- [GenApi::IString & SensorDescription](#)

Description: Returns Sensor Description Visibility:

- [GenApi::IString & DeviceModelName](#)

Description: Model of the device.

- [GenApi::Integer & DeviceTLVersionMajor](#)
- [GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > & DeviceTemperatureSelector](#)
- [GenApi::Integer & EnumerationCount](#)

Description: Number of enumerations since uptime.

- [GenApi::IFloat & PowerSupplyCurrent](#)
- [GenApi::IString & DeviceID](#)

Description: Device identifier (serial number).

- [GenApi::Integer & DeviceUptime](#)

Description: Total time since the device was powered up in seconds.

- [GenApi::Integer & DeviceLinkCurrentThroughput](#)

Description: Current bandwidth of streamed data.

- [GenApi::Integer & DeviceMaxThroughput](#)
- [GenApi::ICommand & FactoryReset](#)

Description: Returns all user tables to factory default Visibility:

- [GenApi::IFloat & PowerSupplyVoltage](#)
- [GenApi::IEnumerationT< DeviceIndicatorModeEnums > & DeviceIndicatorMode](#)

Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

- [GenApi::IFloat & DeviceLinkBandwidthReserve](#)
- [GenApi::Integer & AasRoiOffsetY](#)
- [GenApi::Integer & AasRoiOffsetX](#)
- [GenApi::IEnumerationT< AutoExposureControlPriorityEnums > & AutoExposureControlPriority](#)
- [GenApi::IFloat & BalanceWhiteAutoLowerLimit](#)
- [GenApi::IFloat & BalanceWhiteAutoDamping](#)
- [GenApi::Integer & AasRoiHeight](#)
- [GenApi::IFloat & AutoExposureGreyValueUpperLimit](#)
- [GenApi::IFloat & AutoExposureTargetGreyValue](#)
- [GenApi::IFloat & AutoExposureGainLowerLimit](#)
- [GenApi::IFloat & AutoExposureGreyValueLowerLimit](#)
- [GenApi::IEnumerationT< AutoExposureMeteringModeEnums > & AutoExposureMeteringMode](#)
- [GenApi::IFloat & AutoExposureExposureTimeUpperLimit](#)
- [GenApi::IFloat & AutoExposureGainUpperLimit](#)
- [GenApi::IFloat & AutoExposureControlLoopDamping](#)
- [GenApi::IFloat & AutoExposureEVCompensation](#)
- [GenApi::IFloat & AutoExposureExposureTimeLowerLimit](#)
- [GenApi::IEnumerationT< BalanceWhiteAutoProfileEnums > & BalanceWhiteAutoProfile](#)

Description: Selects the profile used by BalanceWhiteAuto.

- [GenApi::IEnumerationT< AutoAlgorithmSelectorEnums > & AutoAlgorithmSelector](#)

- [GenApi::IEnumerationT< AutoExposureTargetGreyValueAutoEnums > & AutoExposureTargetGreyValueAuto](#)
- [GenApi::IBoolean & AasRoiEnable](#)
- [GenApi::IEnumerationT< AutoExposureLightingModeEnums > & AutoExposureLightingMode](#)
- [GenApi::IInteger & AasRoiWidth](#)
- [GenApi::IFloat & BalanceWhiteAutoUpperLimit](#)
- [GenApi::IInteger & LinkErrorCount](#)
Description: Counts the number of error on the link.
- [GenApi::IBoolean & GevCurrentIPConfigurationDHCP](#)
Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.
- [GenApi::IInteger & GevInterfaceSelector](#)
Description: Selects which logical link to control.
- [GenApi::IInteger & GevSCPD](#)
Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.
- [GenApi::IInteger & GevTimestampTickFrequency](#)
Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).
- [GenApi::IInteger & GevSCPSPacketSize](#)
Description: Specifies the stream packet size (in bytes) to send on this channel.
- [GenApi::IInteger & GevCurrentDefaultGateway](#)
Description: Reports the default gateway IP address to be used on the given logical link.
- [GenApi::IBoolean & GevSCCFGUnconditionalStreaming](#)
Description: Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).
- [GenApi::IInteger & GevMCTT](#)
Description: Indicates the transmission timeout of the message channel.
- [GenApi::IBoolean & GevSCPSPDoNotFragment](#)
Description: The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.
- [GenApi::IInteger & GevCurrentSubnetMask](#)
Description: Reports the subnet mask of the given logical link.
- [GenApi::IInteger & GevStreamChannelSelector](#)
Description: Selects the stream channel to control.
- [GenApi::IInteger & GevCurrentIPAddress](#)
Description: Reports the IP address for the given logical link.
- [GenApi::IInteger & GevMCSP](#)
Description: Indicates the source port of the message channel.
- [GenApi::IInteger & GevGVCPPendingTimeout](#)
Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.
- [GenApi::IEnumerationT< GevIEEE1588StatusEnums > & GevIEEE1588Status](#)
Description: Provides the status of the IEEE 1588 clock.
- [GenApi::IString & GevFirstURL](#)
Description: The first choice of URL for the XML device description file.
- [GenApi::IInteger & GevMACAddress](#)
Description: MAC address of the logical link.
- [GenApi::IInteger & GevPersistentSubnetMask](#)
Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.
- [GenApi::IInteger & GevMCPHostPort](#)
Description: The port to which the device must send messages Visibility:
- [GenApi::IInteger & GevSCPHostPort](#)
Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
- [GenApi::IBoolean & GevGVCPPendingAck](#)
Description: Enables the generation of PENDING_ACK.

- [GenApi::Integer](#) & [GevSCPIInterfaceIndex](#)
Description: Index of the logical link to use.
- [GenApi::IBoolean](#) & [GevSupportedOption](#)
Description: Returns if the selected GEV option is supported.
- [GenApi::IEnumerationT](#) < [GevIEEE1588ModeEnums](#) > & [GevIEEE1588Mode](#)
Description: Provides the mode of the IEEE 1588 clock.
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationLLA](#)
Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.
- [GenApi::Integer](#) & [GevSCSP](#)
Description: Indicates the source port of the stream channel.
- [GenApi::IBoolean](#) & [GevIEEE1588](#)
Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.
- [GenApi::IBoolean](#) & [GevSCCFGExtendedChunkData](#)
Description: Enables cameras to use the extended chunk data payload type for this stream channel.
- [GenApi::Integer](#) & [GevPersistentIPAddress](#)
Description: Controls the Persistent IP address for this logical link.
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationPersistentIP](#)
Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.
- [GenApi::IEnumerationT](#) < [GevIEEE1588ClockAccuracyEnums](#) > & [GevIEEE1588ClockAccuracy](#)
Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.
- [GenApi::Integer](#) & [GevHeartbeatTimeout](#)
Description: Indicates the current heartbeat timeout in milliseconds.
- [GenApi::Integer](#) & [GevPersistentDefaultGateway](#)
Description: Controls the persistent default gateway for this logical link.
- [GenApi::IEnumerationT](#) < [GevCCPEnums](#) > & [GevCCP](#)
Description: Controls the device access privilege of an application.
- [GenApi::Integer](#) & [GevMCDA](#)
Description: Controls the destination IP address of the message channel Visibility:
- [GenApi::Integer](#) & [GevSCDA](#)
Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.
- [GenApi::Integer](#) & [GevSCPDirection](#)
Description: Transmit or Receive of the channel Visibility:
- [GenApi::IBoolean](#) & [GevSCPSFireTestPacket](#)
Description: Sends a test packet.
- [GenApi::IString](#) & [GevSecondURL](#)
Description: The second choice of URL to the XML device description file.
- [GenApi::IEnumerationT](#) < [GevSupportedOptionSelectorEnums](#) > & [GevSupportedOptionSelector](#)
Description: Selects the GEV option to interrogate for existing support.
- [GenApi::IBoolean](#) & [GevGVCPHeartbeatDisable](#)
Description: Disables the GVCP heartbeat.
- [GenApi::Integer](#) & [GevMCRC](#)
Description: Indicates the number of retries of the message channel.
- [GenApi::IBoolean](#) & [GevSCPSBigEndian](#)
Description: Endianness of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [GevNumberOfInterfaces](#)
Description: Indicates the number of physical network interfaces supported by this device.
- [GenApi::Integer](#) & [TLPParamsLocked](#)
Description: Visibility:
- [GenApi::Integer](#) & [PayloadSize](#)

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

- [GenApi::Integer](#) & [PacketResendRequestCount](#)

Description: Counts the number of resend requests received from the host.

- [GenApi::Boolean](#) & [SharpeningEnable](#)
- [GenApi::EnumerationT<BlackLevelSelectorEnums>](#) & [BlackLevelSelector](#)
- [GenApi::Boolean](#) & [GammaEnable](#)

Description: Enables/disables gamma correction.

- [GenApi::Boolean](#) & [SharpeningAuto](#)
- [GenApi::Boolean](#) & [BlackLevelClampingEnable](#)
- [GenApi::Float](#) & [BalanceRatio](#)
- [GenApi::EnumerationT<BalanceWhiteAutoEnums>](#) & [BalanceWhiteAuto](#)
- [GenApi::Float](#) & [SharpeningThreshold](#)
- [GenApi::EnumerationT<GainAutoEnums>](#) & [GainAuto](#)
- [GenApi::Float](#) & [Sharpening](#)
- [GenApi::Float](#) & [Gain](#)
- [GenApi::EnumerationT<BalanceRatioSelectorEnums>](#) & [BalanceRatioSelector](#)
- [GenApi::EnumerationT<GainSelectorEnums>](#) & [GainSelector](#)

Description: Selects which gain to control.

- [GenApi::Float](#) & [BlackLevel](#)
- [GenApi::Integer](#) & [BlackLevelRaw](#)
- [GenApi::Float](#) & [Gamma](#)

Description: Controls the gamma correction of pixel intensity.

- [GenApi::Integer](#) & [DefectTableIndex](#)
- [GenApi::Command](#) & [DefectTableFactoryRestore](#)

Description: Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.

- [GenApi::Integer](#) & [DefectTableCoordinateY](#)
- [GenApi::Command](#) & [DefectTableSave](#)

Description: Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.

- [GenApi::EnumerationT<DefectCorrectionModeEnums>](#) & [DefectCorrectionMode](#)

Description: Controls the method used for replacing defective pixels.

- [GenApi::Integer](#) & [DefectTableCoordinateX](#)
- [GenApi::Integer](#) & [DefectTablePixelCount](#)
- [GenApi::Boolean](#) & [DefectCorrectStaticEnable](#)

Description: Enables/Disables table-based defective pixel correction.

- [GenApi::Command](#) & [DefectTableApply](#)

Description: Applies the current defect table, so that any changes made affect images captured by the camera.

- [GenApi::Boolean](#) & [UserSetFeatureEnable](#)

Description: Whether or not the selected feature is saved to user sets.

- [GenApi::Command](#) & [UserSetSave](#)
- [GenApi::EnumerationT<UserSetSelectorEnums>](#) & [UserSetSelector](#)
- [GenApi::Command](#) & [UserSetLoad](#)
- [GenApi::EnumerationT<UserSetDefaultEnums>](#) & [UserSetDefault](#)
- [GenApi::EnumerationT<SerialPortBaudRateEnums>](#) & [SerialPortBaudRate](#)

Description: This feature controls the baud rate used by the selected serial port.

- [GenApi::Integer](#) & [SerialPortDataBits](#)

Description: This feature controls the number of data bits used by the selected serial port.

- [GenApi::EnumerationT<SerialPortParityEnums>](#) & [SerialPortParity](#)

Description: This feature controls the parity used by the selected serial port.

- [GenApi::Integer](#) & [SerialTransmitQueueMaxCharacterCount](#)

Description: >Returns the maximum number of characters in the serial port transmit queue.

- [GenApi::Integer](#) & [SerialReceiveQueueCurrentCharacterCount](#)

- Description: Returns the number of characters currently in the serial port receive queue.*

 - [GenApi::IEnumerationT< SerialPortSelectorEnums > & SerialPortSelector](#)

Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< SerialPortStopBitsEnums > & SerialPortStopBits](#)

Description: This feature controls the number of stop bits used by the selected serial port.
- [GenApi::ICommand & SerialReceiveQueueClear](#)

Description: This is a command that clears the device serial port receive queue.
- [GenApi::Integer & SerialReceiveFramingErrorCount](#)

Description: Returns the number of framing errors that have occurred on the serial port.
- [GenApi::Integer & SerialTransmitQueueCurrentCharacterCount](#)

Description: Returns the number of characters currently in the serial port transmit queue.
- [GenApi::Integer & SerialReceiveParityErrorCount](#)

Description: Returns the number of parity errors that have occurred on the serial port.
- [GenApi::IEnumerationT< SerialPortSourceEnums > & SerialPortSource](#)

Description: Specifies the physical input Line on which to receive serial data.
- [GenApi::Integer & SerialReceiveQueueMaxCharacterCount](#)

Description: >Returns the maximum number of characters in the serial port receive queue.
- [GenApi::Integer & SequencerSetStart](#)

Description: Sets the first sequencer set to be used.
- [GenApi::IEnumerationT< SequencerModeEnums > & SequencerMode](#)

Description: Controls whether or not a sequencer is active.
- [GenApi::IEnumerationT< SequencerConfigurationValidEnums > & SequencerConfigurationValid](#)
- [GenApi::IEnumerationT< SequencerSetValidEnums > & SequencerSetValid](#)
- [GenApi::Integer & SequencerSetSelector](#)
- [GenApi::IEnumerationT< SequencerTriggerActivationEnums > & SequencerTriggerActivation](#)
- [GenApi::IEnumerationT< SequencerConfigurationModeEnums > & SequencerConfigurationMode](#)
- [GenApi::ICommand & SequencerSetSave](#)
- [GenApi::IEnumerationT< SequencerTriggerSourceEnums > & SequencerTriggerSource](#)
- [GenApi::Integer & SequencerSetActive](#)

Description: Displays the currently active sequencer set.
- [GenApi::Integer & SequencerSetNext](#)

Description: Specifies the next sequencer set.
- [GenApi::ICommand & SequencerSetLoad](#)
- [GenApi::Integer & SequencerPathSelector](#)
- [GenApi::IBoolean & SequencerFeatureEnable](#)
- [GenApi::Integer & TransferBlockCount](#)

Description: Specifies the number of data blocks (images) that the device should stream before stopping.
- [GenApi::ICommand & TransferStart](#)

Description: Starts the streaming of data blocks (images) out of the device.
- [GenApi::Integer & TransferQueueMaxBlockCount](#)

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:
- [GenApi::Integer & TransferQueueCurrentBlockCount](#)

Description: Returns number of data blocks (images) currently in the transfer queue.
- [GenApi::IEnumerationT< TransferQueueModeEnums > & TransferQueueMode](#)

Description: Specifies the operation mode of the transfer queue.
- [GenApi::IEnumerationT< TransferOperationModeEnums > & TransferOperationMode](#)

Description: Selects the operation mode of the transfer.
- [GenApi::ICommand & TransferStop](#)

Description: Stops the streaming of data block (images).
- [GenApi::Integer & TransferQueueOverflowCount](#)

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

- [GenApi::IEnumerationT< TransferControlModeEnums > & TransferControlMode](#)
Description: Selects the control method for the transfers.
- [GenApi::IFloat & ChunkBlackLevel](#)
Description: Returns the black level used to capture the image.
- [GenApi::Integer & ChunkFrameID](#)
Description: Returns the image frame ID.
- [GenApi::IString & ChunkSerialData](#)
Description: Returns the serial data that was received.
- [GenApi::IFloat & ChunkExposureTime](#)
Description: Returns the exposure time used to capture the image.
- [GenApi::Integer & ChunkCompressionMode](#)
Description: Returns the compression mode of the last image payload.
- [GenApi::IFloat & ChunkCompressionRatio](#)
Description: Returns the compression ratio of the last image payload.
- [GenApi::IBoolean & ChunkSerialReceiveOverflow](#)
Description: Returns the status of the chunk serial receive overflow.
- [GenApi::Integer & ChunkTimestamp](#)
Description: Returns the Timestamp of the image.
- [GenApi::IBoolean & ChunkModeActive](#)
Description: Activates the inclusion of Chunk data in the payload of the image.
- [GenApi::Integer & ChunkExposureEndLineStatusAll](#)
Description: Returns the status of all the I/O lines at the end of exposure event.
- [GenApi::IEnumerationT< ChunkGainSelectorEnums > & ChunkGainSelector](#)
Description: Selects which gain to retrieve Visibility:
- [GenApi::IEnumerationT< ChunkSelectorEnums > & ChunkSelector](#)
Description: Selects which chunk data to enable or disable.
- [GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > & ChunkBlackLevelSelector](#)
Description: Selects which black level to retrieve Visibility:
- [GenApi::Integer & ChunkWidth](#)
Description: Returns the width of the image included in the payload.
- [GenApi::Integer & ChunkImage](#)
Description: Returns the image payload.
- [GenApi::Integer & ChunkHeight](#)
Description: Returns the height of the image included in the payload.
- [GenApi::IEnumerationT< ChunkPixelFormatEnums > & ChunkPixelFormat](#)
Description: Format of the pixel provided by the camera Visibility:
- [GenApi::IFloat & ChunkGain](#)
Description: Returns the gain used to capture the image.
- [GenApi::Integer & ChunkSequencerSetActive](#)
Description: Returns the index of the active set of the running sequencer included in the payload.
- [GenApi::Integer & ChunkCRC](#)
Description: Returns the CRC of the image payload.
- [GenApi::Integer & ChunkOffsetX](#)
Description: Returns the Offset X of the image included in the payload.
- [GenApi::Integer & ChunkOffsetY](#)
Description: Returns the Offset Y of the image included in the payload.
- [GenApi::IBoolean & ChunkEnable](#)
Description: Enables the inclusion of the selected Chunk data in the payload of the image.
- [GenApi::Integer & ChunkSerialDataLength](#)
Description: Returns the length of the received serial data that was included in the payload.
- [GenApi::Integer & FileAccessOffset](#)

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

- [GenApi::Integer](#) & [FileAccessLength](#)

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

- [GenApi::EnumerationT](#) < [FileOperationStatusEnums](#) > & [FileOperationStatus](#)

Description: Represents the file operation execution status.

- [GenApi::ICommand](#) & [FileOperationExecute](#)
- [GenApi::EnumerationT](#) < [FileOpenModeEnums](#) > & [FileOpenMode](#)
- [GenApi::Integer](#) & [FileOperationResult](#)

Description: Represents the file operation result.

- [GenApi::EnumerationT](#) < [FileOperationSelectorEnums](#) > & [FileOperationSelector](#)
- [GenApi::EnumerationT](#) < [FileSelectorEnums](#) > & [FileSelector](#)
- [GenApi::Integer](#) & [FileSize](#)

Description: Represents the size of the selected file in bytes.

- [GenApi::EnumerationT](#) < [BinningSelectorEnums](#) > & [BinningSelector](#)
- [GenApi::Integer](#) & [PixelDynamicRangeMin](#)

Description: Minimum value that can be returned during the digitization process.

- [GenApi::Integer](#) & [PixelDynamicRangeMax](#)

Description: Maximum value that can be returned during the digitization process.

- [GenApi::Integer](#) & [OffsetY](#)
- [GenApi::Integer](#) & [BinningHorizontal](#)
- [GenApi::Integer](#) & [Width](#)
- [GenApi::EnumerationT](#) < [TestPatternGeneratorSelectorEnums](#) > & [TestPatternGeneratorSelector](#)
- [GenApi::Float](#) & [CompressionRatio](#)

Description: Reports the ratio between the uncompressed image size and compressed image size.

- [GenApi::EnumerationT](#) < [CompressionSaturationPriorityEnums](#) > & [CompressionSaturationPriority](#)

Description: When FrameRate is enabled, camera drops frames if datarate is saturated.

- [GenApi::Boolean](#) & [ReverseX](#)

Description: Horizontally flips the image sent by the device.

- [GenApi::Boolean](#) & [ReverseY](#)

Description: Vertically flips the image sent by the device.

- [GenApi::EnumerationT](#) < [TestPatternEnums](#) > & [TestPattern](#)
- [GenApi::EnumerationT](#) < [PixelColorFilterEnums](#) > & [PixelColorFilter](#)

Description: Type of color filter that is applied to the image.

- [GenApi::Integer](#) & [WidthMax](#)
- [GenApi::EnumerationT](#) < [AdcBitDepthEnums](#) > & [AdcBitDepth](#)
- [GenApi::Integer](#) & [BinningVertical](#)
- [GenApi::EnumerationT](#) < [DecimationHorizontalModeEnums](#) > & [DecimationHorizontalMode](#)
- [GenApi::EnumerationT](#) < [BinningVerticalModeEnums](#) > & [BinningVerticalMode](#)

Description: Visibility:

- [GenApi::Integer](#) & [OffsetX](#)
- [GenApi::Integer](#) & [HeightMax](#)
- [GenApi::Integer](#) & [DecimationHorizontal](#)
- [GenApi::EnumerationT](#) < [PixelSizeEnums](#) > & [PixelSize](#)

Description: Total size in bits of a pixel of the image.

- [GenApi::Integer](#) & [SensorHeight](#)

Description: Effective height of the sensor in pixels.

- [GenApi::EnumerationT](#) < [DecimationSelectorEnums](#) > & [DecimationSelector](#)

Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

- [GenApi::Boolean](#) & [IspEnable](#)
- [GenApi::Boolean](#) & [AdaptiveCompressionEnable](#)

Description: Controls whether lossless compression adapts to the image content.

- [GenApi::IEnumerationT< ImageCompressionModeEnums > & ImageCompressionMode](#)
Description: Visibility:
- [GenApi::Integer & DecimationVertical](#)
- [GenApi::Integer & Height](#)
- [GenApi::IEnumerationT< BinningHorizontalModeEnums > & BinningHorizontalMode](#)
Description: Visibility:
- [GenApi::IEnumerationT< PixelFormatEnums > & PixelFormat](#)
Description: Format of the pixel provided by the camera.
- [GenApi::Integer & SensorWidth](#)
Description: Effective width of the sensor in pixels.
- [GenApi::IEnumerationT< DecimationVerticalModeEnums > & DecimationVerticalMode](#)
- [GenApi::ICommand & TestEventGenerate](#)
Description: This command generates a test event and sends it to the host.
- [GenApi::ICommand & TriggerEventTest](#)
Description: This command generates a test event and sends it to the host.
- [GenApi::Integer & GuiXmlManifestAddress](#)
Description: Location of the GUI XML manifest table.
- [GenApi::Integer & Test0001](#)
Description: For testing only.
- [GenApi::IBoolean & V3_3Enable](#)
Description: Internally generated 3.3V rail.
- [GenApi::IEnumerationT< LineModeEnums > & LineMode](#)
Description: Controls if the physical Line is used to Input or Output a signal.
- [GenApi::IEnumerationT< LineSourceEnums > & LineSource](#)
Description: Selects which internal acquisition or I/O source signal to output on the selected line.
- [GenApi::IEnumerationT< LineInputFilterSelectorEnums > & LineInputFilterSelector](#)
Description: Selects the kind of input filter to configure: Deglitch or Debounce.
- [GenApi::IBoolean & UserOutputValue](#)
Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).
- [GenApi::Integer & UserOutputValueAll](#)
Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).
- [GenApi::IEnumerationT< UserOutputSelectorEnums > & UserOutputSelector](#)
Description: Selects which bit of the User Output register is set by UserOutputValue.
- [GenApi::IBoolean & LineStatus](#)
Description: Returns the current status of the selected input or output Line Visibility:
- [GenApi::IEnumerationT< LineFormatEnums > & LineFormat](#)
Description: Displays the current electrical format of the selected physical input or output Line.
- [GenApi::Integer & LineStatusAll](#)
Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).
- [GenApi::IEnumerationT< LineSelectorEnums > & LineSelector](#)
Description: Selects the physical line (or pin) of the external device connector to configure Visibility:
- [GenApi::IEnumerationT< ExposureActiveModeEnums > & ExposureActiveMode](#)
Description: Control sensor active exposure mode.
- [GenApi::IBoolean & LineInverter](#)
Description: Controls the inversion of the signal of the selected input or output line.
- [GenApi::Float & LineFilterWidth](#)
Description: Filter width in microseconds for the selected line and filter combination Visibility:
- [GenApi::IEnumerationT< CounterTriggerActivationEnums > & CounterTriggerActivation](#)
Description: Selects the activation mode of the trigger to start the counter.

- [GenApi::Integer](#) & [CounterValue](#)
Description: Current counter value Visibility:
- [GenApi::EnumerationT< CounterSelectorEnums >](#) & [CounterSelector](#)
Description: Selects which counter to configure Visibility:
- [GenApi::Integer](#) & [CounterValueAtReset](#)
Description: Value of the selected Counter when it was reset by a trigger.
- [GenApi::EnumerationT< CounterStatusEnums >](#) & [CounterStatus](#)
Description: Returns the current status of the counter.
- [GenApi::EnumerationT< CounterTriggerSourceEnums >](#) & [CounterTriggerSource](#)
Description: Selects the source of the trigger to start the counter Visibility:
- [GenApi::Integer](#) & [CounterDelay](#)
Description: Sets the delay (or number of events) before the CounterStart event is generated.
- [GenApi::EnumerationT< CounterResetSourceEnums >](#) & [CounterResetSource](#)
Description: Selects the signal that will be the source to reset the counter.
- [GenApi::EnumerationT< CounterEventSourceEnums >](#) & [CounterEventSource](#)
Description: Selects the event that will increment the counter Visibility:
- [GenApi::EnumerationT< CounterEventActivationEnums >](#) & [CounterEventActivation](#)
Description: Selects the activation mode of the event to increment the Counter.
- [GenApi::Integer](#) & [CounterDuration](#)
Description: Sets the duration (or number of events) before the CounterEnd event is generated.
- [GenApi::EnumerationT< CounterResetActivationEnums >](#) & [CounterResetActivation](#)
Description: Selects the Activation mode of the Counter Reset Source signal.
- [GenApi::EnumerationT< DeviceTypeEnums >](#) & [DeviceType](#)
Description: Returns the device type.
- [GenApi::IString](#) & [DeviceFamilyName](#)
Description: Identifier of the product family of the device.
- [GenApi::Integer](#) & [DeviceSFNCVersionMajor](#)
Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionMinor](#)
Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionSubMinor](#)
Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceManifestEntrySelector](#)
Description: Selects the manifest entry to reference.
- [GenApi::Integer](#) & [DeviceManifestXMLMajorVersion](#)
Description: Indicates the major version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLMinorVersion](#)
Description: Indicates the minor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLSubMinorVersion](#)
Description: Indicates the subminor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMajorVersion](#)
Description: Indicates the major version number of the schema file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMinorVersion](#)
Description: Indicates the minor version number of the schema file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestPrimaryURL](#)
Description: Indicates the first URL to the [GenICam XML](#) device description file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestSecondaryURL](#)
Description: Indicates the second URL to the [GenICam XML](#) device description file of the selected manifest entry.

- [GenApi::Integer](#) & [DeviceTLVersionSubMinor](#)
Description: Sub minor version of the Transport Layer of the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMajor](#)
Description: Major version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMinor](#)
Description: Minor version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceConnectionSelector](#)
Description: Selects which Connection of the device to control.
- [GenApi::Integer](#) & [DeviceConnectionSpeed](#)
Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.
- [GenApi::EnumerationT](#) < [DeviceConnectionStatusEnums](#) > & [DeviceConnectionStatus](#)
Description: Indicates the status of the specified Connection.
- [GenApi::Integer](#) & [DeviceLinkSelector](#)
Description: Selects which Link of the device to control.
- [GenApi::EnumerationT](#) < [DeviceLinkThroughputLimitModeEnums](#) > & [DeviceLinkThroughputLimitMode](#)
Description: Controls if the DeviceLinkThroughputLimit is active.
- [GenApi::Integer](#) & [DeviceLinkConnectionCount](#)
Description: Returns the number of physical connection of the device used by a particular Link.
- [GenApi::EnumerationT](#) < [DeviceLinkHeartbeatModeEnums](#) > & [DeviceLinkHeartbeatMode](#)
Description: Activate or deactivate the Link's heartbeat.
- [GenApi::Float](#) & [DeviceLinkHeartbeatTimeout](#)
Description: Controls the current heartbeat timeout of the specific Link.
- [GenApi::Float](#) & [DeviceLinkCommandTimeout](#)
Description: Indicates the command timeout of the specified Link.
- [GenApi::Integer](#) & [DeviceStreamChannelSelector](#)
Description: Selects the stream channel to control.
- [GenApi::EnumerationT](#) < [DeviceStreamChannelTypeEnums](#) > & [DeviceStreamChannelType](#)
Description: Reports the type of the stream channel.
- [GenApi::Integer](#) & [DeviceStreamChannelLink](#)
Description: Index of device's Link to use for streaming the specified stream channel.
- [GenApi::EnumerationT](#) < [DeviceStreamChannelEndiannessEnums](#) > & [DeviceStreamChannelEndianness](#)
Description: Endianness of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [DeviceStreamChannelPacketSize](#)
Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceStart](#)
Description: Indicate to the device and *GenICam* XML to get ready for persisting of all streamable features.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceEnd](#)
Description: Indicate to the device the end of feature persistence.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingStart](#)
Description: Prepare the device for registers streaming without checking for consistency.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingEnd](#)
Description: Announce the end of registers streaming.
- [GenApi::ICommand](#) & [DeviceRegistersCheck](#)
Description: Perform the validation of the current register set for consistency.
- [GenApi::Boolean](#) & [DeviceRegistersValid](#)
Description: Returns if the current register set is valid and consistent.
- [GenApi::EnumerationT](#) < [DeviceClockSelectorEnums](#) > & [DeviceClockSelector](#)
Description: Selects the clock frequency to access from the device.
- [GenApi::Float](#) & [DeviceClockFrequency](#)
Description: Returns the frequency of the selected Clock.

- [GenApi::IEnumerationT< DeviceSerialPortSelectorEnums > & DeviceSerialPortSelector](#)
Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > & DeviceSerialPortBaudRate](#)
Description: This feature controls the baud rate used by the selected serial port.
- [GenApi::Integer & Timestamp](#)
Description: Reports the current value of the device timestamp counter.
- [GenApi::IEnumerationT< SensorTapsEnums > & SensorTaps](#)
Description: Number of taps of the camera sensor.
- [GenApi::IEnumerationT< SensorDigitizationTapsEnums > & SensorDigitizationTaps](#)
Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
- [GenApi::IEnumerationT< RegionSelectorEnums > & RegionSelector](#)
Description: Selects the Region of interest to control.
- [GenApi::IEnumerationT< RegionModeEnums > & RegionMode](#)
Description: Controls if the selected Region of interest is active and streaming.
- [GenApi::IEnumerationT< RegionDestinationEnums > & RegionDestination](#)
Description: Control the destination of the selected region.
- [GenApi::IEnumerationT< ImageComponentSelectorEnums > & ImageComponentSelector](#)
Description: Selects a component to activate data streaming from.
- [GenApi::Boolean & ImageComponentEnable](#)
Description: Controls if the selected component streaming is active.
- [GenApi::Integer & LinePitch](#)
Description: Total number of bytes between 2 successive lines.
- [GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > & PixelFormatInfoSelector](#)
Description: Select the pixel format for which the information will be returned.
- [GenApi::Integer & PixelFormatInfoID](#)
Description: Returns the value used by the streaming channels to identify the selected pixel format.
- [GenApi::IEnumerationT< DeinterlacingEnums > & Deinterlacing](#)
Description: Controls how the device performs de-interlacing.
- [GenApi::IEnumerationT< ImageCompressionRateOptionEnums > & ImageCompressionRateOption](#)
Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- [GenApi::Integer & ImageCompressionQuality](#)
Description: Control the quality of the produced compressed stream.
- [GenApi::Float & ImageCompressionBitrate](#)
Description: Control the rate of the produced compressed stream.
- [GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > & ImageCompressionJPEGFormatOption](#)
Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.
- [GenApi::ICommand & AcquisitionAbort](#)
Description: Aborts the Acquisition immediately.
- [GenApi::ICommand & AcquisitionArm](#)
Description: Arms the device before an AcquisitionStart command.
- [GenApi::IEnumerationT< AcquisitionStatusSelectorEnums > & AcquisitionStatusSelector](#)
Description: Selects the internal acquisition signal to read using AcquisitionStatus.
- [GenApi::Boolean & AcquisitionStatus](#)
Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
- [GenApi::Integer & TriggerDivider](#)
Description: Specifies a division factor for the incoming trigger pulses.
- [GenApi::Integer & TriggerMultiplier](#)
Description: Specifies a multiplication factor for the incoming trigger pulses.
- [GenApi::IEnumerationT< ExposureTimeModeEnums > & ExposureTimeMode](#)
Description: Sets the configuration mode of the ExposureTime feature.

- [GenApi::IEnumerationT< ExposureTimeSelectorEnums > & ExposureTimeSelector](#)
Description: Selects which exposure time is controlled by the ExposureTime feature.
- [GenApi::IEnumerationT< GainAutoBalanceEnums > & GainAutoBalance](#)
Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.
- [GenApi::IEnumerationT< BlackLevelAutoEnums > & BlackLevelAuto](#)
Description: Controls the mode for automatic black level adjustment.
- [GenApi::IEnumerationT< BlackLevelAutoBalanceEnums > & BlackLevelAutoBalance](#)
Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.
- [GenApi::IEnumerationT< WhiteClipSelectorEnums > & WhiteClipSelector](#)
Description: Selects which White Clip to control.
- [GenApi::IFloat & WhiteClip](#)
Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.
- [GenApi::IRegister & LUTValueAll](#)
Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.
- [GenApi::Integer & UserOutputValueAllMask](#)
Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.
- [GenApi::ICommand & CounterReset](#)
Description: Does a software reset of the selected Counter and starts it.
- [GenApi::IEnumerationT< TimerSelectorEnums > & TimerSelector](#)
Description: Selects which Timer to configure.
- [GenApi::IFloat & TimerDuration](#)
Description: Sets the duration (in microseconds) of the Timer pulse.
- [GenApi::IFloat & TimerDelay](#)
Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.
- [GenApi::ICommand & TimerReset](#)
Description: Does a software reset of the selected timer and starts it.
- [GenApi::IFloat & TimerValue](#)
Description: Reads or writes the current value (in microseconds) of the selected Timer.
- [GenApi::IEnumerationT< TimerStatusEnums > & TimerStatus](#)
Description: Returns the current status of the Timer.
- [GenApi::IEnumerationT< TimerTriggerSourceEnums > & TimerTriggerSource](#)
Description: Selects the source of the trigger to start the Timer.
- [GenApi::IEnumerationT< TimerTriggerActivationEnums > & TimerTriggerActivation](#)
Description: Selects the activation mode of the trigger to start the Timer.
- [GenApi::IEnumerationT< EncoderSelectorEnums > & EncoderSelector](#)
Description: Selects which Encoder to configure.
- [GenApi::IEnumerationT< EncoderSourceAEnums > & EncoderSourceA](#)
Description: Selects the signal which will be the source of the A input of the Encoder.
- [GenApi::IEnumerationT< EncoderSourceBEnums > & EncoderSourceB](#)
Description: Selects the signal which will be the source of the B input of the Encoder.
- [GenApi::IEnumerationT< EncoderModeEnums > & EncoderMode](#)
Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.
- [GenApi::Integer & EncoderDivider](#)
Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.
- [GenApi::IEnumerationT< EncoderOutputModeEnums > & EncoderOutputMode](#)
Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.
- [GenApi::IEnumerationT< EncoderStatusEnums > & EncoderStatus](#)

- Description: Returns the motion status of the encoder.*

 - [GenApi::IFloat](#) & [EncoderTimeout](#)

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

- [GenApi::IEnumerationT< EncoderResetSourceEnums >](#) & [EncoderResetSource](#)

Description: Selects the signals that will be the source to reset the Encoder.

- [GenApi::IEnumerationT< EncoderResetActivationEnums >](#) & [EncoderResetActivation](#)

Description: Selects the Activation mode of the Encoder Reset Source signal.

- [GenApi::ICommand](#) & [EncoderReset](#)

Description: Does a software reset of the selected Encoder and starts it.

- [GenApi::Integer](#) & [EncoderValue](#)

Description: Reads or writes the current value of the position counter of the selected Encoder.

- [GenApi::Integer](#) & [EncoderValueAtReset](#)

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

- [GenApi::IEnumerationT< SoftwareSignalSelectorEnums >](#) & [SoftwareSignalSelector](#)

Description: Selects which Software Signal features to control.

- [GenApi::ICommand](#) & [SoftwareSignalPulse](#)

Description: Generates a pulse signal that can be used as a software trigger.

- [GenApi::IEnumerationT< ActionUnconditionalModeEnums >](#) & [ActionUnconditionalMode](#)

Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

- [GenApi::Integer](#) & [ActionDeviceKey](#)

Description: Provides the device key that allows the device to check the validity of action commands.

- [GenApi::Integer](#) & [ActionQueueSize](#)

Description: Indicates the size of the scheduled action commands queue.

- [GenApi::Integer](#) & [ActionSelector](#)

Description: Selects to which Action Signal further Action settings apply.

- [GenApi::Integer](#) & [ActionGroupMask](#)

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

- [GenApi::Integer](#) & [ActionGroupKey](#)

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

- [GenApi::Integer](#) & [EventAcquisitionTrigger](#)

Description: Returns the unique Identifier of the Acquisition Trigger type of Event.

- [GenApi::Integer](#) & [EventAcquisitionTriggerTimestamp](#)

Description: Returns the Timestamp of the Acquisition Trigger Event.

- [GenApi::Integer](#) & [EventAcquisitionTriggerFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.

- [GenApi::Integer](#) & [EventAcquisitionStart](#)

Description: Returns the unique Identifier of the Acquisition Start type of Event.

- [GenApi::Integer](#) & [EventAcquisitionStartTimestamp](#)

Description: Returns the Timestamp of the Acquisition Start Event.

- [GenApi::Integer](#) & [EventAcquisitionStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.

- [GenApi::Integer](#) & [EventAcquisitionEnd](#)

Description: Returns the unique Identifier of the Acquisition End type of Event.

- [GenApi::Integer](#) & [EventAcquisitionEndTimestamp](#)

Description: Returns the Timestamp of the Acquisition End Event.

- [GenApi::Integer](#) & [EventAcquisitionEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.

- [GenApi::Integer](#) & [EventAcquisitionTransferStart](#)
Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferStartTimestamp](#)
Description: Returns the Timestamp of the Acquisition Transfer Start Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEnd](#)
Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEndTimestamp](#)
Description: Returns the Timestamp of the Acquisition Transfer End Event.
- [GenApi::Integer](#) & [EventAcquisitionTransferEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.
- [GenApi::Integer](#) & [EventAcquisitionError](#)
Description: Returns the unique Identifier of the Acquisition Error type of Event.
- [GenApi::Integer](#) & [EventAcquisitionErrorTimestamp](#)
Description: Returns the Timestamp of the Acquisition Error Event.
- [GenApi::Integer](#) & [EventAcquisitionErrorFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.
- [GenApi::Integer](#) & [EventFrameTrigger](#)
Description: Returns the unique Identifier of the FrameTrigger type of Event.
- [GenApi::Integer](#) & [EventFrameTriggerTimestamp](#)
Description: Returns the Timestamp of the FrameTrigger Event.
- [GenApi::Integer](#) & [EventFrameTriggerFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.
- [GenApi::Integer](#) & [EventFrameStart](#)
Description: Returns the unique Identifier of the Frame Start type of Event.
- [GenApi::Integer](#) & [EventFrameStartTimestamp](#)
Description: Returns the Timestamp of the Frame Start Event.
- [GenApi::Integer](#) & [EventFrameStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.
- [GenApi::Integer](#) & [EventFrameEnd](#)
Description: Returns the unique Identifier of the Frame End type of Event.
- [GenApi::Integer](#) & [EventFrameEndTimestamp](#)
Description: Returns the Timestamp of the Frame End Event.
- [GenApi::Integer](#) & [EventFrameEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.
- [GenApi::Integer](#) & [EventFrameBurstStart](#)
Description: Returns the unique Identifier of the Frame Burst Start type of Event.
- [GenApi::Integer](#) & [EventFrameBurstStartTimestamp](#)
Description: Returns the Timestamp of the Frame Burst Start Event.
- [GenApi::Integer](#) & [EventFrameBurstStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.
- [GenApi::Integer](#) & [EventFrameBurstEnd](#)
Description: Returns the unique Identifier of the Frame Burst End type of Event.
- [GenApi::Integer](#) & [EventFrameBurstEndTimestamp](#)
Description: Returns the Timestamp of the Frame Burst End Event.
- [GenApi::Integer](#) & [EventFrameBurstEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.
- [GenApi::Integer](#) & [EventFrameTransferStart](#)
Description: Returns the unique Identifier of the Frame Transfer Start type of Event.
- [GenApi::Integer](#) & [EventFrameTransferStartTimestamp](#)

- Description: Returns the Timestamp of the Frame Transfer Start Event.*

 - [GenApi::Integer](#) & [EventFrameTransferStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.
- [GenApi::Integer](#) & [EventFrameTransferEnd](#)

Description: Returns the unique Identifier of the Frame Transfer End type of Event.
- [GenApi::Integer](#) & [EventFrameTransferEndTimestamp](#)

Description: Returns the Timestamp of the Frame Transfer End Event.
- [GenApi::Integer](#) & [EventFrameTransferEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.
- [GenApi::Integer](#) & [EventExposureStart](#)

Description: Returns the unique Identifier of the Exposure Start type of Event.
- [GenApi::Integer](#) & [EventExposureStartTimestamp](#)

Description: Returns the Timestamp of the Exposure Start Event.
- [GenApi::Integer](#) & [EventExposureStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.
- [GenApi::Integer](#) & [EventStream0TransferStart](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferStartTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Start Event.
- [GenApi::Integer](#) & [EventStream0TransferStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.
- [GenApi::Integer](#) & [EventStream0TransferEnd](#)

Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferEndTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer End Event.
- [GenApi::Integer](#) & [EventStream0TransferEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.
- [GenApi::Integer](#) & [EventStream0TransferPause](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.
- [GenApi::Integer](#) & [EventStream0TransferPauseTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.
- [GenApi::Integer](#) & [EventStream0TransferPauseFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.
- [GenApi::Integer](#) & [EventStream0TransferResume](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.
- [GenApi::Integer](#) & [EventStream0TransferResumeTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.
- [GenApi::Integer](#) & [EventStream0TransferResumeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockStart](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockStartTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockEnd](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockEndTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTrigger](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.
- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEnd](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEndTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.
- [GenApi::Integer](#) & [EventStream0TransferBurstEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflow](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflowTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.
- [GenApi::Integer](#) & [EventStream0TransferOverflowFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.
- [GenApi::Integer](#) & [EventSequencerSetChange](#)
Description: Returns the unique Identifier of the Sequencer Set Change type of Event.
- [GenApi::Integer](#) & [EventSequencerSetChangeTimestamp](#)
Description: Returns the Timestamp of the Sequencer Set Change Event.
- [GenApi::Integer](#) & [EventSequencerSetChangeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.
- [GenApi::Integer](#) & [EventCounter0Start](#)
Description: Returns the unique Identifier of the Counter 0 Start type of Event.
- [GenApi::Integer](#) & [EventCounter0StartTimestamp](#)
Description: Returns the Timestamp of the Counter 0 Start Event.
- [GenApi::Integer](#) & [EventCounter0StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.
- [GenApi::Integer](#) & [EventCounter1Start](#)
Description: Returns the unique Identifier of the Counter 1 Start type of Event.
- [GenApi::Integer](#) & [EventCounter1StartTimestamp](#)
Description: Returns the Timestamp of the Counter 1 Start Event.
- [GenApi::Integer](#) & [EventCounter1StartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.
- [GenApi::Integer](#) & [EventCounter0End](#)
Description: Returns the unique Identifier of the Counter 0 End type of Event.
- [GenApi::Integer](#) & [EventCounter0EndTimestamp](#)

- Description: Returns the Timestamp of the Counter 0 End Event.*

 - [GenApi::Integer](#) & [EventCounter0EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.
- [GenApi::Integer](#) & [EventCounter1End](#)

Description: Returns the unique Identifier of the Counter 1 End type of Event.

 - [GenApi::Integer](#) & [EventCounter1EndTimestamp](#)

Description: Returns the Timestamp of the Counter 1 End Event.

 - [GenApi::Integer](#) & [EventCounter1EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.
- [GenApi::Integer](#) & [EventTimer0Start](#)

Description: Returns the unique Identifier of the Timer 0 Start type of Event.

 - [GenApi::Integer](#) & [EventTimer0StartTimestamp](#)

Description: Returns the Timestamp of the Timer 0 Start Event.

 - [GenApi::Integer](#) & [EventTimer0StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.
- [GenApi::Integer](#) & [EventTimer1Start](#)

Description: Returns the unique Identifier of the Timer 1 Start type of Event.

 - [GenApi::Integer](#) & [EventTimer1StartTimestamp](#)

Description: Returns the Timestamp of the Timer 1 Start Event.

 - [GenApi::Integer](#) & [EventTimer1StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.
- [GenApi::Integer](#) & [EventTimer0End](#)

Description: Returns the unique Identifier of the Timer 0 End type of Event.

 - [GenApi::Integer](#) & [EventTimer0EndTimestamp](#)

Description: Returns the Timestamp of the Timer 0 End Event.

 - [GenApi::Integer](#) & [EventTimer0EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.
- [GenApi::Integer](#) & [EventTimer1End](#)

Description: Returns the unique Identifier of the Timer 1 End type of Event.

 - [GenApi::Integer](#) & [EventTimer1EndTimestamp](#)

Description: Returns the Timestamp of the Timer 1 End Event.

 - [GenApi::Integer](#) & [EventTimer1EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.
- [GenApi::Integer](#) & [EventEncoder0Stopped](#)

Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.

 - [GenApi::Integer](#) & [EventEncoder0StoppedTimestamp](#)

Description: Returns the Timestamp of the Encoder 0 Stopped Event.

 - [GenApi::Integer](#) & [EventEncoder0StoppedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder1Stopped](#)

Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.

 - [GenApi::Integer](#) & [EventEncoder1StoppedTimestamp](#)

Description: Returns the Timestamp of the Encoder 1 Stopped Event.

 - [GenApi::Integer](#) & [EventEncoder1StoppedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.
- [GenApi::Integer](#) & [EventEncoder0Restarted](#)

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.

 - [GenApi::Integer](#) & [EventEncoder0RestartedTimestamp](#)

Description: Returns the Timestamp of the Encoder 0 Restarted Event.

 - [GenApi::Integer](#) & [EventEncoder0RestartedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.

- [GenApi::Integer](#) & [EventEncoder1Restarted](#)
Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.
- [GenApi::Integer](#) & [EventEncoder1RestartedTimestamp](#)
Description: Returns the Timestamp of the Encoder 1 Restarted Event.
- [GenApi::Integer](#) & [EventEncoder1RestartedFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.
- [GenApi::Integer](#) & [EventLine0RisingEdge](#)
Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.
- [GenApi::Integer](#) & [EventLine0RisingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine0RisingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine1RisingEdge](#)
Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.
- [GenApi::Integer](#) & [EventLine1RisingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine1RisingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.
- [GenApi::Integer](#) & [EventLine0FallingEdge](#)
Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.
- [GenApi::Integer](#) & [EventLine0FallingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine0FallingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine1FallingEdge](#)
Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.
- [GenApi::Integer](#) & [EventLine1FallingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine1FallingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.
- [GenApi::Integer](#) & [EventLine0AnyEdge](#)
Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.
- [GenApi::Integer](#) & [EventLine0AnyEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Any Edge Event.
- [GenApi::Integer](#) & [EventLine0AnyEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.
- [GenApi::Integer](#) & [EventLine1AnyEdge](#)
Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.
- [GenApi::Integer](#) & [EventLine1AnyEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Any Edge Event.
- [GenApi::Integer](#) & [EventLine1AnyEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.
- [GenApi::Integer](#) & [EventLinkTrigger0](#)
Description: Returns the unique Identifier of the Link Trigger 0 type of Event.
- [GenApi::Integer](#) & [EventLinkTrigger0Timestamp](#)
Description: Returns the Timestamp of the Link Trigger 0 Event.
- [GenApi::Integer](#) & [EventLinkTrigger0FrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.
- [GenApi::Integer](#) & [EventLinkTrigger1](#)
Description: Returns the unique Identifier of the Link Trigger 1 type of Event.
- [GenApi::Integer](#) & [EventLinkTrigger1Timestamp](#)

- Description: Returns the Timestamp of the Link Trigger 1 Event.*

 - [GenApi::Integer](#) & [EventLinkTrigger1FrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.
- [GenApi::Integer](#) & [EventActionLate](#)

Description: Returns the unique Identifier of the Action Late type of Event.
- [GenApi::Integer](#) & [EventActionLateTimestamp](#)

Description: Returns the Timestamp of the Action Late Event.
- [GenApi::Integer](#) & [EventActionLateFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.
- [GenApi::Integer](#) & [EventLinkSpeedChange](#)

Description: Returns the unique Identifier of the Link Speed Change type of Event.
- [GenApi::Integer](#) & [EventLinkSpeedChangeTimestamp](#)

Description: Returns the Timestamp of the Link Speed Change Event.
- [GenApi::Integer](#) & [EventLinkSpeedChangeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.
- [GenApi::IRegister](#) & [FileAccessBuffer](#)

Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
- [GenApi::Integer](#) & [SourceCount](#)

Description: Controls or returns the number of sources supported by the device.
- [GenApi::IEnumerationT< SourceSelectorEnums >](#) & [SourceSelector](#)

Description: Selects the source to control.
- [GenApi::IEnumerationT< TransferSelectorEnums >](#) & [TransferSelector](#)

Description: Selects which stream transfers are currently controlled by the selected Transfer features.
- [GenApi::Integer](#) & [TransferBurstCount](#)

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.
- [GenApi::ICommand](#) & [TransferAbort](#)

Description: Aborts immediately the streaming of data block(s).
- [GenApi::ICommand](#) & [TransferPause](#)

Description: Pauses the streaming of data Block(s).
- [GenApi::ICommand](#) & [TransferResume](#)

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.
- [GenApi::IEnumerationT< TransferTriggerSelectorEnums >](#) & [TransferTriggerSelector](#)

Description: Selects the type of transfer trigger to configure.
- [GenApi::IEnumerationT< TransferTriggerModeEnums >](#) & [TransferTriggerMode](#)

Description: Controls if the selected trigger is active.
- [GenApi::IEnumerationT< TransferTriggerSourceEnums >](#) & [TransferTriggerSource](#)

Description: Specifies the signal to use as the trigger source for transfers.
- [GenApi::IEnumerationT< TransferTriggerActivationEnums >](#) & [TransferTriggerActivation](#)

Description: Specifies the activation mode of the transfer control trigger.
- [GenApi::IEnumerationT< TransferStatusSelectorEnums >](#) & [TransferStatusSelector](#)

Description: Selects which status of the transfer module to read.
- [GenApi::IBoolean](#) & [TransferStatus](#)

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.
- [GenApi::IEnumerationT< TransferComponentSelectorEnums >](#) & [TransferComponentSelector](#)

Description: Selects the color component for the control of the TransferStreamChannel feature.
- [GenApi::Integer](#) & [TransferStreamChannel](#)

Description: Selects the streaming channel that will be used to transfer the selected stream of data.
- [GenApi::IEnumerationT< Scan3dDistanceUnitEnums >](#) & [Scan3dDistanceUnit](#)

Description: Specifies the unit used when delivering calibrated distance data.
- [GenApi::IEnumerationT< Scan3dCoordinateSystemEnums >](#) & [Scan3dCoordinateSystem](#)

- Description: Specifies the Coordinate system to use for the device.*

 - [GenApi::IEnumerationT < Scan3dOutputModeEnums > & Scan3dOutputMode](#)

Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.

- [GenApi::IEnumerationT < Scan3dCoordinateSystemReferenceEnums > & Scan3dCoordinateSystemReference](#)

Description: Defines coordinate system reference location.

- [GenApi::IEnumerationT < Scan3dCoordinateSelectorEnums > & Scan3dCoordinateSelector](#)

Description: Selects the individual coordinates in the vectors for 3D information/transformation.

- [GenApi::IFloat & Scan3dCoordinateScale](#)

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

- [GenApi::IFloat & Scan3dCoordinateOffset](#)

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

- [GenApi::IBoolean & Scan3dInvalidDataFlag](#)

Description: Enables the definition of a non-valid flag value in the data stream.

- [GenApi::IFloat & Scan3dInvalidDataValue](#)

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

- [GenApi::IFloat & Scan3dAxisMin](#)

Description: Minimum valid transmitted coordinate value of the selected Axis.

- [GenApi::IFloat & Scan3dAxisMax](#)

Description: Maximum valid transmitted coordinate value of the selected Axis.

- [GenApi::IEnumerationT < Scan3dCoordinateTransformSelectorEnums > & Scan3dCoordinateTransformSelector](#)

Description: Sets the index to read/write a coordinate transform value.

- [GenApi::IFloat & Scan3dTransformValue](#)

Description: Specifies the transform value selected.

- [GenApi::IEnumerationT < Scan3dCoordinateReferenceSelectorEnums > & Scan3dCoordinateReferenceSelector](#)

Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

- [GenApi::IFloat & Scan3dCoordinateReferenceValue](#)

Description: Returns the reference value selected.

- [GenApi::Integer & ChunkPartSelector](#)

Description: Selects the part to access in chunk data in a multipart transmission.

- [GenApi::IEnumerationT < ChunkImageComponentEnums > & ChunkImageComponent](#)

Description: Returns the component of the payload image.

- [GenApi::Integer & ChunkPixelDynamicRangeMin](#)

Description: Returns the minimum value of dynamic range of the image included in the payload.

- [GenApi::Integer & ChunkPixelDynamicRangeMax](#)

Description: Returns the maximum value of dynamic range of the image included in the payload.

- [GenApi::Integer & ChunkTimestampLatchValue](#)

Description: Returns the last Timestamp latched with the TimestampLatch command.

- [GenApi::Integer & ChunkLineStatusAll](#)

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

- [GenApi::IEnumerationT < ChunkCounterSelectorEnums > & ChunkCounterSelector](#)

Description: Selects which counter to retrieve data from.

- [GenApi::Integer & ChunkCounterValue](#)

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

- [GenApi::IEnumerationT < ChunkTimerSelectorEnums > & ChunkTimerSelector](#)

Description: Selects which Timer to retrieve data from.

- [GenApi::IFloat & ChunkTimerValue](#)

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

- [GenApi::IEnumerationT < ChunkEncoderSelectorEnums > & ChunkEncoderSelector](#)

Description: Selects which Encoder to retrieve data from.

- [GenApi::Integer & ChunkScanLineSelector](#)

- Description: Index for vector representation of one chunk value per line in an image.*
- [GenApi::Integer](#) & [ChunkEncoderValue](#)
Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.
 - [GenApi::EnumerationT< ChunkEncoderStatusEnums >](#) & [ChunkEncoderStatus](#)
Description: Returns the motion status of the selected encoder.
 - [GenApi::EnumerationT< ChunkExposureTimeSelectorEnums >](#) & [ChunkExposureTimeSelector](#)
Description: Selects which exposure time is read by the ChunkExposureTime feature.
 - [GenApi::Integer](#) & [ChunkLinePitch](#)
Description: Returns the LinePitch of the image included in the payload.
 - [GenApi::EnumerationT< ChunkSourceIDEnums >](#) & [ChunkSourceID](#)
Description: Returns the identifier of Source that the image comes from.
 - [GenApi::EnumerationT< ChunkRegionIDEnums >](#) & [ChunkRegionID](#)
Description: Returns the identifier of Region that the image comes from.
 - [GenApi::Integer](#) & [ChunkTransferBlockID](#)
Description: Returns the unique identifier of the transfer block used to transport the payload.
 - [GenApi::EnumerationT< ChunkTransferStreamIDEnums >](#) & [ChunkTransferStreamID](#)
Description: Returns identifier of the stream that generated this block.
 - [GenApi::Integer](#) & [ChunkTransferQueueCurrentBlockCount](#)
Description: Returns the current number of blocks in the transfer queue.
 - [GenApi::Integer](#) & [ChunkStreamChannelID](#)
Description: Returns identifier of the stream channel used to carry the block.
 - [GenApi::EnumerationT< ChunkScan3dDistanceUnitEnums >](#) & [ChunkScan3dDistanceUnit](#)
Description: Returns the Distance Unit of the payload image.
 - [GenApi::EnumerationT< ChunkScan3dOutputModeEnums >](#) & [ChunkScan3dOutputMode](#)
Description: Returns the Calibrated Mode of the payload image.
 - [GenApi::EnumerationT< ChunkScan3dCoordinateSystemEnums >](#) & [ChunkScan3dCoordinateSystem](#)
Description: Returns the Coordinate System of the image included in the payload.
 - [GenApi::EnumerationT< ChunkScan3dCoordinateSystemReferenceEnums >](#) & [ChunkScan3dCoordinateSystemReference](#)
Description: Returns the Coordinate System Position of the image included in the payload.
 - [GenApi::EnumerationT< ChunkScan3dCoordinateSelectorEnums >](#) & [ChunkScan3dCoordinateSelector](#)
Description: Selects which Coordinate to retrieve data from.
 - [GenApi::IFloat](#) & [ChunkScan3dCoordinateScale](#)
Description: Returns the Scale for the selected coordinate axis of the image included in the payload.
 - [GenApi::IFloat](#) & [ChunkScan3dCoordinateOffset](#)
Description: Returns the Offset for the selected coordinate axis of the image included in the payload.
 - [GenApi::Boolean](#) & [ChunkScan3dInvalidDataFlag](#)
Description: Returns if a specific non-valid data flag is used in the data stream.
 - [GenApi::IFloat](#) & [ChunkScan3dInvalidDataValue](#)
Description: Returns the Invalid Data Value used for the image included in the payload.
 - [GenApi::IFloat](#) & [ChunkScan3dAxisMin](#)
Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.
 - [GenApi::IFloat](#) & [ChunkScan3dAxisMax](#)
Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.
 - [GenApi::EnumerationT< ChunkScan3dCoordinateTransformSelectorEnums >](#) & [ChunkScan3dCoordinateTransformSelector](#)
Description: Selector for transform values.
 - [GenApi::IFloat](#) & [ChunkScan3dTransformValue](#)
Description: Returns the transform value.
 - [GenApi::EnumerationT< ChunkScan3dCoordinateReferenceSelectorEnums >](#) & [ChunkScan3dCoordinateReferenceSelector](#)
Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

- [GenApi::IFloat](#) & [ChunkScan3dCoordinateReferenceValue](#)
Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.
- [GenApi::Integer](#) & [TestPendingAck](#)
Description: Tests the device's pending acknowledge feature.
- [GenApi::IEnumerationT](#) < [DeviceTapGeometryEnums](#) > & [DeviceTapGeometry](#)
Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.
- [GenApi::IEnumerationT](#) < [GevPhysicalLinkConfigurationEnums](#) > & [GevPhysicalLinkConfiguration](#)
Description: Controls the principal physical link configuration to use on next restart/power-up of the device.
- [GenApi::IEnumerationT](#) < [GevCurrentPhysicalLinkConfigurationEnums](#) > & [GevCurrentPhysicalLinkConfiguration](#)
Description: Indicates the current physical link configuration of the device.
- [GenApi::Integer](#) & [GevActiveLinkCount](#)
Description: Indicates the current number of active logical links.
- [GenApi::Boolean](#) & [GevPAUSEFrameReception](#)
Description: Controls whether incoming PAUSE Frames are handled on the given logical link.
- [GenApi::Boolean](#) & [GevPAUSEFrameTransmission](#)
Description: Controls whether PAUSE Frames can be generated on the given logical link.
- [GenApi::IEnumerationT](#) < [GevIPConfigurationStatusEnums](#) > & [GevIPConfigurationStatus](#)
Description: Reports the current IP configuration status.
- [GenApi::Integer](#) & [GevDiscoveryAckDelay](#)
Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
- [GenApi::IEnumerationT](#) < [GevGVCPEExtendedStatusCodesSelectorEnums](#) > & [GevGVCPEExtendedStatusCodesSelector](#)
Description: Selects the GigE Vision version to control extended status codes for.
- [GenApi::Boolean](#) & [GevGVCPEExtendedStatusCodes](#)
Description: Enables the generation of extended status codes.
- [GenApi::Integer](#) & [GevPrimaryApplicationSwitchoverKey](#)
Description: Controls the key to use to authenticate primary application switchover requests.
- [GenApi::IEnumerationT](#) < [GevGVSPExtendedIDModeEnums](#) > & [GevGVSPExtendedIDMode](#)
Description: Enables the extended IDs mode.
- [GenApi::Integer](#) & [GevPrimaryApplicationSocket](#)
Description: Returns the UDP source port of the primary application.
- [GenApi::Integer](#) & [GevPrimaryApplicationIPAddress](#)
Description: Returns the address of the primary application.
- [GenApi::Boolean](#) & [GevSCCFGPacketResendDestination](#)
Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.
- [GenApi::Boolean](#) & [GevSCCFGAllInTransmission](#)
Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.
- [GenApi::Integer](#) & [GevSCZoneCount](#)
Description: Reports the number of zones per block transmitted on the selected stream channel.
- [GenApi::Integer](#) & [GevSCZoneDirectionAll](#)
Description: Reports the transmission direction of each zone transmitted on the selected stream channel.
- [GenApi::Boolean](#) & [GevSCZoneConfigurationLock](#)
Description: Controls whether the selected stream channel multi-zone configuration is locked.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesTransmitted](#)
Description: Reports the number of transmitted PAUSE frames.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesReceived](#)
Description: Reports the number of received PAUSE frames.
- [GenApi::IEnumerationT](#) < [CIConfigurationEnums](#) > & [CIConfiguration](#)
Description: This [Camera](#) Link specific feature describes the configuration used by the camera.
- [GenApi::IEnumerationT](#) < [CITimeSlotsCountEnums](#) > & [CITimeSlotsCount](#)

Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationStatusEnums](#) > & [CxpLinkConfigurationStatus](#)

Description: This feature indicates the current and active Link configuration used by the Device.

- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationPreferredEnums](#) > & [CxpLinkConfigurationPreferred](#)

Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

- [GenApi::IEnumerationT](#) < [CxpLinkConfigurationEnums](#) > & [CxpLinkConfiguration](#)

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.

- [GenApi::Integer](#) & [CxpConnectionSelector](#)

Description: Selects the CoaXPress physical connection to control.

- [GenApi::IEnumerationT](#) < [CxpConnectionTestModeEnums](#) > & [CxpConnectionTestMode](#)

Description: Enables the test mode for an individual physical connection of the Device.

- [GenApi::Integer](#) & [CxpConnectionTestErrorCount](#)

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).

- [GenApi::Integer](#) & [CxpConnectionTestPacketCount](#)

Description: Reports the current count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).

- [GenApi::ICommand](#) & [CxpPoCxpAuto](#)

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

- [GenApi::ICommand](#) & [CxpPoCxpTurnOff](#)

Description: Disable Power over CoaXPress (PoCXP) for the Link.

- [GenApi::ICommand](#) & [CxpPoCxpTripReset](#)

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

- [GenApi::IEnumerationT](#) < [CxpPoCxpStatusEnums](#) > & [CxpPoCxpStatus](#)

Description: Returns the Power over CoaXPress (PoCXP) status of the Device.

- [GenApi::Integer](#) & [ChunkInferenceFrameId](#)

Description: Returns the frame ID associated with the most recent inference result.

- [GenApi::Integer](#) & [ChunkInferenceResult](#)

Description: Returns the chunk data inference result.

- [GenApi::IFloat](#) & [ChunkInferenceConfidence](#)

Description: Returns the chunk data inference confidence percentage.

- [GenApi::IRegister](#) & [ChunkInferenceBoundingBoxResult](#)

Description: Returns the chunk inference bounding box result data.

Protected Member Functions

- [Camera](#) ()

Additional Inherited Members

15.9.1 Detailed Description

The camera object class.

15.9.2 Constructor & Destructor Documentation

15.9.2.1 ~Camera()

`~Camera ( )`

15.9.2.2 Camera()

`Camera ( )` [protected]

15.9.3 Member Function Documentation

15.9.3.1 Init()

`void Init ( )` [virtual]

Implements [ICameraBase](#).

15.9.4 Member Data Documentation

15.9.4.1 AasRoiEnable

`GenApi::IBoolean& AasRoiEnable`

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↔ Selector feature.

Visibility:

15.9.4.2 AasRoiHeight

`GenApi::IInteger& AasRoiHeight`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

15.9.4.3 AasRoiOffsetX

`GenApi::Integer& AasRoiOffsetX`

Description:

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

15.9.4.4 AasRoiOffsetY

`GenApi::Integer& AasRoiOffsetY`

Description:

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

15.9.4.5 AasRoiWidth

`GenApi::Integer& AasRoiWidth`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

15.9.4.6 AcquisitionAbort

`GenApi::ICommand& AcquisitionAbort`

Description: Aborts the Acquisition immediately.

This will end the capture without completing the current Frame or waiting on a trigger. If no Acquisition is in progress, the command is ignored. Visibility: Expert

15.9.4.7 AcquisitionArm

`GenApi::ICommand& AcquisitionArm`

Description: Arms the device before an AcquisitionStart command.

This optional command validates all the current features for consistency and prepares the device for a fast start of the Acquisition. Visibility: Expert

15.9.4.8 AcquisitionBurstFrameCount

`GenApi::IInteger& AcquisitionBurstFrameCount`

Description:

This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled. Note that the total number of frames captured is also conditioned by AcquisitionFrameCount if AcquisitionMode is Multi-Frame and ignored if AcquisitionMode is Single.

Visibility:

15.9.4.9 AcquisitionFrameCount

`GenApi::IInteger& AcquisitionFrameCount`

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

15.9.4.10 AcquisitionFrameRate

`GenApi::IFloat& AcquisitionFrameRate`

Description: User controlled acquisition frame rate in Hertz Visibility:

15.9.4.11 AcquisitionFrameRateEnable

`GenApi::IBoolean& AcquisitionFrameRateEnable`

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

15.9.4.12 AcquisitionLineRate

`GenApi::IFloat& AcquisitionLineRate`

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

15.9.4.13 AcquisitionMode

`GenApi::IEnumerationT<AcquisitionModeEnums>& AcquisitionMode`

Description: Sets the acquisition mode of the device.

Continuous: acquires images continuously. Multi Frame: acquires a specified number of images before stopping acquisition. Single Frame: acquires 1 image before stopping acquisition. Visibility:

15.9.4.14 AcquisitionResultingFrameRate

`GenApi::IFloat& AcquisitionResultingFrameRate`

Description: Resulting frame rate in Hertz.

If this does not equal the Acquisition Frame Rate it is because the Exposure Time is greater than the frame time.

Visibility:

15.9.4.15 AcquisitionStart

`GenApi::ICommand& AcquisitionStart`

Description: This command starts the acquisition of images.

Visibility:

15.9.4.16 AcquisitionStatus

`GenApi::IBoolean& AcquisitionStatus`

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

15.9.4.17 AcquisitionStatusSelector

`GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector`

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

15.9.4.18 AcquisitionStop

`GenApi::ICommand& AcquisitionStop`

Description: This command stops the acquisition of images.

Visibility:

15.9.4.19 ActionDeviceKey

`GenApi::IInteger& ActionDeviceKey`

Description: Provides the device key that allows the device to check the validity of action commands.

The device internal assertion of an action signal is only authorized if the ActionDeviceKey and the action device key value in the protocol message are equal. Visibility: Guru

15.9.4.20 ActionGroupKey

`GenApi::IInteger& ActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

15.9.4.21 ActionGroupMask

`GenApi::IInteger& ActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Guru

15.9.4.22 ActionQueueSize

`GenApi::IInteger& ActionQueueSize`

Description: Indicates the size of the scheduled action commands queue.

This number represents the maximum number of scheduled action commands that can be pending at a given point in time. Visibility: Guru

15.9.4.23 ActionSelector

`GenApi::IInteger& ActionSelector`

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

15.9.4.24 ActionUnconditionalMode

`GenApi::IEnumerationT<ActionUnconditionalModeEnums>& ActionUnconditionalMode`

Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.

Visibility: Guru

15.9.4.25 AdaptiveCompressionEnable

`GenApi::IBoolean& AdaptiveCompressionEnable`

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

15.9.4.26 AdcBitDepth

`GenApi::IEnumerationT<AdcBitDepthEnums>& AdcBitDepth`

Description:

Selects which ADC bit depth to use. A higher ADC bit depth results in better image quality but slower maximum frame rate.

Visibility:

15.9.4.27 aPAUSEMACCtrlFramesReceived

`GenApi::IInteger& aPAUSEMACCtrlFramesReceived`

Description: Reports the number of received PAUSE frames.

Visibility: Guru

15.9.4.28 aPAUSEMACCtrlFramesTransmitted

`GenApi::IInteger& aPAUSEMACCtrlFramesTransmitted`

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

15.9.4.29 AutoAlgorithmSelector

`GenApi::IEnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector`

Description:

Selects which Auto Algorithm is controlled by the RoiEnable, OffsetX, OffsetY, Width, Height features.

Visibility:

15.9.4.30 AutoExposureControlLoopDamping

`GenApi::IFloat& AutoExposureControlLoopDamping`

Description:

It controls how fast the exposure and gain get settled. If the value is too small, it may cause the system to be unstable. Range is from 0.0 to 1.0. Default = 0.2.

Visibility:

15.9.4.31 AutoExposureControlPriority

`GenApi::IEnumerationT<AutoExposureControlPriorityEnums>& AutoExposureControlPriority`

Description:

Selects whether to adjust gain or exposure first. When gain priority is selected, the camera fixes the gain to 0 dB, and the exposure is adjusted according to the target grey level. If the maximum exposure is reached before the target grey level is hit, the gain starts to change to meet the target. This mode is used to have the minimum noise. When exposure priority is selected, the camera sets the exposure to a small value (default is 5 ms). The gain is adjusted according to the target grey level. If maximum gain is reached before the target grey level is hit, the exposure starts to change to meet the target. This mode is used to capture fast motion.

Visibility:

15.9.4.32 AutoExposureEVCompensation

`GenApi::IFloat& AutoExposureEVCompensation`

Description:

The EV compensation value used in the exposure compensation. This allows you to adjust the resultant image intensity with one control. A positive value makes the image brighter. A negative value makes the image darker. Range from -3 to 3 with a step of 1/3. Default = 0.

Visibility:

15.9.4.33 AutoExposureExposureTimeLowerLimit

`GenApi::IFloat & AutoExposureExposureTimeLowerLimit`

Description:

The smallest exposure time that auto exposure can set.

Visibility:

15.9.4.34 AutoExposureExposureTimeUpperLimit

`GenApi::IFloat & AutoExposureExposureTimeUpperLimit`

Description:

The largest exposure time that auto exposure can set.

Visibility:

15.9.4.35 AutoExposureGainLowerLimit

`GenApi::IFloat & AutoExposureGainLowerLimit`

Description:

The smallest gain that auto exposure can set.

Visibility:

15.9.4.36 AutoExposureGainUpperLimit

`GenApi::IFloat & AutoExposureGainUpperLimit`

Description:

The largest gain that auto exposure can set.

Visibility:

15.9.4.37 AutoExposureGreyValueLowerLimit

`GenApi::IFloat& AutoExposureGreyValueLowerLimit`

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

15.9.4.38 AutoExposureGreyValueUpperLimit

`GenApi::IFloat& AutoExposureGreyValueUpperLimit`

Description:

The highest value in percentage that the target mean may reach.

Visibility:

15.9.4.39 AutoExposureLightingMode

`GenApi::IEnumerationT<AutoExposureLightingModeEnums>& AutoExposureLightingMode`

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default). a. Backlight compensation: used when a strong light is coming from the back of the object. b. Frontlight compensation: used when a strong light is shining in the front of the object while the background is dark. c. Normal lighting: used when the object is not under backlight or frontlight conditions. When normal lighting is selected, metering modes are available.

Visibility:

15.9.4.40 AutoExposureMeteringMode

`GenApi::IEnumerationT<AutoExposureMeteringModeEnums>& AutoExposureMeteringMode`

Description:

Selects a metering mode: average, spot, or partial metering. a. Average: Measures the light from the entire scene uniformly to determine the final exposure value. Every portion of the exposed area has the same contribution. b. Spot: Measures a small area (about 3%) in the center of the scene while the rest of the scene is ignored. This mode is used when the scene has a high contrast and the object of interest is relatively small. c. Partial: Measures the light from a larger area (about 11%) in the center of the scene. This mode is used when very dark or bright regions appear at the edge of the frame. Note: Metering mode is available only when Lighting Mode Selector is Normal.

Visibility:

15.9.4.41 AutoExposureTargetGreyValue

`GenApi::IFloat& AutoExposureTargetGreyValue`

Description:

This is the user-specified target grey level (image mean) to apply to the current image. Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

15.9.4.42 AutoExposureTargetGreyValueAuto

`GenApi::IEnumerationT<AutoExposureTargetGreyValueAutoEnums>& AutoExposureTargetGreyValueAuto`

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user. Note that the target grey level is in the linear domain before gamma correction is applied.

Visibility:

15.9.4.43 BalanceRatio

`GenApi::IFloat& BalanceRatio`

Description:

Controls the balance ratio of the selected color relative to green. Used for white balancing.

Visibility:

15.9.4.44 BalanceRatioSelector

`GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector`

Description:

Selects a balance ratio to configure once a balance ratio control has been selected.

Visibility:

15.9.4.45 BalanceWhiteAuto

`GenApi::IEnumerationT<BalanceWhiteAutoEnums>& BalanceWhiteAuto`

Description:

White Balance compensates for color shifts caused by different lighting conditions. It can be automatically or manually controlled. For manual control, set to Off. For automatic control, set to Once or Continuous.

Visibility:

15.9.4.46 BalanceWhiteAutoDamping

`GenApi::IFloat& BalanceWhiteAutoDamping`

Description:

Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions. Higher damping means the changes are more gradual.

Visibility:

15.9.4.47 BalanceWhiteAutoLowerLimit

`GenApi::IFloat& BalanceWhiteAutoLowerLimit`

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

15.9.4.48 BalanceWhiteAutoProfile

`GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile`

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

15.9.4.49 BalanceWhiteAutoUpperLimit

`GenApi::IFloat& BalanceWhiteAutoUpperLimit`

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

15.9.4.50 BinningHorizontal

`GenApi::IInteger& BinningHorizontal`

Description:

Number of horizontal photo-sensitive cells to combine together. This reduces the horizontal resolution (width) of the image. A value of 1 indicates that no horizontal binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

15.9.4.51 BinningHorizontalMode

`GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode`

Description: Visibility:

15.9.4.52 BinningSelector

`GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector`

Description:

Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.

Visibility:

15.9.4.53 BinningVertical

`GenApi::Integer& BinningVertical`

Description:

Number of vertical photo-sensitive cells to combine together. This reduces the vertical resolution (height) of the image. A value of 1 indicates that no vertical binning is performed by the camera. This value must be 1 for decimation to be active.

Visibility:

15.9.4.54 BinningVerticalMode

`GenApi::EnumerationT<BinningVerticalModeEnums>& BinningVerticalMode`

Description: Visibility:

15.9.4.55 BlackLevel

`GenApi::Float& BlackLevel`

Description:

Controls the offset of the video signal in percent.

Visibility:

15.9.4.56 BlackLevelAuto

`GenApi::EnumerationT<BlackLevelAutoEnums>& BlackLevelAuto`

Description: Controls the mode for automatic black level adjustment.

The exact algorithm used to implement this adjustment is device-specific. Visibility: Expert

15.9.4.57 BlackLevelAutoBalance

`GenApi::EnumerationT<BlackLevelAutoBalanceEnums>& BlackLevelAutoBalance`

Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.

The black level coefficients of each channel are adjusted so they are matched. Visibility: Expert

15.9.4.58 BlackLevelClampingEnable

`GenApi::IBoolean& BlackLevelClampingEnable`

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

15.9.4.59 BlackLevelRaw

`GenApi::IInteger& BlackLevelRaw`

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

15.9.4.60 BlackLevelSelector

`GenApi::IEnumerationT<BlackLevelSelectorEnums>& BlackLevelSelector`

Description:

Selects which black level to control. Only All can be set by the user. Analog and Digital are read-only.

Visibility:

15.9.4.61 ChunkBlackLevel

`GenApi::IFloat& ChunkBlackLevel`

Description: Returns the black level used to capture the image.

Visibility:

15.9.4.62 ChunkBlackLevelSelector

`GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector`

Description: Selects which black level to retrieve Visibility:

15.9.4.63 ChunkCompressionMode

`GenApi::IInteger& ChunkCompressionMode`

Description: Returns the compression mode of the last image payload.

Visibility:

15.9.4.64 ChunkCompressionRatio

`GenApi::IFloat& ChunkCompressionRatio`

Description: Returns the compression ratio of the last image payload.

Visibility:

15.9.4.65 ChunkCounterSelector

`GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector`

Description: Selects which counter to retrieve data from.

Visibility: Expert

15.9.4.66 ChunkCounterValue

`GenApi::IInteger& ChunkCounterValue`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

15.9.4.67 ChunkCRC

`GenApi::IInteger& ChunkCRC`

Description: Returns the CRC of the image payload.

Visibility:

15.9.4.68 ChunkEnable

`GenApi::IBoolean & ChunkEnable`

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

15.9.4.69 ChunkEncoderSelector

`GenApi::IEnumerationT<ChunkEncoderSelectorEnums> & ChunkEncoderSelector`

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

15.9.4.70 ChunkEncoderStatus

`GenApi::IEnumerationT<ChunkEncoderStatusEnums> & ChunkEncoderStatus`

Description: Returns the motion status of the selected encoder.

Visibility: Expert

15.9.4.71 ChunkEncoderValue

`GenApi::IInteger & ChunkEncoderValue`

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

15.9.4.72 ChunkExposureEndLineStatusAll

`GenApi::IInteger & ChunkExposureEndLineStatusAll`

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

15.9.4.73 ChunkExposureTime

`GenApi::IFloat & ChunkExposureTime`

Description: Returns the exposure time used to capture the image.

Visibility:

15.9.4.74 ChunkExposureTimeSelector

`GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector`

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

15.9.4.75 ChunkFrameID

`GenApi::IInteger& ChunkFrameID`

Description: Returns the image frame ID.

Visibility:

15.9.4.76 ChunkGain

`GenApi::IFloat& ChunkGain`

Description: Returns the gain used to capture the image.

Visibility:

15.9.4.77 ChunkGainSelector

`GenApi::IEnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector`

Description: Selects which gain to retrieve Visibility:

15.9.4.78 ChunkHeight

`GenApi::IInteger& ChunkHeight`

Description: Returns the height of the image included in the payload.

Visibility:

15.9.4.79 ChunkImage

`GenApi::IInteger& ChunkImage`

Description: Returns the image payload.

Visibility:

15.9.4.80 ChunkImageComponent

`GenApi::IEnumerationT<ChunkImageComponentEnums>& ChunkImageComponent`

Description: Returns the component of the payload image.

This can be used to identify the image component of a generic part in a multipart transfer. Visibility: Expert

15.9.4.81 ChunkInferenceBoundingBoxResult

`GenApi::IRegister& ChunkInferenceBoundingBoxResult`

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

15.9.4.82 ChunkInferenceConfidence

`GenApi::IFloat& ChunkInferenceConfidence`

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

15.9.4.83 ChunkInferenceFrameId

`GenApi::IInteger& ChunkInferenceFrameId`

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

15.9.4.84 ChunkInferenceResult

`GenApi::IInteger& ChunkInferenceResult`

Description: Returns the chunk data inference result.

Visibility: Expert

15.9.4.85 ChunkLinePitch

`GenApi::IInteger& ChunkLinePitch`

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

15.9.4.86 ChunkLineStatusAll

`GenApi::Integer& ChunkLineStatusAll`

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

15.9.4.87 ChunkModeActive

`GenApi::Boolean& ChunkModeActive`

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

15.9.4.88 ChunkOffsetX

`GenApi::Integer& ChunkOffsetX`

Description: Returns the Offset X of the image included in the payload.

Visibility:

15.9.4.89 ChunkOffsetY

`GenApi::Integer& ChunkOffsetY`

Description: Returns the Offset Y of the image included in the payload.

Visibility:

15.9.4.90 ChunkPartSelector

`GenApi::Integer& ChunkPartSelector`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

15.9.4.91 ChunkPixelDynamicRangeMax

`GenApi::Integer& ChunkPixelDynamicRangeMax`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

15.9.4.92 ChunkPixelDynamicRangeMin

`GenApi::Integer& ChunkPixelDynamicRangeMin`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

15.9.4.93 ChunkPixelFormat

`GenApi::EnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat`

Description: Format of the pixel provided by the camera Visibility:

15.9.4.94 ChunkRegionID

`GenApi::EnumerationT<ChunkRegionIDEnums>& ChunkRegionID`

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

15.9.4.95 ChunkScan3dAxisMax

`GenApi::IFloat& ChunkScan3dAxisMax`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.9.4.96 ChunkScan3dAxisMin

`GenApi::IFloat& ChunkScan3dAxisMin`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.9.4.97 ChunkScan3dCoordinateOffset

`GenApi::IFloat& ChunkScan3dCoordinateOffset`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.9.4.98 ChunkScan3dCoordinateReferenceSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateReferenceSelectorEnums>& ChunkScan3dCoordinateReferenceSelector`

Description: Selector to read a coordinate system reference value defining the transform of a point from one system to the other.

Visibility: Expert

15.9.4.99 ChunkScan3dCoordinateReferenceValue

`GenApi::IFloat& ChunkScan3dCoordinateReferenceValue`

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

15.9.4.100 ChunkScan3dCoordinateScale

`GenApi::IFloat& ChunkScan3dCoordinateScale`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

15.9.4.101 ChunkScan3dCoordinateSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector`

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

15.9.4.102 ChunkScan3dCoordinateSystem

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem`

Description: Returns the Coordinate [System](#) of the image included in the payload.

Visibility: Expert

15.9.4.103 ChunkScan3dCoordinateSystemReference

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinateSystemReference`

Description: Returns the Coordinate [System](#) Position of the image included in the payload.

Visibility: Expert

15.9.4.104 ChunkScan3dCoordinateTransformSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinateTransformSelector`

Description: Selector for transform values.

Visibility: Expert

15.9.4.105 ChunkScan3dDistanceUnit

`GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit`

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

15.9.4.106 ChunkScan3dInvalidDataFlag

`GenApi::IBoolean& ChunkScan3dInvalidDataFlag`

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

15.9.4.107 ChunkScan3dInvalidDataValue

`GenApi::IFloat& ChunkScan3dInvalidDataValue`

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

15.9.4.108 ChunkScan3dOutputMode

`GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode`

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

15.9.4.109 ChunkScan3dTransformValue

`GenApi::IFloat& ChunkScan3dTransformValue`

Description: Returns the transform value.

Visibility: Expert

15.9.4.110 ChunkScanLineSelector

`GenApi::Integer& ChunkScanLineSelector`

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

15.9.4.111 ChunkSelector

`GenApi::EnumerationT<ChunkSelectorEnums>& ChunkSelector`

Description: Selects which chunk data to enable or disable.

Visibility:

15.9.4.112 ChunkSequencerSetActive

`GenApi::Integer& ChunkSequencerSetActive`

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

15.9.4.113 ChunkSerialData

`GenApi::IString& ChunkSerialData`

Description: Returns the serial data that was received.

Visibility:

15.9.4.114 ChunkSerialDataLength

`GenApi::Integer& ChunkSerialDataLength`

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

15.9.4.115 ChunkSerialReceiveOverflow

`GenApi::Boolean& ChunkSerialReceiveOverflow`

Description: Returns the status of the chunk serial receive overflow.

Visibility:

15.9.4.116 ChunkSourceID

`GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID`

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

15.9.4.117 ChunkStreamChannelID

`GenApi::IInteger& ChunkStreamChannelID`

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

15.9.4.118 ChunkTimerSelector

`GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector`

Description: Selects which Timer to retrieve data from.

Visibility: Expert

15.9.4.119 ChunkTimerValue

`GenApi::IFloat& ChunkTimerValue`

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

15.9.4.120 ChunkTimestamp

`GenApi::IInteger& ChunkTimestamp`

Description: Returns the Timestamp of the image.

Visibility:

15.9.4.121 ChunkTimestampLatchValue

`GenApi::IInteger& ChunkTimestampLatchValue`

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

15.9.4.122 ChunkTransferBlockID

`GenApi::Integer& ChunkTransferBlockID`

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

15.9.4.123 ChunkTransferQueueCurrentBlockCount

`GenApi::Integer& ChunkTransferQueueCurrentBlockCount`

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

15.9.4.124 ChunkTransferStreamID

`GenApi::EnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID`

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

15.9.4.125 ChunkWidth

`GenApi::Integer& ChunkWidth`

Description: Returns the width of the image included in the payload.

Visibility:

15.9.4.126 ClConfiguration

`GenApi::EnumerationT<ClConfigurationEnums>& ClConfiguration`

Description: This [Camera](#) Link specific feature describes the configuration used by the camera.

It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera. Visibility: Beginner

15.9.4.127 ClTimeSlotsCount

`GenApi::EnumerationT<ClTimeSlotsCountEnums>& ClTimeSlotsCount`

Description: This [Camera](#) Link specific feature describes the time multiplexing of the camera link connection to transfer more than the configuration allows, in one single clock.

Visibility: Expert

15.9.4.128 ColorTransformationEnable

`GenApi::IBoolean& ColorTransformationEnable`

Description:

Enables/disables the color transform selected with ColorTransformationSelector. For RGB to YUV this is read-only. Enabling/disabling RGB to YUV can only be done by changing pixel format.

Visibility:

15.9.4.129 ColorTransformationSelector

`GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector`

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

15.9.4.130 ColorTransformationValue

`GenApi::IFloat& ColorTransformationValue`

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

15.9.4.131 ColorTransformationValueSelector

`GenApi::IEnumerationT<ColorTransformationValueSelectorEnums>& ColorTransformationValueSelector`

Description:

Selects the Gain factor or Offset of the Transformation matrix to access in the selected Color Transformation module

Visibility:

15.9.4.132 CompressionRatio

`GenApi::IFloat& CompressionRatio`

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

15.9.4.133 CompressionSaturationPriority

`GenApi::IEnumerationT<CompressionSaturationPriorityEnums>& CompressionSaturationPriority`

Description: When FrameRate is enabled, camera drops frames if datarate is saturated.

If FrameRate is disabled, camera adjusts the framerate to match the maximum achievable datarate. Visibility:

15.9.4.134 CounterDelay

`GenApi::IInteger& CounterDelay`

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

15.9.4.135 CounterDuration

`GenApi::IInteger& CounterDuration`

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

15.9.4.136 CounterEventActivation

`GenApi::IEnumerationT<CounterEventActivationEnums>& CounterEventActivation`

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

15.9.4.137 CounterEventSource

`GenApi::IEnumerationT<CounterEventSourceEnums>& CounterEventSource`

Description: Selects the event that will increment the counter Visibility:

15.9.4.138 CounterReset

```
GenApi::ICommand& CounterReset
```

Description: Does a software reset of the selected Counter and starts it.

The counter starts counting events immediately after the reset unless a Counter trigger is active. CounterReset can be used to reset the Counter independently from the CounterResetSource. To disable the counter temporarily, set CounterEventSource to Off. Visibility: Expert

15.9.4.139 CounterResetActivation

```
GenApi::IEnumerationT<CounterResetActivationEnums>& CounterResetActivation
```

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

15.9.4.140 CounterResetSource

```
GenApi::IEnumerationT<CounterResetSourceEnums>& CounterResetSource
```

Description: Selects the signal that will be the source to reset the counter.

Visibility:

15.9.4.141 CounterSelector

```
GenApi::IEnumerationT<CounterSelectorEnums>& CounterSelector
```

Description: Selects which counter to configure Visibility:

15.9.4.142 CounterStatus

```
GenApi::IEnumerationT<CounterStatusEnums>& CounterStatus
```

Description: Returns the current status of the counter.

Visibility:

15.9.4.143 CounterTriggerActivation

```
GenApi::IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation
```

Description: Selects the activation mode of the trigger to start the counter.

Visibility:

15.9.4.144 CounterTriggerSource

`GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource`

Description: Selects the source of the trigger to start the counter Visibility:

15.9.4.145 CounterValue

`GenApi::IInteger& CounterValue`

Description: Current counter value Visibility:

15.9.4.146 CounterValueAtReset

`GenApi::IInteger& CounterValueAtReset`

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

15.9.4.147 CxpConnectionSelector

`GenApi::IInteger& CxpConnectionSelector`

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

15.9.4.148 CxpConnectionTestErrorCount

`GenApi::IInteger& CxpConnectionTestErrorCount`

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

15.9.4.149 CxpConnectionTestMode

`GenApi::IEnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode`

Description: Enables the test mode for an individual physical connection of the Device.

Visibility: Expert

15.9.4.150 CxpConnectionTestPacketCount

`GenApi::Integer& CxpConnectionTestPacketCount`

Description: Reports the current count for test packets recieved by the device on the connection selected by Cxp↔ConnectionSelector.

Visibility: Expert

15.9.4.151 CxpLinkConfiguration

`GenApi::EnumerationT<CxpLinkConfigurationEnums>& CxpLinkConfiguration`

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.

In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus. Visibility: Beginner

15.9.4.152 CxpLinkConfigurationPreferred

`GenApi::EnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred`

Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.

Visibility: Expert

15.9.4.153 CxpLinkConfigurationStatus

`GenApi::EnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus`

Description: This feature indicates the current and active Link configuration used by the Device.

Visibility: Beginner

15.9.4.154 CxpPoCxpAuto

`GenApi::Command& CxpPoCxpAuto`

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

15.9.4.155 CxpPoCxpStatus

`GenApi::EnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus`

Description: Returns the Power over CoaXPress (PoCXP) status of the Device.

Visibility: Expert

15.9.4.156 CxpPoCxpTripReset

`GenApi::ICommand& CxpPoCxpTripReset`

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

15.9.4.157 CxpPoCxpTurnOff

`GenApi::ICommand& CxpPoCxpTurnOff`

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

15.9.4.158 DecimationHorizontal

`GenApi::Integer& DecimationHorizontal`

Description:

Horizontal decimation of the image. This reduces the horizontal resolution (width) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no horizontal decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

15.9.4.159 DecimationHorizontalMode

`GenApi::IEnumerationT<DecimationHorizontalModeEnums>& DecimationHorizontalMode`

Description:

The mode used to reduce the horizontal resolution when DecimationHorizontal is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

15.9.4.160 DecimationSelector

`GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector`

Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.

Visibility:

15.9.4.161 DecimationVertical

`GenApi::Integer& DecimationVertical`

Description:

Vertical decimation of the image. This reduces the vertical resolution (height) of the image by only retaining a single pixel within a window whose size is the decimation factor specified here. A value of 1 indicates that no vertical decimation is performed by the camera. This value must be 1 for binning to be active.

Visibility:

15.9.4.162 DecimationVerticalMode

`GenApi::EnumerationT<DecimationVerticalModeEnums>& DecimationVerticalMode`

Description:

The mode used to reduce the vertical resolution when DecimationVertical is used. The current implementation only supports a single decimation mode: Discard. Average should be achieved via Binning.

Visibility:

15.9.4.163 DefectCorrectionMode

`GenApi::EnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode`

Description: Controls the method used for replacing defective pixels.

Visibility:

15.9.4.164 DefectCorrectStaticEnable

`GenApi::Boolean& DefectCorrectStaticEnable`

Description: Enables/Disables table-based defective pixel correction.

Visibility:

15.9.4.165 DefectTableApply

`GenApi::ICommand& DefectTableApply`

Description: Applies the current defect table, so that any changes made affect images captured by the camera.

This writes the table to volatile memory, so changes to the table are lost if the camera loses power. To save the table to non-volatile memory, use DefectTableSave.

Visibility:

15.9.4.166 DefectTableCoordinateX

`GenApi::Integer& DefectTableCoordinateX`

Description:

Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table. Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

15.9.4.167 DefectTableCoordinateY

`GenApi::Integer& DefectTableCoordinateY`

Description:

Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table. Changes made do not take effect in captured images until the command DefectTableApply is written.

Visibility:

15.9.4.168 DefectTableFactoryRestore

`GenApi::ICommand& DefectTableFactoryRestore`

Description: Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.

This permanently overwrites any changes made to the defect table.

Visibility:

15.9.4.169 DefectTableIndex

`GenApi::Integer& DefectTableIndex`

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

15.9.4.170 DefectTablePixelCount

`GenApi::Integer& DefectTablePixelCount`

Description:

The number of defective pixel locations in the current table.

Visibility:

15.9.4.171 DefectTableSave

`GenApi::ICommand& DefectTableSave`

Description: Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.

This overwrites the existing defective pixel table.
The new table is loaded whenever the camera powers up.

Visibility:

15.9.4.172 Deinterlacing

`GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing`

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

15.9.4.173 DeviceCharacterSet

`GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet`

Description:

Character set used by the strings of the device`s bootstrap registers.

Visibility:

15.9.4.174 DeviceClockFrequency

`GenApi::IFloat& DeviceClockFrequency`

Description: Returns the frequency of the selected Clock.

Visibility: Expert

15.9.4.175 DeviceClockSelector

`GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector`

Description: Selects the clock frequency to access from the device.

Visibility: Expert

15.9.4.176 DeviceConnectionSelector

`GenApi::IInteger& DeviceConnectionSelector`

Description: Selects which Connection of the device to control.

Visibility: Beginner

15.9.4.177 DeviceConnectionSpeed

`GenApi::IInteger& DeviceConnectionSpeed`

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

15.9.4.178 DeviceConnectionStatus

`GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus`

Description: Indicates the status of the specified Connection.

Visibility: Expert

15.9.4.179 DeviceEventChannelCount

`GenApi::IInteger& DeviceEventChannelCount`

Description:

Indicates the number of event channels supported by the device.

Visibility:

15.9.4.180 DeviceFamilyName

`GenApi::IString& DeviceFamilyName`

Description: Identifier of the product family of the device.

Visibility: Beginner

15.9.4.181 DeviceFeaturePersistenceEnd

`GenApi::ICommand& DeviceFeaturePersistenceEnd`

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

15.9.4.182 DeviceFeaturePersistenceStart

`GenApi::ICommand& DeviceFeaturePersistenceStart`

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

15.9.4.183 DeviceFirmwareVersion

`GenApi::IString& DeviceFirmwareVersion`

Description: Version of the firmware on the device.

Visibility:

15.9.4.184 DeviceGenCPVersionMajor

`GenApi::IInteger& DeviceGenCPVersionMajor`

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

15.9.4.185 DeviceGenCPVersionMinor

`GenApi::IInteger& DeviceGenCPVersionMinor`

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

15.9.4.186 DeviceID

`GenApi::IString& DeviceID`

Description: Device identifier (serial number).

Visibility:

15.9.4.187 DeviceIndicatorMode

`GenApi::IEnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode`

Description: Controls the LED behaviour: Inactive (off), Active (current status), or Error Status (off unless an error occurs).

Visibility:

15.9.4.188 DeviceLinkBandwidthReserve

`GenApi::IFloat& DeviceLinkBandwidthReserve`

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

15.9.4.189 DeviceLinkCommandTimeout

`GenApi::IFloat& DeviceLinkCommandTimeout`

Description: Indicates the command timeout of the specified Link.

This corresponds to the maximum response time of the device for a command sent on that link. Visibility: Guru

15.9.4.190 DeviceLinkConnectionCount

`GenApi::IInteger& DeviceLinkConnectionCount`

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

15.9.4.191 DeviceLinkCurrentThroughput

`GenApi::IInteger& DeviceLinkCurrentThroughput`

Description: Current bandwidth of streamed data.

Visibility:

15.9.4.192 DeviceLinkHeartbeatMode

`GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode`

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

15.9.4.193 DeviceLinkHeartbeatTimeout

`GenApi::IFloat& DeviceLinkHeartbeatTimeout`

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

15.9.4.194 DeviceLinkSelector

`GenApi::IInteger& DeviceLinkSelector`

Description: Selects which Link of the device to control.

Visibility: Beginner

15.9.4.195 DeviceLinkSpeed

`GenApi::Integer& DeviceLinkSpeed`

Description:

Indicates the speed of transmission negotiated on the specified Link. (Bps)

Visibility:

15.9.4.196 DeviceLinkThroughputLimit

`GenApi::Integer& DeviceLinkThroughputLimit`

Description:

Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link. If necessary, delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth.

Visibility:

15.9.4.197 DeviceLinkThroughputLimitMode

`GenApi::EnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode`

Description: Controls if the DeviceLinkThroughputLimit is active.

When disabled, lower level TL specific features are expected to control the throughput. When enabled, Device↔LinkThroughputLimit controls the overall throughput. Visibility: Expert

15.9.4.198 DeviceManifestEntrySelector

`GenApi::Integer& DeviceManifestEntrySelector`

Description: Selects the manifest entry to reference.

Visibility: Guru

15.9.4.199 DeviceManifestPrimaryURL

`GenApi::String& DeviceManifestPrimaryURL`

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

15.9.4.200 DeviceManifestSchemaMajorVersion

`GenApi::Integer& DeviceManifestSchemaMajorVersion`

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

15.9.4.201 DeviceManifestSchemaMinorVersion

`GenApi::Integer& DeviceManifestSchemaMinorVersion`

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

15.9.4.202 DeviceManifestSecondaryURL

`GenApi::String& DeviceManifestSecondaryURL`

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

15.9.4.203 DeviceManifestXMLMajorVersion

`GenApi::Integer& DeviceManifestXMLMajorVersion`

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.9.4.204 DeviceManifestXMLMinorVersion

`GenApi::Integer& DeviceManifestXMLMinorVersion`

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.9.4.205 DeviceManifestXMLSubMinorVersion

`GenApi::Integer& DeviceManifestXMLSubMinorVersion`

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

15.9.4.206 DeviceManufacturerInfo

`GenApi::IString& DeviceManufacturerInfo`

Description: Manufacturer information about the device.

Visibility:

15.9.4.207 DeviceMaxThroughput

`GenApi::IInteger& DeviceMaxThroughput`

Description:

Maximum bandwidth of the data that can be streamed out of the device. This can be used to estimate if the physical connection(s) can sustain transfer of free-running images from the camera at its maximum speed.

Visibility:

15.9.4.208 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Model of the device.

Visibility:

15.9.4.209 DevicePowerSupplySelector

`GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector`

Description:

Selects the power supply source to control or read.

Visibility:

15.9.4.210 DeviceRegistersCheck

`GenApi::ICommand& DeviceRegistersCheck`

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

15.9.4.211 DeviceRegistersEndianness

`GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness`

Description: Endianness of the registers of the device.

Visibility:

15.9.4.212 DeviceRegistersStreamingEnd

`GenApi::ICommand& DeviceRegistersStreamingEnd`

Description: Announce the end of registers streaming.

This will do a register set validation for consistency and activate it. This will also update the DeviceRegistersValid flag. Visibility: Guru

15.9.4.213 DeviceRegistersStreamingStart

`GenApi::ICommand& DeviceRegistersStreamingStart`

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

15.9.4.214 DeviceRegistersValid

`GenApi::IBoolean& DeviceRegistersValid`

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

15.9.4.215 DeviceReset

`GenApi::ICommand& DeviceReset`

Description: This is a command that immediately resets and reboots the device.

Visibility:

15.9.4.216 DeviceScanType

`GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType`

Description: Scan type of the sensor of the device.

Visibility:

15.9.4.217 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description:

Device's serial number. This string is a unique identifier of the device.

Visibility:

15.9.4.218 DeviceSerialPortBaudRate

`GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate`

Description: This feature controls the baud rate used by the selected serial port.

Visibility: Expert

15.9.4.219 DeviceSerialPortSelector

`GenApi::IEnumerationT<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector`

Description: Selects which serial port of the device to control.

Visibility: Expert

15.9.4.220 DeviceSFNCVersionMajor

`GenApi::IInteger& DeviceSFNCVersionMajor`

Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.9.4.221 DeviceSFNCVersionMinor

`GenApi::IInteger& DeviceSFNCVersionMinor`

Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.9.4.222 DeviceSFNCVersionSubMinor

`GenApi::Integer& DeviceSFNCVersionSubMinor`

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).

Visibility: Beginner

15.9.4.223 DeviceStreamChannelCount

`GenApi::Integer& DeviceStreamChannelCount`

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

15.9.4.224 DeviceStreamChannelEndianness

`GenApi::EnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness`

Description: Endianness of multi-byte pixel data for this stream.

Visibility: Guru

15.9.4.225 DeviceStreamChannelLink

`GenApi::Integer& DeviceStreamChannelLink`

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

15.9.4.226 DeviceStreamChannelPacketSize

`GenApi::Integer& DeviceStreamChannelPacketSize`

Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.

Visibility: Expert

15.9.4.227 DeviceStreamChannelSelector

`GenApi::Integer& DeviceStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility: Expert

15.9.4.228 DeviceStreamChannelType

`GenApi::EnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType`

Description: Reports the type of the stream channel.

Visibility: Guru

15.9.4.229 DeviceTapGeometry

`GenApi::EnumerationT<DeviceTapGeometryEnums>& DeviceTapGeometry`

Description: This device tap geometry feature describes the geometrical properties characterizing the taps of a camera as presented at the output of the device.

Visibility: Expert

15.9.4.230 DeviceTemperature

`GenApi::IFloat& DeviceTemperature`

Description: Device temperature in degrees Celsius (C).

Visibility:

15.9.4.231 DeviceTemperatureSelector

`GenApi::EnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector`

Description:

Selects the location within the device, where the temperature will be measured.

Visibility:

15.9.4.232 DeviceTLType

`GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType`

Description: Transport Layer type of the device.

Visibility:

15.9.4.233 DeviceTLVersionMajor

`GenApi::IInteger& DeviceTLVersionMajor`

Description:

Major version of the Transport Layer of the device.

Visibility:

15.9.4.234 DeviceTLVersionMinor

`GenApi::IInteger& DeviceTLVersionMinor`

Description:

Minor version of the Transport Layer of the device.

Visibility:

15.9.4.235 DeviceTLVersionSubMinor

`GenApi::IInteger& DeviceTLVersionSubMinor`

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

15.9.4.236 DeviceType

`GenApi::IEnumerationT<DeviceTypeEnums>& DeviceType`

Description: Returns the device type.

Visibility: Guru

15.9.4.237 DeviceUptime

`GenApi::Integer& DeviceUptime`

Description: Total time since the device was powered up in seconds.

Visibility:

15.9.4.238 DeviceUserID

`GenApi::IString& DeviceUserID`

Description: User-programmable device identifier.

Visibility:

15.9.4.239 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the manufacturer of the device.

Visibility:

15.9.4.240 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility:

15.9.4.241 EncoderDivider

`GenApi::Integer& EncoderDivider`

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

15.9.4.242 EncoderMode

`GenApi::EnumerationT<EncoderModeEnums>& EncoderMode`

Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.

Visibility: Expert

15.9.4.243 EncoderOutputMode

`GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode`

Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.

Visibility: Expert

15.9.4.244 EncoderReset

`GenApi::ICommand& EncoderReset`

Description: Does a software reset of the selected Encoder and starts it.

The Encoder starts counting events immediately after the reset. EncoderReset can be used to reset the Encoder independently from the EncoderResetSource. Visibility: Expert

15.9.4.245 EncoderResetActivation

`GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation`

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

15.9.4.246 EncoderResetSource

`GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource`

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

15.9.4.247 EncoderSelector

`GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector`

Description: Selects which Encoder to configure.

Visibility: Expert

15.9.4.248 EncoderSourceA

`GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA`

Description: Selects the signal which will be the source of the A input of the Encoder.

Visibility: Expert

15.9.4.249 EncoderSourceB

`GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB`

Description: Selects the signal which will be the source of the B input of the Encoder.

Visibility: Expert

15.9.4.250 EncoderStatus

`GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus`

Description: Returns the motion status of the encoder.

Visibility: Expert

15.9.4.251 EncoderTimeout

`GenApi::IFloat& EncoderTimeout`

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

15.9.4.252 EncoderValue

`GenApi::IInteger& EncoderValue`

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

15.9.4.253 EncoderValueAtReset

`GenApi::IInteger& EncoderValueAtReset`

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

15.9.4.254 EnumerationCount

`GenApi::IInteger& EnumerationCount`

Description: Number of enumerations since uptime.

Visibility:

15.9.4.255 EventAcquisitionEnd

`GenApi::Integer& EventAcquisitionEnd`

Description: Returns the unique Identifier of the Acquisition End type of Event.

Visibility: Expert

15.9.4.256 EventAcquisitionEndFrameID

`GenApi::Integer& EventAcquisitionEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.

Visibility: Expert

15.9.4.257 EventAcquisitionEndTimestamp

`GenApi::Integer& EventAcquisitionEndTimestamp`

Description: Returns the Timestamp of the Acquisition End Event.

Visibility: Expert

15.9.4.258 EventAcquisitionError

`GenApi::Integer& EventAcquisitionError`

Description: Returns the unique Identifier of the Acquisition Error type of Event.

Visibility: Expert

15.9.4.259 EventAcquisitionErrorFrameID

`GenApi::Integer& EventAcquisitionErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.

Visibility: Expert

15.9.4.260 EventAcquisitionErrorTimestamp

`GenApi::Integer& EventAcquisitionErrorTimestamp`

Description: Returns the Timestamp of the Acquisition Error Event.

Visibility: Expert

15.9.4.261 EventAcquisitionStart

`GenApi::Integer& EventAcquisitionStart`

Description: Returns the unique Identifier of the Acquisition Start type of Event.

Visibility: Expert

15.9.4.262 EventAcquisitionStartFrameID

`GenApi::Integer& EventAcquisitionStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.

Visibility: Expert

15.9.4.263 EventAcquisitionStartTimestamp

`GenApi::Integer& EventAcquisitionStartTimestamp`

Description: Returns the Timestamp of the Acquisition Start Event.

Visibility: Expert

15.9.4.264 EventAcquisitionTransferEnd

`GenApi::Integer& EventAcquisitionTransferEnd`

Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.

Visibility: Expert

15.9.4.265 EventAcquisitionTransferEndFrameID

`GenApi::Integer& EventAcquisitionTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.

Visibility: Expert

15.9.4.266 EventAcquisitionTransferEndTimestamp

`GenApi::Integer& EventAcquisitionTransferEndTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer End Event.

Visibility: Expert

15.9.4.267 EventAcquisitionTransferStart

`GenApi::Integer& EventAcquisitionTransferStart`

Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.

Visibility: Expert

15.9.4.268 EventAcquisitionTransferStartFrameID

`GenApi::Integer& EventAcquisitionTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.

Visibility: Expert

15.9.4.269 EventAcquisitionTransferStartTimestamp

`GenApi::Integer& EventAcquisitionTransferStartTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer Start Event.

Visibility: Expert

15.9.4.270 EventAcquisitionTrigger

`GenApi::Integer& EventAcquisitionTrigger`

Description: Returns the unique Identifier of the Acquisition Trigger type of Event.

Visibility: Expert

15.9.4.271 EventAcquisitionTriggerFrameID

`GenApi::Integer& EventAcquisitionTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.

Visibility: Expert

15.9.4.272 EventAcquisitionTriggerTimestamp

`GenApi::Integer& EventAcquisitionTriggerTimestamp`

Description: Returns the Timestamp of the Acquisition Trigger Event.

Visibility: Expert

15.9.4.273 EventActionLate

`GenApi::Integer& EventActionLate`

Description: Returns the unique Identifier of the Action Late type of Event.

Visibility: Expert

15.9.4.274 EventActionLateFrameID

`GenApi::Integer& EventActionLateFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.

Visibility: Expert

15.9.4.275 EventActionLateTimestamp

`GenApi::Integer& EventActionLateTimestamp`

Description: Returns the Timestamp of the Action Late Event.

Visibility: Expert

15.9.4.276 EventCounter0End

`GenApi::Integer& EventCounter0End`

Description: Returns the unique Identifier of the Counter 0 End type of Event.

Visibility: Expert

15.9.4.277 EventCounter0EndFrameID

`GenApi::Integer& EventCounter0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.

Visibility: Expert

15.9.4.278 EventCounter0EndTimestamp

`GenApi::Integer& EventCounter0EndTimestamp`

Description: Returns the Timestamp of the Counter 0 End Event.

Visibility: Expert

15.9.4.279 EventCounter0Start

`GenApi::Integer& EventCounter0Start`

Description: Returns the unique Identifier of the Counter 0 Start type of Event.

Visibility: Expert

15.9.4.280 EventCounter0StartFrameID

`GenApi::Integer& EventCounter0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.

Visibility: Expert

15.9.4.281 EventCounter0StartTimestamp

`GenApi::Integer& EventCounter0StartTimestamp`

Description: Returns the Timestamp of the Counter 0 Start Event.

Visibility: Expert

15.9.4.282 EventCounter1End

`GenApi::Integer& EventCounter1End`

Description: Returns the unique Identifier of the Counter 1 End type of Event.

Visibility: Expert

15.9.4.283 EventCounter1EndFrameID

`GenApi::Integer& EventCounter1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.

Visibility: Expert

15.9.4.284 EventCounter1EndTimestamp

`GenApi::Integer& EventCounter1EndTimestamp`

Description: Returns the Timestamp of the Counter 1 End Event.

Visibility: Expert

15.9.4.285 EventCounter1Start

`GenApi::Integer& EventCounter1Start`

Description: Returns the unique Identifier of the Counter 1 Start type of Event.

Visibility: Expert

15.9.4.286 EventCounter1StartFrameID

`GenApi::Integer& EventCounter1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.

Visibility: Expert

15.9.4.287 EventCounter1StartTimestamp

`GenApi::Integer& EventCounter1StartTimestamp`

Description: Returns the Timestamp of the Counter 1 Start Event.

Visibility: Expert

15.9.4.288 EventEncoder0Restarted

`GenApi::Integer& EventEncoder0Restarted`

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.

Visibility: Expert

15.9.4.289 EventEncoder0RestartedFrameID

`GenApi::Integer& EventEncoder0RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.

Visibility: Expert

15.9.4.290 EventEncoder0RestartedTimestamp

`GenApi::Integer& EventEncoder0RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Restarted Event.

Visibility: Expert

15.9.4.291 EventEncoder0Stopped

`GenApi::Integer& EventEncoder0Stopped`

Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.

Visibility: Expert

15.9.4.292 EventEncoder0StoppedFrameID

`GenApi::Integer& EventEncoder0StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.

Visibility: Expert

15.9.4.293 EventEncoder0StoppedTimestamp

`GenApi::Integer& EventEncoder0StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Stopped Event.

Visibility: Expert

15.9.4.294 EventEncoder1Restarted

`GenApi::Integer& EventEncoder1Restarted`

Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.

Visibility: Expert

15.9.4.295 EventEncoder1RestartedFrameID

`GenApi::Integer& EventEncoder1RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.

Visibility: Expert

15.9.4.296 EventEncoder1RestartedTimestamp

`GenApi::Integer& EventEncoder1RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Restarted Event.

Visibility: Expert

15.9.4.297 EventEncoder1Stopped

`GenApi::Integer& EventEncoder1Stopped`

Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.

Visibility: Expert

15.9.4.298 EventEncoder1StoppedFrameID

`GenApi::Integer& EventEncoder1StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.

Visibility: Expert

15.9.4.299 EventEncoder1StoppedTimestamp

`GenApi::Integer& EventEncoder1StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Stopped Event.

Visibility: Expert

15.9.4.300 EventError

`GenApi::Integer& EventError`

Description: Returns the unique identifier of the Error type of Event.

Visibility:

15.9.4.301 EventErrorCode

`GenApi::Integer& EventErrorCode`

Description: Returns the error code for the error that happened Visibility:

15.9.4.302 EventErrorFrameID

`GenApi::Integer& EventErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.

Visibility:

15.9.4.303 EventErrorTimestamp

`GenApi::Integer& EventErrorTimestamp`

Description: Returns the Timestamp of the Error Event.

Visibility:

15.9.4.304 EventExposureEnd

`GenApi::Integer& EventExposureEnd`

Description: Returns the unique identifier of the Exposure End type of Event.

Visibility:

15.9.4.305 EventExposureEndFrameID

`GenApi::Integer& EventExposureEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.

Visibility:

15.9.4.306 EventExposureEndTimestamp

`GenApi::Integer& EventExposureEndTimestamp`

Description: Returns the Timestamp of the Exposure End Event.

Visibility:

15.9.4.307 EventExposureStart

`GenApi::Integer& EventExposureStart`

Description: Returns the unique Identifier of the Exposure Start type of Event.

Visibility: Expert

15.9.4.308 EventExposureStartFrameID

`GenApi::Integer& EventExposureStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.

Visibility: Expert

15.9.4.309 EventExposureStartTimestamp

`GenApi::Integer& EventExposureStartTimestamp`

Description: Returns the Timestamp of the Exposure Start Event.

Visibility: Expert

15.9.4.310 EventFrameBurstEnd

`GenApi::Integer& EventFrameBurstEnd`

Description: Returns the unique Identifier of the Frame Burst End type of Event.

Visibility: Expert

15.9.4.311 EventFrameBurstEndFrameID

`GenApi::Integer& EventFrameBurstEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.

Visibility: Expert

15.9.4.312 EventFrameBurstEndTimestamp

`GenApi::Integer& EventFrameBurstEndTimestamp`

Description: Returns the Timestamp of the Frame Burst End Event.

Visibility: Expert

15.9.4.313 EventFrameBurstStart

`GenApi::Integer& EventFrameBurstStart`

Description: Returns the unique Identifier of the Frame Burst Start type of Event.

Visibility: Expert

15.9.4.314 EventFrameBurstStartFrameID

`GenApi::Integer& EventFrameBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.

Visibility: Expert

15.9.4.315 EventFrameBurstStartTimestamp

`GenApi::Integer& EventFrameBurstStartTimestamp`

Description: Returns the Timestamp of the Frame Burst Start Event.

Visibility: Expert

15.9.4.316 EventFrameEnd

`GenApi::Integer& EventFrameEnd`

Description: Returns the unique Identifier of the Frame End type of Event.

Visibility: Expert

15.9.4.317 EventFrameEndFrameID

`GenApi::Integer& EventFrameEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.

Visibility: Expert

15.9.4.318 EventFrameEndTimestamp

`GenApi::Integer& EventFrameEndTimestamp`

Description: Returns the Timestamp of the Frame End Event.

Visibility: Expert

15.9.4.319 EventFrameStart

`GenApi::Integer& EventFrameStart`

Description: Returns the unique Identifier of the Frame Start type of Event.

Visibility: Expert

15.9.4.320 EventFrameStartFrameID

`GenApi::Integer& EventFrameStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.

Visibility: Expert

15.9.4.321 EventFrameStartTimestamp

`GenApi::Integer& EventFrameStartTimestamp`

Description: Returns the Timestamp of the Frame Start Event.

Visibility: Expert

15.9.4.322 EventFrameTransferEnd

`GenApi::Integer& EventFrameTransferEnd`

Description: Returns the unique Identifier of the Frame Transfer End type of Event.

Visibility: Expert

15.9.4.323 EventFrameTransferEndFrameID

`GenApi::Integer& EventFrameTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.

Visibility: Expert

15.9.4.324 EventFrameTransferEndTimestamp

`GenApi::Integer& EventFrameTransferEndTimestamp`

Description: Returns the Timestamp of the Frame Transfer End Event.

Visibility: Expert

15.9.4.325 EventFrameTransferStart

`GenApi::Integer& EventFrameTransferStart`

Description: Returns the unique Identifier of the Frame Transfer Start type of Event.

Visibility: Expert

15.9.4.326 EventFrameTransferStartFrameID

`GenApi::Integer& EventFrameTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.

Visibility: Expert

15.9.4.327 EventFrameTransferStartTimestamp

`GenApi::Integer& EventFrameTransferStartTimestamp`

Description: Returns the Timestamp of the Frame Transfer Start Event.

Visibility: Expert

15.9.4.328 EventFrameTrigger

`GenApi::Integer& EventFrameTrigger`

Description: Returns the unique Identifier of the FrameTrigger type of Event.

It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type event received. Visibility: Expert

15.9.4.329 EventFrameTriggerFrameID

`GenApi::Integer& EventFrameTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Visibility: Expert

15.9.4.330 EventFrameTriggerTimestamp

`GenApi::Integer& EventFrameTriggerTimestamp`

Description: Returns the Timestamp of the FrameTrigger Event.

It can be used to determine precisely when the event occurred. Visibility: Expert

15.9.4.331 EventLine0AnyEdge

`GenApi::Integer& EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.

Visibility: Expert

15.9.4.332 EventLine0AnyEdgeFrameID

`GenApi::Integer& EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.

Visibility: Expert

15.9.4.333 EventLine0AnyEdgeTimestamp

`GenApi::Integer& EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge Event.

Visibility: Expert

15.9.4.334 EventLine0FallingEdge

`GenApi::Integer& EventLine0FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.

Visibility: Expert

15.9.4.335 EventLine0FallingEdgeFrameID

`GenApi::Integer& EventLine0FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.

Visibility: Expert

15.9.4.336 EventLine0FallingEdgeTimestamp

`GenApi::Integer& EventLine0FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Falling Edge Event.

Visibility: Expert

15.9.4.337 EventLine0RisingEdge

`GenApi::Integer& EventLine0RisingEdge`

Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.

Visibility: Expert

15.9.4.338 EventLine0RisingEdgeFrameID

`GenApi::Integer& EventLine0RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.

Visibility: Expert

15.9.4.339 EventLine0RisingEdgeTimestamp

`GenApi::Integer& EventLine0RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Rising Edge Event.

Visibility: Expert

15.9.4.340 EventLine1AnyEdge

`GenApi::Integer& EventLine1AnyEdge`

Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.

Visibility: Expert

15.9.4.341 EventLine1AnyEdgeFrameID

`GenApi::Integer& EventLine1AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.

Visibility: Expert

15.9.4.342 EventLine1AnyEdgeTimestamp

`GenApi::Integer& EventLine1AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Any Edge Event.

Visibility: Expert

15.9.4.343 EventLine1FallingEdge

`GenApi::Integer& EventLine1FallingEdge`

Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.

Visibility: Expert

15.9.4.344 EventLine1FallingEdgeFrameID

`GenApi::Integer& EventLine1FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.

Visibility: Expert

15.9.4.345 EventLine1FallingEdgeTimestamp

`GenApi::Integer& EventLine1FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Falling Edge Event.

Visibility: Expert

15.9.4.346 EventLine1RisingEdge

`GenApi::Integer& EventLine1RisingEdge`

Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.

Visibility: Expert

15.9.4.347 EventLine1RisingEdgeFrameID

`GenApi::Integer& EventLine1RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.

Visibility: Expert

15.9.4.348 EventLine1RisingEdgeTimestamp

`GenApi::Integer& EventLine1RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Rising Edge Event.

Visibility: Expert

15.9.4.349 EventLinkSpeedChange

`GenApi::Integer& EventLinkSpeedChange`

Description: Returns the unique Identifier of the Link Speed Change type of Event.

Visibility: Expert

15.9.4.350 EventLinkSpeedChangeFrameID

`GenApi::Integer& EventLinkSpeedChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.

Visibility: Expert

15.9.4.351 EventLinkSpeedChangeTimestamp

`GenApi::Integer& EventLinkSpeedChangeTimestamp`

Description: Returns the Timestamp of the Link Speed Change Event.

Visibility: Expert

15.9.4.352 EventLinkTrigger0

`GenApi::Integer& EventLinkTrigger0`

Description: Returns the unique Identifier of the Link Trigger 0 type of Event.

Visibility: Expert

15.9.4.353 EventLinkTrigger0FrameID

`GenApi::Integer& EventLinkTrigger0FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.

Visibility: Expert

15.9.4.354 EventLinkTrigger0Timestamp

`GenApi::Integer& EventLinkTrigger0Timestamp`

Description: Returns the Timestamp of the Link Trigger 0 Event.

Visibility: Expert

15.9.4.355 EventLinkTrigger1

`GenApi::Integer& EventLinkTrigger1`

Description: Returns the unique Identifier of the Link Trigger 1 type of Event.

Visibility: Expert

15.9.4.356 EventLinkTrigger1FrameID

`GenApi::Integer& EventLinkTrigger1FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.

Visibility: Expert

15.9.4.357 EventLinkTrigger1Timestamp

`GenApi::Integer& EventLinkTrigger1Timestamp`

Description: Returns the Timestamp of the Link Trigger 1 Event.

Visibility: Expert

15.9.4.358 EventNotification

`GenApi::EnumerationT<EventNotificationEnums>& EventNotification`

Description: Enables/Disables the selected event.

Visibility:

15.9.4.359 EventSelector

`GenApi::EnumerationT<EventSelectorEnums>& EventSelector`

Description: Selects which Event to enable or disable.

Visibility:

15.9.4.360 EventSequencerSetChange

`GenApi::Integer& EventSequencerSetChange`

Description: Returns the unique Identifier of the Sequencer Set Change type of Event.

Visibility: Expert

15.9.4.361 EventSequencerSetChangeFrameID

`GenApi::Integer& EventSequencerSetChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.

Visibility: Expert

15.9.4.362 EventSequencerSetChangeTimestamp

`GenApi::Integer& EventSequencerSetChangeTimestamp`

Description: Returns the Timestamp of the Sequencer Set Change Event.

Visibility: Expert

15.9.4.363 EventSerialData

`GenApi::IString& EventSerialData`

Description: Returns the serial data that was received.

Visibility:

15.9.4.364 EventSerialDataLength

`GenApi::IInteger& EventSerialDataLength`

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

15.9.4.365 EventSerialPortReceive

`GenApi::IInteger& EventSerialPortReceive`

Description: Returns the unique identifier of the Serial Port Receive type of Event.

Visibility:

15.9.4.366 EventSerialPortReceiveTimestamp

`GenApi::IInteger& EventSerialPortReceiveTimestamp`

Description: Returns the Timestamp of the Serial Port Receive Event.

Visibility:

15.9.4.367 EventSerialReceiveOverflow

`GenApi::IBoolean& EventSerialReceiveOverflow`

Description: Returns the status of the event serial receive overflow.

Visibility:

15.9.4.368 EventStream0TransferBlockEnd

`GenApi::IInteger& EventStream0TransferBlockEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.

Visibility: Expert

15.9.4.369 EventStream0TransferBlockEndFrameID

`GenApi::Integer& EventStream0TransferBlockEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.

Visibility: Expert

15.9.4.370 EventStream0TransferBlockEndTimestamp

`GenApi::Integer& EventStream0TransferBlockEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

Visibility: Expert

15.9.4.371 EventStream0TransferBlockStart

`GenApi::Integer& EventStream0TransferBlockStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.

Visibility: Expert

15.9.4.372 EventStream0TransferBlockStartFrameID

`GenApi::Integer& EventStream0TransferBlockStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.

Visibility: Expert

15.9.4.373 EventStream0TransferBlockStartTimestamp

`GenApi::Integer& EventStream0TransferBlockStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.

Visibility: Expert

15.9.4.374 EventStream0TransferBlockTrigger

`GenApi::Integer& EventStream0TransferBlockTrigger`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.

Visibility: Expert

15.9.4.375 EventStream0TransferBlockTriggerFrameID

`GenApi::Integer& EventStream0TransferBlockTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

15.9.4.376 EventStream0TransferBlockTriggerTimestamp

`GenApi::Integer& EventStream0TransferBlockTriggerTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

15.9.4.377 EventStream0TransferBurstEnd

`GenApi::Integer& EventStream0TransferBurstEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.

Visibility: Expert

15.9.4.378 EventStream0TransferBurstEndFrameID

`GenApi::Integer& EventStream0TransferBurstEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.

Visibility: Expert

15.9.4.379 EventStream0TransferBurstEndTimestamp

`GenApi::Integer& EventStream0TransferBurstEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.

Visibility: Expert

15.9.4.380 EventStream0TransferBurstStart

`GenApi::Integer& EventStream0TransferBurstStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.

Visibility: Expert

15.9.4.381 EventStream0TransferBurstStartFrameID

`GenApi::Integer& EventStream0TransferBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.

Visibility: Expert

15.9.4.382 EventStream0TransferBurstStartTimestamp

`GenApi::Integer& EventStream0TransferBurstStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.

Visibility: Expert

15.9.4.383 EventStream0TransferEnd

`GenApi::Integer& EventStream0TransferEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.

Visibility: Expert

15.9.4.384 EventStream0TransferEndFrameID

`GenApi::Integer& EventStream0TransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.

Visibility: Expert

15.9.4.385 EventStream0TransferEndTimestamp

`GenApi::Integer& EventStream0TransferEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer End Event.

Visibility: Expert

15.9.4.386 EventStream0TransferOverflow

`GenApi::Integer& EventStream0TransferOverflow`

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.

Visibility: Expert

15.9.4.387 EventStream0TransferOverflowFrameID

`GenApi::Integer& EventStream0TransferOverflowFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.

Visibility: Expert

15.9.4.388 EventStream0TransferOverflowTimestamp

`GenApi::Integer& EventStream0TransferOverflowTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.

Visibility: Expert

15.9.4.389 EventStream0TransferPause

`GenApi::Integer& EventStream0TransferPause`

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.

Visibility: Expert

15.9.4.390 EventStream0TransferPauseFrameID

`GenApi::Integer& EventStream0TransferPauseFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.

Visibility: Expert

15.9.4.391 EventStream0TransferPauseTimestamp

`GenApi::Integer& EventStream0TransferPauseTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.

Visibility: Expert

15.9.4.392 EventStream0TransferResume

`GenApi::Integer& EventStream0TransferResume`

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.

Visibility: Expert

15.9.4.393 EventStream0TransferResumeFrameID

`GenApi::Integer& EventStream0TransferResumeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.

Visibility: Expert

15.9.4.394 EventStream0TransferResumeTimestamp

`GenApi::Integer& EventStream0TransferResumeTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.

Visibility: Expert

15.9.4.395 EventStream0TransferStart

`GenApi::Integer& EventStream0TransferStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.

Visibility: Expert

15.9.4.396 EventStream0TransferStartFrameID

`GenApi::Integer& EventStream0TransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.

Visibility: Expert

15.9.4.397 EventStream0TransferStartTimestamp

`GenApi::Integer& EventStream0TransferStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Start Event.

Visibility: Expert

15.9.4.398 EventTest

`GenApi::Integer& EventTest`

Description: Returns the unique identifier of the Test type of Event.

Visibility:

15.9.4.399 EventTestTimestamp

`GenApi::Integer& EventTestTimestamp`

Description: Returns the Timestamp of the Test Event.

Visibility:

15.9.4.400 EventTimer0End

`GenApi::Integer& EventTimer0End`

Description: Returns the unique Identifier of the Timer 0 End type of Event.

Visibility: Expert

15.9.4.401 EventTimer0EndFrameID

`GenApi::Integer& EventTimer0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.

Visibility: Expert

15.9.4.402 EventTimer0EndTimestamp

`GenApi::Integer& EventTimer0EndTimestamp`

Description: Returns the Timestamp of the Timer 0 End Event.

Visibility: Expert

15.9.4.403 EventTimer0Start

`GenApi::Integer& EventTimer0Start`

Description: Returns the unique Identifier of the Timer 0 Start type of Event.

Visibility: Expert

15.9.4.404 EventTimer0StartFrameID

`GenApi::Integer& EventTimer0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.

Visibility: Expert

15.9.4.405 EventTimer0StartTimestamp

`GenApi::Integer& EventTimer0StartTimestamp`

Description: Returns the Timestamp of the Timer 0 Start Event.

Visibility: Expert

15.9.4.406 EventTimer1End

`GenApi::Integer& EventTimer1End`

Description: Returns the unique Identifier of the Timer 1 End type of Event.

Visibility: Expert

15.9.4.407 EventTimer1EndFrameID

`GenApi::Integer& EventTimer1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.

Visibility: Expert

15.9.4.408 EventTimer1EndTimestamp

`GenApi::Integer& EventTimer1EndTimestamp`

Description: Returns the Timestamp of the Timer 1 End Event.

Visibility: Expert

15.9.4.409 EventTimer1Start

`GenApi::Integer& EventTimer1Start`

Description: Returns the unique Identifier of the Timer 1 Start type of Event.

Visibility: Expert

15.9.4.410 EventTimer1StartFrameID

`GenApi::Integer& EventTimer1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.

Visibility: Expert

15.9.4.411 EventTimer1StartTimestamp

`GenApi::Integer& EventTimer1StartTimestamp`

Description: Returns the Timestamp of the Timer 1 Start Event.

Visibility: Expert

15.9.4.412 ExposureActiveMode

`GenApi::EnumerationT<ExposureActiveModeEnums>& ExposureActiveMode`

Description: Control sensor active exposure mode.

Visibility:

15.9.4.413 ExposureAuto

`GenApi::EnumerationT<ExposureAutoEnums>& ExposureAuto`

Description: Sets the automatic exposure mode Visibility:

15.9.4.414 ExposureMode

`GenApi::EnumerationT<ExposureModeEnums>& ExposureMode`

Description:

Sets the operation mode of the Exposure.

Visibility:

15.9.4.415 ExposureTime

`GenApi::Float& ExposureTime`

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

15.9.4.416 ExposureTimeMode

`GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode`

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

15.9.4.417 ExposureTimeSelector

`GenApi::IEnumerationT<ExposureTimeSelectorEnums>& ExposureTimeSelector`

Description: Selects which exposure time is controlled by the ExposureTime feature.

This allows for independent control over the exposure components. Visibility: Beginner

15.9.4.418 FactoryReset

`GenApi::ICommand& FactoryReset`

Description: Returns all user tables to factory default Visibility:

15.9.4.419 FileAccessBuffer

`GenApi::IRegister& FileAccessBuffer`

Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.

Visibility: Guru

15.9.4.420 FileAccessLength

`GenApi::IInteger& FileAccessLength`

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

15.9.4.421 FileAccessOffset

`GenApi::IInteger& FileAccessOffset`

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

15.9.4.422 FileOpenMode

`GenApi::IEnumerationT<FileOpenModeEnums>& FileOpenMode`

Description:

The mode of the file when it is opened. The file can be opened for reading, writing or both. This must be set before opening the file.

Visibility:

15.9.4.423 FileOperationExecute

`GenApi::ICommand& FileOperationExecute`

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

15.9.4.424 FileOperationResult

`GenApi::IInteger& FileOperationResult`

Description: Represents the file operation result.

For Read or Write operations, the number of successfully read/written bytes is returned. Visibility:

15.9.4.425 FileOperationSelector

`GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector`

Description:

Sets operation to execute on the selected file when the execute command is given.

Visibility:

15.9.4.426 FileOperationStatus

`GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus`

Description: Represents the file operation execution status.

Visibility:

15.9.4.427 FileSelector

`GenApi::IEnumerationT<FileSelectorEnums>& FileSelector`

Description:

Selects which file is being operated on. This must be set before performing any file operations.

Visibility:

15.9.4.428 FileSize

`GenApi::IInteger& FileSize`

Description: Represents the size of the selected file in bytes.

Visibility:

15.9.4.429 Gain

`GenApi::IFloat& Gain`

Description:

Controls the amplification of the video signal in dB.

Visibility:

15.9.4.430 GainAuto

`GenApi::IEnumerationT<GainAutoEnums>& GainAuto`

Description:

Sets the automatic gain mode. Set to Off for manual control. Set to Once for a single automatic adjustment then return to Off. Set to Continuous for constant adjustment. In automatic modes, the camera adjusts the gain to maximize the dynamic range. Visibility:

15.9.4.431 GainAutoBalance

`GenApi::IEnumerationT<GainAutoBalanceEnums>& GainAutoBalance`

Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.

The gain coefficients of each channel or tap are adjusted so they are matched. Visibility: Beginner

15.9.4.432 GainSelector

`GenApi::IEnumerationT<GainSelectorEnums>& GainSelector`

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

15.9.4.433 Gamma

`GenApi::IFloat& Gamma`

Description: Controls the gamma correction of pixel intensity.

Visibility:

15.9.4.434 GammaEnable

`GenApi::IBoolean& GammaEnable`

Description: Enables/disables gamma correction.

Visibility:

15.9.4.435 GevActiveLinkCount

`GenApi::IInteger& GevActiveLinkCount`

Description: Indicates the current number of active logical links.

Visibility: Expert

15.9.4.436 GevCCP

`GenApi::IEnumerationT<GevCCPEnums>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility:

15.9.4.437 **GevCurrentDefaultGateway**

`GenApi::Integer& GevCurrentDefaultGateway`

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

15.9.4.438 **GevCurrentIPAddress**

`GenApi::Integer& GevCurrentIPAddress`

Description: Reports the IP address for the given logical link.

Visibility:

15.9.4.439 **GevCurrentIPConfigurationDHCP**

`GenApi::Boolean& GevCurrentIPConfigurationDHCP`

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

15.9.4.440 **GevCurrentIPConfigurationLLA**

`GenApi::Boolean& GevCurrentIPConfigurationLLA`

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

15.9.4.441 **GevCurrentIPConfigurationPersistentIP**

`GenApi::Boolean& GevCurrentIPConfigurationPersistentIP`

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

15.9.4.442 **GevCurrentPhysicalLinkConfiguration**

`GenApi::EnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration`

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

15.9.4.443 GevCurrentSubnetMask

`GenApi::Integer& GevCurrentSubnetMask`

Description: Reports the subnet mask of the given logical link.

Visibility:

15.9.4.444 GevDiscoveryAckDelay

`GenApi::Integer& GevDiscoveryAckDelay`

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

15.9.4.445 GevFirstURL

`GenApi::IString& GevFirstURL`

Description: The first choice of URL for the XML device description file.

Visibility:

15.9.4.446 GevGVCPExtendedStatusCodes

`GenApi::IBoolean& GevGVCPExtendedStatusCodes`

Description: Enables the generation of extended status codes.

Visibility: Guru

15.9.4.447 GevGVCPExtendedStatusCodesSelector

`GenApi::IEnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodesSelector`

Description: Selects the GigE Vision version to control extended status codes for.

Visibility: Guru

15.9.4.448 GevGVCPHeartbeatDisable

`GenApi::IBoolean& GevGVCPHeartbeatDisable`

Description: Disables the GVCP heartbeat.

Visibility:

15.9.4.449 GevGVCPPendingAck

`GenApi::IBoolean& GevGVCPPendingAck`

Description: Enables the generation of PENDING_ACK.

Visibility:

15.9.4.450 GevGVCPPendingTimeout

`GenApi::IInteger& GevGVCPPendingTimeout`

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.

Visibility:

15.9.4.451 GevGVSPExtendedIDMode

`GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode`

Description: Enables the extended IDs mode.

Visibility: Expert

15.9.4.452 GevHeartbeatTimeout

`GenApi::IInteger& GevHeartbeatTimeout`

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

15.9.4.453 GevIEEE1588

`GenApi::IBoolean& GevIEEE1588`

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

15.9.4.454 GevIEEE1588ClockAccuracy

`GenApi::IEnumerationT<GevIEEE1588ClockAccuracyEnums>& GevIEEE1588ClockAccuracy`

Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.

Visibility:

15.9.4.455 GevIEEE1588Mode

`GenApi::IEnumerationT<GevIEEE1588ModeEnums>&   GevIEEE1588Mode`

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

15.9.4.456 GevIEEE1588Status

`GenApi::IEnumerationT<GevIEEE1588StatusEnums>&   GevIEEE1588Status`

Description: Provides the status of the IEEE 1588 clock.

Visibility:

15.9.4.457 GevInterfaceSelector

`GenApi::IInteger&   GevInterfaceSelector`

Description: Selects which logical link to control.

Visibility:

15.9.4.458 GevIPConfigurationStatus

`GenApi::IEnumerationT<GevIPConfigurationStatusEnums>&   GevIPConfigurationStatus`

Description: Reports the current IP configuration status.

Visibility: Beginner

15.9.4.459 GevMACAddress

`GenApi::IInteger&   GevMACAddress`

Description: MAC address of the logical link.

Visibility:

15.9.4.460 GevMCDA

`GenApi::IInteger&   GevMCDA`

Description: Controls the destination IP address of the message channel Visibility:

15.9.4.461 GevMCPHostPort

`GenApi::Integer&   GevMCPHostPort`

Description: The port to which the device must send messages Visibility:

15.9.4.462 GevMCRC

`GenApi::Integer&   GevMCRC`

Description: Indicates the number of retries of the message channel.

Visibility:

15.9.4.463 GevMCSP

`GenApi::Integer&   GevMCSP`

Description: Indicates the source port of the message channel.

Visibility:

15.9.4.464 GevMCTT

`GenApi::Integer&   GevMCTT`

Description: Indicates the transmission timeout of the message channel.

Visibility:

15.9.4.465 GevNumberOfInterfaces

`GenApi::Integer&   GevNumberOfInterfaces`

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

15.9.4.466 GevPAUSEFrameReception

`GenApi::Boolean&   GevPAUSEFrameReception`

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

15.9.4.467 GevPAUSEFrameTransmission

`GenApi::IBoolean& GevPAUSEFrameTransmission`

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

15.9.4.468 GevPersistentDefaultGateway

`GenApi::IInteger& GevPersistentDefaultGateway`

Description: Controls the persistent default gateway for this logical link.

Visibility:

15.9.4.469 GevPersistentIPAddress

`GenApi::IInteger& GevPersistentIPAddress`

Description: Controls the Persistent IP address for this logical link.

Visibility:

15.9.4.470 GevPersistentSubnetMask

`GenApi::IInteger& GevPersistentSubnetMask`

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

15.9.4.471 GevPhysicalLinkConfiguration

`GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration`

Description: Controls the principal physical link configuration to use on next restart/power-up of the device.

Visibility: Expert

15.9.4.472 GevPrimaryApplicationIPAddress

`GenApi::IInteger& GevPrimaryApplicationIPAddress`

Description: Returns the address of the primary application.

Visibility: Guru

15.9.4.473 GevPrimaryApplicationSocket

`GenApi::Integer& GevPrimaryApplicationSocket`

Description: Returns the UDP source port of the primary application.

Visibility: Guru

15.9.4.474 GevPrimaryApplicationSwitchoverKey

`GenApi::Integer& GevPrimaryApplicationSwitchoverKey`

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

15.9.4.475 GevSCCFGAllInTransmission

`GenApi::Boolean& GevSCCFGAllInTransmission`

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

15.9.4.476 GevSCCFGExtendedChunkData

`GenApi::Boolean& GevSCCFGExtendedChunkData`

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

15.9.4.477 GevSCCFGPacketResendDestination

`GenApi::Boolean& GevSCCFGPacketResendDestination`

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.

When True, the source IP address provided in the packet resend command packet is used. When False, the value set in the `GevSCDA[GevStreamChannelSelector]` feature is used. Visibility: Guru

15.9.4.478 GevSCCFGUnconditionalStreaming

`GenApi::Boolean& GevSCCFGUnconditionalStreaming`

Description: Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

Visibility:

15.9.4.479 GevSCDA

`GenApi::Integer & GevSCDA`

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

Visibility:

15.9.4.480 GevSCPD

`GenApi::Integer & GevSCPD`

Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.

This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device. Visibility:

15.9.4.481 GevSCPDirection

`GenApi::Integer & GevSCPDirection`

Description: Transmit or Receive of the channel Visibility:

15.9.4.482 GevSCPHostPort

`GenApi::Integer & GevSCPHostPort`

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.

Visibility:

15.9.4.483 GevSCPInterfaceIndex

`GenApi::Integer & GevSCPInterfaceIndex`

Description: Index of the logical link to use.

Visibility:

15.9.4.484 GevSCPSBigEndian

`GenApi::Boolean & GevSCPSBigEndian`

Description: Endianness of multi-byte pixel data for this stream.

Visibility:

15.9.4.485 GevSCPSDoNotFragment

`GenApi::IBoolean&   GevSCPSDoNotFragment`

Description: The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.

Visibility:

15.9.4.486 GevSCPSFireTestPacket

`GenApi::IBoolean&   GevSCPSFireTestPacket`

Description: Sends a test packet.

Visibility:

15.9.4.487 GevSCPSPacketSize

`GenApi::IInteger&   GevSCPSPacketSize`

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

15.9.4.488 GevSCSP

`GenApi::IInteger&   GevSCSP`

Description: Indicates the source port of the stream channel.

Visibility:

15.9.4.489 GevSCZoneConfigurationLock

`GenApi::IBoolean&   GevSCZoneConfigurationLock`

Description: Controls whether the selected stream channel multi-zone configuration is locked.

When locked, the GVSP transmitter is not allowed to change the number of zones and their direction during block acquisition and transmission. Visibility: Guru

15.9.4.490 GevSCZoneCount

`GenApi::IInteger&   GevSCZoneCount`

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

15.9.4.491 GevSCZoneDirectionAll

`GenApi::Integer&   GevSCZoneDirectionAll`

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

15.9.4.492 GevSecondURL

`GenApi::String&   GevSecondURL`

Description: The second choice of URL to the XML device description file.

Visibility:

15.9.4.493 GevStreamChannelSelector

`GenApi::Integer&   GevStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility:

15.9.4.494 GevSupportedOption

`GenApi::Boolean&   GevSupportedOption`

Description: Returns if the selected GEV option is supported.

Visibility:

15.9.4.495 GevSupportedOptionSelector

`GenApi::EnumerationT<GevSupportedOptionSelectorEnums>&   GevSupportedOptionSelector`

Description: Selects the GEV option to interrogate for existing support.

Visibility:

15.9.4.496 GevTimestampTickFrequency

`GenApi::Integer&   GevTimestampTickFrequency`

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

15.9.4.497 GuiXmlManifestAddress

`GenApi::Integer& GuiXmlManifestAddress`

Description: Location of the GUI XML manifest table.

Visibility:

15.9.4.498 Height

`GenApi::Integer& Height`

Description:

Height of the image provided by the device (in pixels).

Visibility:

15.9.4.499 HeightMax

`GenApi::Integer& HeightMax`

Description: Maximum height of the image (in pixels). This dimension is calculated after vertical binning.

HeightMax does not take into account the current Region of interest (Height or OffsetY). Visibility:

15.9.4.500 ImageComponentEnable

`GenApi::Boolean& ImageComponentEnable`

Description: Controls if the selected component streaming is active.

Visibility: Beginner

15.9.4.501 ImageComponentSelector

`GenApi::EnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector`

Description: Selects a component to activate data streaming from.

Visibility: Beginner

15.9.4.502 ImageCompressionBitrate

`GenApi::IFloat& ImageCompressionBitrate`

Description: Control the rate of the produced compressed stream.

Visibility: Expert

15.9.4.503 ImageCompressionJPEGFormatOption

`GenApi::IEnumerationT<ImageCompressionJPEGFormatOptionEnums>& ImageCompressionJPEGFormatOption`

Description: When JPEG is selected as the compression format, a device might optionally offer better control over JPEG-specific options through this feature.

Visibility: Expert

15.9.4.504 ImageCompressionMode

`GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode`

Description: Visibility:

15.9.4.505 ImageCompressionQuality

`GenApi::IInteger& ImageCompressionQuality`

Description: Control the quality of the produced compressed stream.

Visibility: Expert

15.9.4.506 ImageCompressionRateOption

`GenApi::IEnumerationT<ImageCompressionRateOptionEnums>& ImageCompressionRateOption`

Description: Two rate controlling options are offered: fixed bit rate or fixed quality.

The exact implementation to achieve one or the other is vendor-specific. Visibility: Expert

15.9.4.507 IspEnable

`GenApi::IBoolean& IspEnable`

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e. mono).

Visibility:

15.9.4.508 LineFilterWidth

`GenApi::IFloat& LineFilterWidth`

Description: Filter width in microseconds for the selected line and filter combination Visibility:

15.9.4.509 LineFormat

`GenApi::IEnumerationT<LineFormatEnums>& LineFormat`

Description: Displays the current electrical format of the selected physical input or output Line.

Visibility:

15.9.4.510 LineInputFilterSelector

`GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector`

Description: Selects the kind of input filter to configure: Deglitch or Debounce.

Visibility:

15.9.4.511 LineInverter

`GenApi::IBoolean& LineInverter`

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

15.9.4.512 LineMode

`GenApi::IEnumerationT<LineModeEnums>& LineMode`

Description: Controls if the physical Line is used to Input or Output a signal.

Visibility:

15.9.4.513 LinePitch

`GenApi::IInteger& LinePitch`

Description: Total number of bytes between 2 successive lines.

This feature is used to facilitate alignment of image data. Visibility: Expert

15.9.4.514 LineSelector

```
GenApi::IEnumerationT<LineSelectorEnums>& LineSelector
```

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

15.9.4.515 LineSource

```
GenApi::IEnumerationT<LineSourceEnums>& LineSource
```

Description: Selects which internal acquisition or I/O source signal to output on the selected line.

LineMode must be Output. Visibility:

15.9.4.516 LineStatus

```
GenApi::IBoolean& LineStatus
```

Description: Returns the current status of the selected input or output Line Visibility:

15.9.4.517 LineStatusAll

```
GenApi::IInteger& LineStatusAll
```

Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).

This allows simultaneous reading of all line statuses at once. Visibility:

15.9.4.518 LinkErrorCount

```
GenApi::IInteger& LinkErrorCount
```

Description: Counts the number of error on the link.

Visibility:

15.9.4.519 LinkUptime

```
GenApi::IInteger& LinkUptime
```

Description: Time since the last phy negotiation (enumeration).

Visibility:

15.9.4.520 LogicBlockLUTInputActivation

`GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation`

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

15.9.4.521 LogicBlockLUTInputSelector

`GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector`

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

15.9.4.522 LogicBlockLUTInputSource

`GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource`

Description: Selects the source for the input into the Logic LUT.

Visibility:

15.9.4.523 LogicBlockLUTOutputValue

`GenApi::IBoolean& LogicBlockLUTOutputValue`

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

15.9.4.524 LogicBlockLUTOutputValueAll

`GenApi::IInteger& LogicBlockLUTOutputValueAll`

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

15.9.4.525 LogicBlockLUTRowIndex

`GenApi::IInteger& LogicBlockLUTRowIndex`

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

15.9.4.526 LogicBlockLUTSelector

`GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector`

Description: Selects which LogicBlock LUT to configure Visibility:

15.9.4.527 LogicBlockSelector

`GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector`

Description: Selects which LogicBlock to configure Visibility:

15.9.4.528 LUTEnable

`GenApi::IBoolean& LUTEnable`

Description:

Activates the selected LUT.

Visibility:

15.9.4.529 LUTIndex

`GenApi::IInteger& LUTIndex`

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

15.9.4.530 LUTSelector

`GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector`

Description:

Selects which LUT to control.

Visibility:

15.9.4.531 LUTValue

`GenApi::Integer& LUTValue`

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

15.9.4.532 LUTValueAll

`GenApi::Register& LUTValueAll`

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

15.9.4.533 MaxDeviceResetTime

`GenApi::Integer& MaxDeviceResetTime`

Description: Time to wait until device reset complete (ms).

Visibility:

15.9.4.534 OffsetX

`GenApi::Integer& OffsetX`

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

15.9.4.535 OffsetY

`GenApi::Integer& OffsetY`

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

15.9.4.536 PacketResendRequestCount

`GenApi::Integer& PacketResendRequestCount`

Description: Counts the number of resend requests received from the host.

Visibility:

15.9.4.537 PayloadSize

`GenApi::Integer& PayloadSize`

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

15.9.4.538 PixelColorFilter

`GenApi::EnumerationT<PixelColorFilterEnums>& PixelColorFilter`

Description: Type of color filter that is applied to the image.

Only applies to Bayer pixel formats. All others have no color filter.

Visibility:

15.9.4.539 PixelDynamicRangeMax

`GenApi::Integer& PixelDynamicRangeMax`

Description: Maximum value that can be returned during the digitization process.

This corresponds to the brightest value of the camera. For color cameras, this returns the biggest value that each color component can take.

Visibility:

15.9.4.540 PixelDynamicRangeMin

`GenApi::Integer& PixelDynamicRangeMin`

Description: Minimum value that can be returned during the digitization process.

This corresponds to the darkest value of the camera. For color cameras, this returns the smallest value that each color component can take.

Visibility:

15.9.4.541 PixelFormat

`GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat`

Description: Format of the pixel provided by the camera.

Visibility:

15.9.4.542 PixelFormatInfoID

`GenApi::IInteger& PixelFormatInfoID`

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

15.9.4.543 PixelFormatInfoSelector

`GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector`

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

15.9.4.544 PixelSize

`GenApi::IEnumerationT<PixelSizeEnums>& PixelSize`

Description: Total size in bits of a pixel of the image.

Visibility:

15.9.4.545 PowerSupplyCurrent

`GenApi::IFloat& PowerSupplyCurrent`

Description:

Indicates the output current of the selected power supply (A).

Visibility:

15.9.4.546 PowerSupplyVoltage

`GenApi::IFloat& PowerSupplyVoltage`

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

15.9.4.547 RegionDestination

`GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination`

Description: Control the destination of the selected region.

Visibility: Expert

15.9.4.548 RegionMode

`GenApi::IEnumerationT<RegionModeEnums>& RegionMode`

Description: Controls if the selected Region of interest is active and streaming.

Visibility: Beginner

15.9.4.549 RegionSelector

`GenApi::IEnumerationT<RegionSelectorEnums>& RegionSelector`

Description: Selects the Region of interest to control.

The RegionSelector feature allows devices that are able to extract multiple regions out of an image, to configure the features of those individual regions independently. Visibility: Beginner

15.9.4.550 ReverseX

`GenApi::IBoolean& ReverseX`

Description: Horizontally flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGR16.

Visibility:

15.9.4.551 ReverseY

`GenApi::IBoolean& ReverseY`

Description: Vertically flips the image sent by the device.

The region of interest is applied after flipping. For color cameras the bayer pixel format is affected. For example, BayerRG16 changes to BayerGB16.

Visibility:

15.9.4.552 RgbTransformLightSource

`GenApi::IEnumerationT<RgbTransformLightSourceEnums>& RgbTransformLightSource`

Description:

Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources. Selecting a value also sets the white balance ratios (BalanceRatioRed and BalanceRatioBlue), but those can be overwritten through manual or auto white balance.

Visibility:

15.9.4.553 Saturation

`GenApi::IFloat& Saturation`

Description: Controls the color saturation.

Visibility:

15.9.4.554 SaturationEnable

`GenApi::IBoolean& SaturationEnable`

Description: Enables/disables Saturation adjustment.

Visibility:

15.9.4.555 Scan3dAxisMax

`GenApi::IFloat& Scan3dAxisMax`

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

15.9.4.556 Scan3dAxisMin

`GenApi::IFloat& Scan3dAxisMin`

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

15.9.4.557 Scan3dCoordinateOffset

`GenApi::IFloat& Scan3dCoordinateOffset`

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

15.9.4.558 Scan3dCoordinateReferenceSelector

`GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReferenceSelector`

Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.

Visibility: Expert

15.9.4.559 Scan3dCoordinateReferenceValue

`GenApi::IFloat& Scan3dCoordinateReferenceValue`

Description: Returns the reference value selected.

Reads the value of a rotation or translation value for the current (Anchor or Transformed) coordinate system transformation to the Reference system. Visibility: Expert

15.9.4.560 Scan3dCoordinateScale

`GenApi::IFloat& Scan3dCoordinateScale`

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

15.9.4.561 Scan3dCoordinateSelector

`GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector`

Description: Selects the individual coordinates in the vectors for 3D information/transformation.

Visibility: Expert

15.9.4.562 Scan3dCoordinateSystem

`GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem`

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

15.9.4.563 Scan3dCoordinateSystemReference

`GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference`

Description: Defines coordinate system reference location.

Visibility: Expert

15.9.4.564 Scan3dCoordinateTransformSelector

`GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransformSelector`

Description: Sets the index to read/write a coordinate transform value.

Visibility: Expert

15.9.4.565 Scan3dDistanceUnit

`GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit`

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

15.9.4.566 Scan3dInvalidDataFlag

`GenApi::IBoolean& Scan3dInvalidDataFlag`

Description: Enables the definition of a non-valid flag value in the data stream.

Note that the confidence output is an alternate recommended way to identify non-valid pixels. Using an `Scan3dInvalidDataValue` may give processing penalties due to special handling. Visibility: Expert

15.9.4.567 Scan3dInvalidDataValue

`GenApi::IFloat& Scan3dInvalidDataValue`

Description: Value which identifies a non-valid pixel if `Scan3dInvalidDataFlag` is enabled.

Visibility: Expert

15.9.4.568 Scan3dOutputMode

`GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode`

Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.

Visibility: Expert

15.9.4.569 Scan3dTransformValue

`GenApi::IFloat& Scan3dTransformValue`

Description: Specifies the transform value selected.

For translations (Scan3dCoordinateTransformSelector = TranslationX/Y/Z) it is expressed in the distance unit of the system, for rotations (Scan3dCoordinateTransformSelector =RotationX/Y/Z) in degrees. Visibility: Expert

15.9.4.570 SensorDescription

`GenApi::IString& SensorDescription`

Description: Returns Sensor Description Visibility:

15.9.4.571 SensorDigitizationTaps

`GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps`

Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.

Visibility: Expert

15.9.4.572 SensorHeight

`GenApi::IInteger& SensorHeight`

Description: Effective height of the sensor in pixels.

Visibility:

15.9.4.573 SensorShutterMode

`GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode`

Description: Sets the shutter mode of the device.

Visibility:

15.9.4.574 SensorTaps

`GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps`

Description: Number of taps of the camera sensor.

Visibility: Expert

15.9.4.575 SensorWidth

`GenApi::IInteger& SensorWidth`

Description: Effective width of the sensor in pixels.

Visibility:

15.9.4.576 SequencerConfigurationMode

`GenApi::IEnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode`

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

15.9.4.577 SequencerConfigurationValid

`GenApi::IEnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid`

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

15.9.4.578 SequencerFeatureEnable

`GenApi::IBoolean& SequencerFeatureEnable`

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

15.9.4.579 SequencerMode

`GenApi::IEnumerationT<SequencerModeEnums>& SequencerMode`

Description: Controls whether or not a sequencer is active.

Visibility:

15.9.4.580 SequencerPathSelector

`GenApi::IInteger& SequencerPathSelector`

Description:

Selects branching path to be used for subsequent settings.

Visibility:

15.9.4.581 SequencerSetActive

`GenApi::IInteger& SequencerSetActive`

Description: Displays the currently active sequencer set.

Visibility:

15.9.4.582 SequencerSetLoad

`GenApi::ICommand& SequencerSetLoad`

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

15.9.4.583 SequencerSetNext

`GenApi::IInteger& SequencerSetNext`

Description: Specifies the next sequencer set.

Visibility:

15.9.4.584 SequencerSetSave

`GenApi::ICommand& SequencerSetSave`

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

15.9.4.585 SequencerSetSelector

`GenApi::IInteger& SequencerSetSelector`

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

15.9.4.586 SequencerSetStart

`GenApi::IInteger& SequencerSetStart`

Description: Sets the first sequencer set to be used.

Visibility:

15.9.4.587 SequencerSetValid

`GenApi::IEnumerationT<SequencerSetValidEnums>& SequencerSetValid`

Description:

Displays whether the currently selected sequencer set's register contents are valid to use.

Visibility:

15.9.4.588 SequencerTriggerActivation

`GenApi::IEnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation`

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

15.9.4.589 SequencerTriggerSource

`GenApi::IEnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource`

Description:

Specifies the internal signal or physical input line to use as the sequencer trigger source.

Visibility:

15.9.4.590 SerialPortBaudRate

`GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate`

Description: This feature controls the baud rate used by the selected serial port.

Visibility:

15.9.4.591 SerialPortDataBits

`GenApi::IInteger& SerialPortDataBits`

Description: This feature controls the number of data bits used by the selected serial port.

Possible values that can be used are between 5 and 9. Visibility:

15.9.4.592 SerialPortParity

`GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity`

Description: This feature controls the parity used by the selected serial port.

Visibility:

15.9.4.593 SerialPortSelector

`GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector`

Description: Selects which serial port of the device to control.

Visibility:

15.9.4.594 SerialPortSource

`GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource`

Description: Specifies the physical input Line on which to receive serial data.

Visibility:

15.9.4.595 SerialPortStopBits

`GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits`

Description: This feature controls the number of stop bits used by the selected serial port.

Visibility:

15.9.4.596 SerialReceiveFramingErrorCount

`GenApi::IInteger& SerialReceiveFramingErrorCount`

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

15.9.4.597 SerialReceiveParityErrorCount

`GenApi::IInteger& SerialReceiveParityErrorCount`

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

15.9.4.598 SerialReceiveQueueClear

`GenApi::ICommand& SerialReceiveQueueClear`

Description: This is a command that clears the device serial port receive queue.

Visibility:

15.9.4.599 SerialReceiveQueueCurrentCharacterCount

`GenApi::Integer& SerialReceiveQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

15.9.4.600 SerialReceiveQueueMaxCharacterCount

`GenApi::Integer& SerialReceiveQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

15.9.4.601 SerialTransmitQueueCurrentCharacterCount

`GenApi::Integer& SerialTransmitQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

15.9.4.602 SerialTransmitQueueMaxCharacterCount

`GenApi::Integer& SerialTransmitQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

15.9.4.603 Sharpening

`GenApi::Float& Sharpening`

Description:

Controls the amount to sharpen a signal. The sharpened amount is proportional to the difference between a pixel and its neighbors. A negative value smooths out the difference, while a positive value amplifies the difference. You can boost by a maximum of 8x, but smoothing is limited to 1x (in float). Default value: 2.0

Visibility:

15.9.4.604 SharpeningAuto

`GenApi::IBoolean& SharpeningAuto`

Description:

Enables/disables the auto sharpening feature. When enabled, the camera automatically determines the sharpening threshold based on the noise level of the camera.

Visibility:

15.9.4.605 SharpeningEnable

`GenApi::IBoolean& SharpeningEnable`

Description:

Enables/disables the sharpening feature. Sharpening is disabled by default.

Visibility:

15.9.4.606 SharpeningThreshold

`GenApi::IFloat& SharpeningThreshold`

Description:

Controls the minimum intensity gradient change to invoke sharpening. When "Sharpening Auto" is enabled, this is determined automatically by the device. The threshold is specified as a fraction of the total intensity range, and ranges from 0 to 0.25. A threshold higher than 25% produces little to no difference than 25%. High thresholds sharpen only areas with significant intensity changes. Low thresholds sharpen more areas.

Visibility:

15.9.4.607 SoftwareSignalPulse

`GenApi::ICommand& SoftwareSignalPulse`

Description: Generates a pulse signal that can be used as a software trigger.

This command can be used to trigger other modules that accept a SoftwareSignal as trigger source. Visibility: Beginner

15.9.4.608 SoftwareSignalSelector

`GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector`

Description: Selects which Software Signal features to control.

Visibility: Beginner

15.9.4.609 SourceCount

`GenApi::IInteger& SourceCount`

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

15.9.4.610 SourceSelector

`GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector`

Description: Selects the source to control.

Visibility: Beginner

15.9.4.611 Test0001

`GenApi::IInteger& Test0001`

Description: For testing only.

Visibility:

15.9.4.612 TestEventGenerate

`GenApi::ICommand& TestEventGenerate`

Description: This command generates a test event and sends it to the host.

Visibility:

15.9.4.613 TestPattern

`GenApi::IEnumerationT<TestPatternEnums>& TestPattern`

Description:

Selects the type of test pattern that is generated by the device as image source.

Visibility:

15.9.4.614 TestPatternGeneratorSelector

GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector

Description:

Selects which test pattern generator is controlled by the TestPattern feature.

Visibility:

15.9.4.615 TestPendingAck

GenApi::IInteger& TestPendingAck

Description: Tests the device's pending acknowledge feature.

When this feature is written, the device waits a time period corresponding to the value of TestPendingAck before acknowledging the write. Visibility: Guru

15.9.4.616 TimerDelay

GenApi::IFloat& TimerDelay

Description: Sets the duration (in microseconds) of the delay to apply at the reception of a trigger before starting the Timer.

Visibility: Expert

15.9.4.617 TimerDuration

GenApi::IFloat& TimerDuration

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

15.9.4.618 TimerReset

GenApi::ICommand& TimerReset

Description: Does a software reset of the selected timer and starts it.

The timer starts immediately after the reset unless a timer trigger is active. Visibility: Expert

15.9.4.619 TimerSelector

`GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector`

Description: Selects which Timer to configure.

Visibility: Expert

15.9.4.620 TimerStatus

`GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus`

Description: Returns the current status of the Timer.

Visibility: Expert

15.9.4.621 TimerTriggerActivation

`GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation`

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

15.9.4.622 TimerTriggerSource

`GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource`

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

15.9.4.623 TimerValue

`GenApi::IFloat& TimerValue`

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

15.9.4.624 Timestamp

`GenApi::IInteger& Timestamp`

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

15.9.4.625 TimestampLatch

`GenApi::ICommand& TimestampLatch`

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

15.9.4.626 TimestampLatchValue

`GenApi::IInteger& TimestampLatchValue`

Description: Returns the latched value of the timestamp counter.

Visibility:

15.9.4.627 TimestampReset

`GenApi::ICommand& TimestampReset`

Description: Resets the current value of the device timestamp counter.

Visibility:

15.9.4.628 TLParamsLocked

`GenApi::IInteger& TLParamsLocked`

Description: Visibility:

15.9.4.629 TransferAbort

`GenApi::ICommand& TransferAbort`

Description: Aborts immediately the streaming of data block(s).

Aborting the transfer will result in the lost of the data that is present or currently entering in the block queue. However, the next new block received will be stored in the queue and transferred to the host when the streaming is restarted. If implemented, this feature should be available when the TransferControlMode is set to "UserControlled". Visibility: Expert

15.9.4.630 TransferBlockCount

`GenApi::IInteger& TransferBlockCount`

Description: Specifies the number of data blocks (images) that the device should stream before stopping.

This feature is only active if the Transfer Operation Mode is set to Multi Block. Visibility:

15.9.4.631 TransferBurstCount

`GenApi::Integer& TransferBurstCount`

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

15.9.4.632 TransferComponentSelector

`GenApi::EnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector`

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

15.9.4.633 TransferControlMode

`GenApi::EnumerationT<TransferControlModeEnums>& TransferControlMode`

Description: Selects the control method for the transfers.

Basic and Automatic start transmitting data as soon as there is enough data to fill a link layer packet. User Controlled allows you to directly control the transfer of blocks. Visibility:

15.9.4.634 TransferOperationMode

`GenApi::EnumerationT<TransferOperationModeEnums>& TransferOperationMode`

Description: Selects the operation mode of the transfer.

Continuous is similar to Basic/Automatic but you can start/stop the transfer while acquisition runs independently. Multi Block transmits a specified number of blocks and then stops. Visibility:

15.9.4.635 TransferPause

`GenApi::Command& TransferPause`

Description: Pauses the streaming of data Block(s).

Pausing the streaming will immediately suspend the ongoing data transfer even if a block is partially transferred. The device will resume its transmission at the reception of a TransferResume command. Visibility: Guru

15.9.4.636 TransferQueueCurrentBlockCount

`GenApi::Integer& TransferQueueCurrentBlockCount`

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

15.9.4.637 TransferQueueMaxBlockCount

`GenApi::Integer& TransferQueueMaxBlockCount`

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

15.9.4.638 TransferQueueMode

`GenApi::EnumerationT<TransferQueueModeEnums>& TransferQueueMode`

Description: Specifies the operation mode of the transfer queue.

Visibility:

15.9.4.639 TransferQueueOverflowCount

`GenApi::Integer& TransferQueueOverflowCount`

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

Visibility:

15.9.4.640 TransferResume

`GenApi::Command& TransferResume`

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

15.9.4.641 TransferSelector

`GenApi::EnumerationT<TransferSelectorEnums>& TransferSelector`

Description: Selects which stream transfers are currently controlled by the selected Transfer features.

Visibility: Expert

15.9.4.642 TransferStart

`GenApi::Command& TransferStart`

Description: Starts the streaming of data blocks (images) out of the device.

This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

15.9.4.643 TransferStatus

`GenApi::IBoolean& TransferStatus`

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

15.9.4.644 TransferStatusSelector

`GenApi::IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector`

Description: Selects which status of the transfer module to read.

Visibility: Guru

15.9.4.645 TransferStop

`GenApi::ICommand& TransferStop`

Description: Stops the streaming of data block (images).

The current block transmission is completed. This feature is available when the Transfer Control Mode is set to User Controlled. Visibility:

15.9.4.646 TransferStreamChannel

`GenApi::IInteger& TransferStreamChannel`

Description: Selects the streaming channel that will be used to transfer the selected stream of data.

In general, this feature can be omitted and the default streaming channel will be used. Visibility: Guru

15.9.4.647 TransferTriggerActivation

`GenApi::IEnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation`

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

15.9.4.648 TransferTriggerMode

`GenApi::IEnumerationT<TransferTriggerModeEnums>& TransferTriggerMode`

Description: Controls if the selected trigger is active.

Visibility: Guru

15.9.4.649 TransferTriggerSelector

`GenApi::IEnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector`

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

15.9.4.650 TransferTriggerSource

`GenApi::IEnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource`

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

15.9.4.651 TriggerActivation

`GenApi::IEnumerationT<TriggerActivationEnums>& TriggerActivation`

Description: Specifies the activation mode of the trigger.

Visibility:

15.9.4.652 TriggerDelay

`GenApi::IFloat& TriggerDelay`

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

15.9.4.653 TriggerDivider

`GenApi::IInteger& TriggerDivider`

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

15.9.4.654 TriggerEventTest

`GenApi::ICommand& TriggerEventTest`

Description: This command generates a test event and sends it to the host.

Visibility:

15.9.4.655 TriggerMode

`GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode`

Description:

Controls whether or not trigger is active.

Visibility:

15.9.4.656 TriggerMultiplier

`GenApi::IInteger& TriggerMultiplier`

Description: Specifies a multiplication factor for the incoming trigger pulses.

It is used generally used in conjunction with TriggerDivider to control the ratio of triggers that are accepted.

Visibility: Expert

15.9.4.657 TriggerOverlap

`GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap`

Description: Specifies the overlap mode of the trigger.

Visibility:

15.9.4.658 TriggerSelector

`GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector`

Description: Selects the type of trigger to configure.

Visibility:

15.9.4.659 TriggerSoftware

`GenApi::ICommand& TriggerSoftware`

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

15.9.4.660 TriggerSource

`GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource`

Description:

Specifies the internal signal or physical input line to use as the trigger source.

Visibility:

15.9.4.661 UserOutputSelector

`GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector`

Description: Selects which bit of the User Output register is set by UserOutputValue.

Visibility:

15.9.4.662 UserOutputValue

`GenApi::IBoolean& UserOutputValue`

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

15.9.4.663 UserOutputValueAll

`GenApi::IInteger& UserOutputValueAll`

Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).

This allows simultaneous reading of all user output statuses at once. Visibility:

15.9.4.664 UserOutputValueAllMask

```
GenApi::Integer& UserOutputValueAllMask
```

Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.

If the UserOutputValueAllMask feature is present, setting the user Output register using UserOutputValueAll will only change the bits that have a corresponding bit in the mask set to one. Visibility: Expert

15.9.4.665 UserSetDefault

```
GenApi::EnumerationT<UserSetDefaultEnums>& UserSetDefault
```

Description:

Selects the feature User Set to load and make active by default when the device is restarted.

Visibility:

15.9.4.666 UserSetFeatureEnable

```
GenApi::Boolean& UserSetFeatureEnable
```

Description: Whether or not the selected feature is saved to user sets.

Visibility:

15.9.4.667 UserSetLoad

```
GenApi::Command& UserSetLoad
```

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

15.9.4.668 UserSetSave

`GenApi::ICommand& UserSetSave`

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

15.9.4.669 UserSetSelector

`GenApi::IEnumerationT<UserSetSelectorEnums>& UserSetSelector`

Description:

Selects the feature User Set to load, save or configure.

Visibility:

15.9.4.670 V3_3Enable

`GenApi::IBoolean& V3_3Enable`

Description: Internally generated 3.3V rail.

Enable to supply external circuits with power. This is different than standard logic outputs in that it is comparatively slow to switch but can supply a more significant amount of power. This is only available on some pins. Visibility:

15.9.4.671 WhiteClip

`GenApi::IFloat& WhiteClip`

Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.

The video signal will never exceed the white clipping point: it will saturate at that level. Visibility: Expert

15.9.4.672 WhiteClipSelector

`GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector`

Description: Selects which White Clip to control.

Visibility: Expert

15.9.4.673 Width

`GenApi::Integer& Width`

Description:

Width of the image provided by the device (in pixels).

Visibility:

15.9.4.674 WidthMax

`GenApi::Integer& WidthMax`

Description:

Maximum width of the image (in pixels). The dimension is calculated after horizontal binning. WidthMax does not take into account the current Region of interest (Width or OffsetX).

Visibility:

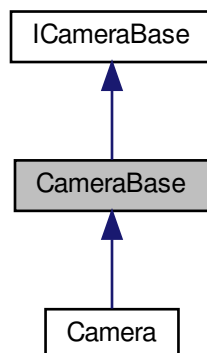
The documentation for this class was generated from the following file:

- include/[Camera.h](#)

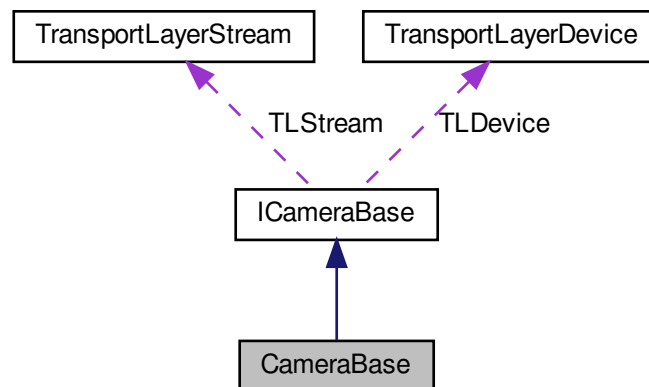
15.10 CameraBase Class Reference

The base class for the camera object.

Inheritance diagram for CameraBase:



Collaboration diagram for CameraBase:



Public Member Functions

- virtual [~CameraBase](#) (void)
Virtual Destructor.
- void [Init](#) ()
Init Connects to camera with read and write access, retrieves remote device and GUI XML and generates node map references.
- void [DeInit](#) ()
DeInit Disconnects camera port, resets camera back to read access and frees [GenICam](#) node map and GUI XML.
- bool [IsInitialized](#) ()
IsInitialized Checks if camera is initialized.
- bool [IsValid](#) ()
IsValid Checks a flag to determine if camera is still valid for use.
- [GenApi::INodeMap](#) & [GetNodeMap](#) () const
GetNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file.
- [GenApi::INodeMap](#) & [GetTLDeviceNodeMap](#) () const
GetTLDeviceNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.
- [GenApi::INodeMap](#) & [GetTLStreamNodeMap](#) (uint64_t streamIndex=0) const
GetTLStreamNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.
- GenApi::EAccessMode [GetAccessMode](#) () const
GetAccessMode Returns the access mode that the software has on the [Camera](#).
- void [ReadPort](#) (uint64_t iAddress, void *pBuffer, size_t iSize)
- void [WritePort](#) (uint64_t iAddress, const void *pBuffer, size_t iSize)
- void [BeginAcquisition](#) ()
BeginAcquisition Starts the image acquisition engine.
- void [EndAcquisition](#) ()
EndAcquisition Stops the image acquisition engine.
- BufferOwnership [GetBufferOwnership](#) () const
GetBufferOwnership Gets data buffer ownership.

- void [SetBufferOwnership](#) (const [BufferOwnership](#) mode)
SetBufferOwnership Sets data buffer ownership.
- uint64_t [GetUserBufferCount](#) () const
GetUserBufferCount Gets the number of user memory buffers.
- uint64_t [GetUserBufferSize](#) () const
GetUserBufferSize Gets the size of one user memory buffer (in bytes).
- uint64_t [GetUserBufferTotalSize](#) () const
GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).
- void [SetUserBuffers](#) (void *const pMemBuffers, uint64_t totalSize)
SetUserBuffers Specify contiguous user allocated memory to use as data buffers.
- void [SetUserBuffers](#) (void **const ppMemBuffers, const uint64_t bufferCount, const uint64_t bufferSize)
SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.
- [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout=[EVENT_TIMEOUT_INFINITE](#), uint64_t streamIndex=0)
GetNextImage Gets the next image that was received by the transport layer.
- [ImageList](#) [GetNextImageSync](#) (uint64_t grabTimeout=[EVENT_TIMEOUT_INFINITE](#))
GetNextImageSync If a camera supports one or more streams, this function gets one image from each of the camera's streams, and returns the image(s) in a list.
- [GenlCam::gcstring](#) [GetUniqueID](#) ()
GetUniqueID This returns a unique id string that identifies the camera.
- bool [IsStreaming](#) () const
IsStreaming Returns true if the camera is currently streaming or false if it is not.
- [GenlCam::gcstring](#) [GetGuiXml](#) () const
Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.
- void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)
RegisterEventHandler(EventHandler &) Registers a generic device, image or image list event handler for the camera.
- void [RegisterEventHandler](#) ([DeviceEventHandler](#) &evtHandlerToRegister, const [GenlCam::gcstring](#) &eventName)
RegisterEventHandler(DeviceEventHandler &, const GenlCam::gcstring &) Registers a specific device event handler for the camera given a device event name.
- void [RegisterEventHandler](#) ([ImageEventHandler](#) &evtHandlerToRegister, uint64_t streamIndex)
RegisterEventHandler(ImageEventHandler &, uint64_t) Registers a specific stream handler for the camera given a stream index.
- void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)
UnregisterEventHandler Unregisters any type of event handler for the camera.
- unsigned int [GetNumImagesInUse](#) ()
GetNumImagesInUse Returns the number of images that are currently in use.
- unsigned int [GetNumDataStreams](#) ()
GetNumDataStreams Returns the number of streams that a device supports.
- unsigned int [DiscoverMaxPacketSize](#) ()
DiscoverMaxPacketSize Returns the largest packet size that can be safely used on the interface that device is connected to.
- void [ForceIP](#) ()
ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Protected Member Functions

- [CameraBase](#) (void)
Default constructor.
- [CameraBase](#) (const [CameraBase](#) &)
Copy constructor.
- [CameraBase](#) & [operator=](#) (const [CameraBase](#) &)
Assignment operator.

Friends

- class [InterfaceImpl](#)

Additional Inherited Members

15.10.1 Detailed Description

The base class for the camera object.

15.10.2 Constructor & Destructor Documentation

15.10.2.1 `~CameraBase()`

```
virtual ~CameraBase (  
    void ) [virtual]
```

Virtual Destructor.

15.10.2.2 `CameraBase()` [1/2]

```
CameraBase (  
    void ) [protected]
```

Default constructor.

15.10.2.3 `CameraBase()` [2/2]

```
CameraBase (  
    const CameraBase & ) [protected]
```

Copy constructor.

15.10.3 Member Function Documentation

15.10.3.1 BeginAcquisition()

```
void BeginAcquisition ( ) [virtual]
```

BeginAcquisition Starts the image acquisition engine.

The camera must be initialized via a call to [Init\(\)](#) before starting an acquisition.

See also

[Init\(\)](#)

Implements [ICameraBase](#).

15.10.3.2 DeInit()

```
void DeInit ( ) [virtual]
```

DeInit Disconnects camera port, resets camera back to read access and frees [GenICam](#) node map and GUI XML.

Do not call more functions that access the remote device such as WritePort/ReadPort after calling [DeInit\(\)](#). Events should also be unregistered before calling camera [DeInit\(\)](#). Otherwise an exception will be thrown in the [DeInit\(\)](#) call and require the user to unregister events before the camera can be re-initialized again.

See also

[Init\(\)](#)

[UnregisterEventHandler\(EventHandler & evtHandlerToUnregister\)](#)

Implements [ICameraBase](#).

15.10.3.3 DiscoverMaxPacketSize()

```
unsigned int DiscoverMaxPacketSize ( ) [virtual]
```

DiscoverMaxPacketSize Returns the largest packet size that can be safely used on the interface that device is connected to.

Returns

The maximum packet size returned.

Implements [ICameraBase](#).

15.10.3.4 EndAcquisition()

```
void EndAcquisition ( ) [virtual]
```

EndAcquisition Stops the image acquisition engine.

If [EndAcquisition\(\)](#) is called without a prior call to [BeginAcquisition\(\)](#) an error message "Camera is not started" will be thrown. All Images that were acquired using [GetNextImage\(\)](#) or [GetNextImageSync\(\)](#) need to be released first using `image->Release()` or `imageList->Release()` before calling [EndAcquisition\(\)](#). All buffers in the input pool and output queue will be discarded when [EndAcquisition\(\)](#) is called.

See also

[Init\(\)](#)

[BeginAcquisition\(\)](#)

[GetNextImage\(grabTimeout \)](#)

[GetNextImageSync\(grabTimeout \)](#)

[Image::Release\(\)](#)

Implements [ICameraBase](#).

15.10.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements [ICameraBase](#).

15.10.3.6 GetAccessMode()

```
GenApi::EAccessMode GetAccessMode ( ) const [virtual]
```

GetAccessMode Returns the access mode that the software has on the [Camera](#).

The camera does not need to be initialized before calling this function.

See also

[Init\(\)](#)

Returns

An enumeration value indicating the access mode

Implements [ICameraBase](#).

15.10.3.7 GetBufferOwnership()

```
BufferOwnership GetBufferOwnership ( ) const [virtual]
```

GetBufferOwnership Gets data buffer ownership.

The data buffers can be owned by [System](#) or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

[SetBufferOwnership\(\)](#)

[SetUserBuffers\(\)](#)

Returns

Buffer ownership (system or user)

Implements [ICameraBase](#).

15.10.3.8 GetGuiXml()

```
GenICam::gcstring GetGuiXml ( ) const [virtual]
```

Returns the GUI XML that can be passed into the [Spinnaker](#) GUI framework.

Returns

[GenICam::gcstring](#) that represents the uncompressed GUI XML file

Implements [ICameraBase](#).

15.10.3.9 GetNextImage()

```
ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE,
    uint64_t streamIndex = 0 ) [virtual]
```

GetNextImage Gets the next image that was received by the transport layer.

This function will block for the specified timeout period until an image arrives. Most cameras support one stream so the default streamIndex is 0, but if a camera supports multiple streams the user can input the streamIndex to select from which stream to grab images.

See also

[Init\(\)](#)

[BeginAcquisition\(\)](#)

[EndAcquisition\(\)](#)

Parameters

<i>grabTimeout</i>	A 64bit value that represents a timeout in milliseconds
<i>streamIndex</i>	The index of the stream to grab the image from

Returns

pointer to an [Image](#) object

Implements [ICameraBase](#).

15.10.3.10 GetNextImageSync()

```
ImageList GetNextImageSync (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE )
```

GetNextImageSync If a camera supports one or more streams, this function gets one image from each of the camera's streams, and returns the image(s) in a list.

This function will block for the specified timeout period until an image arrives on all the streams.

See also

[Init\(\)](#)
[BeginAcquisition\(\)](#)
[EndAcquisition\(\)](#)

Parameters

<i>grabTimeout</i>	A 64bit value that represents a timeout in milliseconds

Returns

a list containing the [Image](#) objects

15.10.3.11 GetNodeMap()

```
GenApi::INodeMap& GetNodeMap ( ) const [virtual]
```

GetNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file.

The camera must be initialized by a call to [Init\(\)](#) first before a node map reference can be successfully acquired.

See also

[Init\(\)](#)

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.10.3.12 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

Returns

The number of data streams

Implements [ICameraBase](#).

15.10.3.13 GetNumImagesInUse()

```
unsigned int GetNumImagesInUse ( ) [virtual]
```

GetNumImagesInUse Returns the number of images that are currently in use.

Each of the images that are currently in use must be cleaned up with a call to `image->Release()` before calling `system->ReleaseInstance()`.

Returns

The number of images that needs to be cleaned up.

Implements [ICameraBase](#).

15.10.3.14 GetTLDeviceNodeMap()

```
GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [virtual]
```

GetTLDeviceNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Device module.

The camera does not need to be initialized before acquiring this node map.

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.10.3.15 GetTLStreamNodeMap()

```
GenApi::INodeMap& GetTLStreamNodeMap (
    uint64_t streamIndex = 0 ) const [virtual]
```

GetTLStreamNodeMap Gets a reference to the node map that is generated from a [GenICam](#) XML file for the GenTL Stream module.

The camera does not need to be initialized before acquiring this node map.

Parameters

<i>streamIndex</i>	The index of the stream to grab the image from

Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

15.10.3.16 GetUniqueID()

```
GenICam::gcstring GetUniqueID ( ) [virtual]
```

GetUniqueID This returns a unique id string that identifies the camera.

This is the camera serial number.

Returns

string that uniquely identifies the camera (serial number)

Implements [ICameraBase](#).

15.10.3.17 GetUserBufferCount()

```
uint64_t GetUserBufferCount ( ) const [virtual]
```

GetUserBufferCount Gets the number of user memory buffers.

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless [BeginAcquisition\(\)](#) has been called.

See also

[BeginAcquisition\(\)](#)
[SetUserBuffers\(\)](#)

Returns

The number of user memory buffers

Implements [ICameraBase](#).

15.10.3.18 GetUserBufferSize()

```
uint64_t GetUserBufferSize ( ) const [virtual]
```

GetUserBufferSize Gets the size of one user memory buffer (in bytes).

This will throw an exception if user memory buffer has not been set. If the user memory is contiguous, this will throw an exception unless [BeginAcquisition\(\)](#) has been called. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$ where 1024 is the USB3 packet size.

See also

[BeginAcquisition\(\)](#)

[SetUserBuffers\(\)](#)

Returns

The size of one user memory buffer (in bytes)

Implements [ICameraBase](#).

15.10.3.19 GetUserBufferTotalSize()

```
uint64_t GetUserBufferTotalSize ( ) const [virtual]
```

GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).

This will throw an exception if user memory buffer has not been set. The total size should be [GetUserBufferSize\(\)](#) multiplied by [GetUserBufferCount\(\)](#) or larger.

See also

[GetUserBufferCount\(\)](#)

[GetUserBufferSize\(\)](#)

[SetUserBuffers\(\)](#)

Returns

The total size of all the user memory buffers (in bytes)

Implements [ICameraBase](#).

15.10.3.20 Init()

```
void Init ( ) [virtual]
```

Init Connects to camera with read and write access, retrieves remote device and GUI XML and generates node map references.

This function needs to be called before any camera related API calls such as [BeginAcquisition\(\)](#), [EndAcquisition\(\)](#), [GetNodeMap\(\)](#), [GetNextImage\(\)](#) and [GetNextImageSync\(\)](#).

See also

[BeginAcquisition\(\)](#)
[EndAcquisition\(\)](#)
[GetNodeMap\(\)](#)
[GetNextImage\(\)](#)
[GetNextImageSync\(\)](#)

Implements [ICameraBase](#).

15.10.3.21 IsInitialized()

```
bool IsInitialized ( ) [virtual]
```

IsInitialized Checks if camera is initialized.

This function needs to return true in order to retrieve a valid NodeMap from the [GetNodeMap\(\)](#) call.

See also

[GetNodeMap\(\)](#)

Returns

If camera is initialized or not

Implements [ICameraBase](#).

15.10.3.22 IsStreaming()

```
bool IsStreaming ( ) const [virtual]
```

IsStreaming Returns true if the camera is currently streaming or false if it is not.

See also

[Init\(\)](#)

Returns

returns true if camera is streaming and false otherwise.

Implements [ICameraBase](#).

15.10.3.23 IsValid()

```
bool IsValid ( ) [virtual]
```

IsValid Checks a flag to determine if camera is still valid for use.

Returns

If camera is valid or not

Implements [ICameraBase](#).

15.10.3.24 operator=()

```
CameraBase& operator= (
    const CameraBase & ) [protected]
```

Assignment operator.

15.10.3.25 ReadPort()

```
void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

15.10.3.26 RegisterEventHandler() [1/3]

```
void RegisterEventHandler (
    DeviceEventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [virtual]
```

[RegisterEventHandler\(DeviceEventHandler &, const GenICam::gcstring &\)](#) Registers a specific device event handler for the camera given a device event name.

The camera has to be initialized first with a call to [Init\(\)](#) before registering handlers for events.

See also

[Init\(\)](#)

[UnregisterEventHandler\(\)](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the camera
<i>eventName</i>	The event name to register

Implements [ICameraBase](#).

15.10.3.27 RegisterEventHandler() [2/3]

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

[RegisterEventHandler\(EventHandler &\)](#) Registers a generic device, image or image list event handler for the camera.

The camera has to be initialized first with a call to [Init\(\)](#) before registering handlers for events.

See also

[Init\(\)](#)
[UnregisterEventHandler\(\)](#)
[DeviceEventHandler](#)
[ImageEventHandler](#)
[ImageListEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the camera

Implements [ICameraBase](#).

15.10.3.28 RegisterEventHandler() [3/3]

```
void RegisterEventHandler (
    ImageEventHandler & evtHandlerToRegister,
    uint64_t streamIndex ) [virtual]
```

[RegisterEventHandler\(ImageEventHandler &, uint64_t\)](#) Registers a specific stream handler for the camera given a stream index.

The camera has to be initialized first with a call to [Init\(\)](#) before registering handlers for events.

See also

[Init\(\)](#)
[UnregisterEventHandler\(\)](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the camera
<i>streamIndex</i>	The index of the stream of where this handler will be registered to

Implements [ICameraBase](#).

15.10.3.29 SetBufferOwnership()

```
void SetBufferOwnership (
    const BufferOwnership mode ) [virtual]
```

SetBufferOwnership Sets data buffer ownership.

The data buffers can be owned by [System](#) or User. If the system owns the buffers, the memory required for the buffers are allocated and freed by the library. If user owns the buffers, the user is responsible for allocating and ultimately freeing the memory. By default, data buffers are owned by the library.

See also

[GetBufferOwnership\(\)](#)
[SetUserBuffers\(\)](#)

Parameters

<i>mode</i>	System owned or User owned buffers

Implements [ICameraBase](#).

15.10.3.30 SetUserBuffers() [1/2]

```
void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [virtual]
```

SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.

Each pointer to a buffer must have enough memory to hold one image. To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$ where 1024 is the USB3 packet size.

See also

[GetBufferOwnership\(\)](#)
[SetBufferOwnership\(\)](#)
[GetUserBufferCount\(\)](#)
[GetUserBufferSize\(\)](#)
[GetUserBufferTotalSize\(\)](#)

Parameters

<i>ppMemBuffers</i>	Pointer to pointers that each point to a single user memory buffer to be written to
<i>bufferCount</i>	The number of user memory buffers
<i>bufferSize</i>	The size of the memory allocated for each user buffer (in bytes)

Implements [ICameraBase](#).

15.10.3.31 SetUserBuffers() [2/2]

```
void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [virtual]
```

SetUserBuffers Specify contiguous user allocated memory to use as data buffers.

To prevent image tearing when working with USB3 cameras, the size of each buffer should be equal to: ((unsigned int) (bufferSize + 1024 - 1) / 1024) * 1024 where 1024 is the USB3 packet size.

See also

[GetBufferOwnership\(\)](#)
[SetBufferOwnership\(\)](#)
[GetUserBufferCount\(\)](#)
[GetUserBufferSize\(\)](#)
[GetUserBufferTotalSize\(\)](#)

Parameters

<i>pMemBuffers</i>	Pointer to memory buffers to be written to
<i>totalSize</i>	The total size of the memory allocated for the user buffers (in bytes)

Implements [ICameraBase](#).

15.10.3.32 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

UnregisterEventHandler Unregisters any type of event handler for the camera.

Event handlers should be unregistered first before calling camera [Delnit\(\)](#). Otherwise an exception will be thrown in the [Delnit\(\)](#) call and require the user to unregister event handlers before the camera can be re-initialized again.

See also

[Delnit\(\)](#)
[RegisterEventHandler\(\)](#)

Parameters

<code>evtHandlerToUnregister</code>	The event handler to unregister from the camera

Implements [ICameraBase](#).

15.10.3.33 WritePort()

```
void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

15.10.4 Friends And Related Function Documentation

15.10.4.1 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

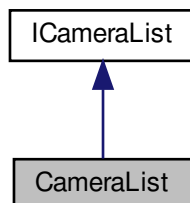
The documentation for this class was generated from the following file:

- include/[CameraBase.h](#)

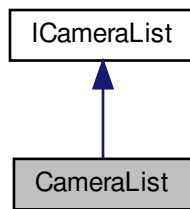
15.11 CameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for CameraList:



Collaboration diagram for CameraList:



Public Member Functions

- [CameraList](#) (void)
Default constructor.
- virtual [~CameraList](#) (void)
Virtual destructor.
- [CameraList](#) (const [CameraList](#) &iface)
Copy constructor.
- [CameraList](#) & [operator=](#) (const [CameraList](#) &iface)
Assignment operator.
- [CameraPtr](#) [operator\[\]](#) (unsigned int index)
Array subscription operators.
- unsigned int [GetSize](#) () const
Returns the size of the camera list.
- [CameraPtr](#) [GetByIndex](#) (unsigned int index) const
Returns a pointer to a camera object at the "index".
- [CameraPtr](#) [GetBySerial](#) (std::string serialNumber) const
Returns a pointer to a camera object with the specified serial number.
- [CameraPtr](#) [GetByDeviceID](#) (std::string deviceID) const
Returns a pointer to a camera object with the specified device identifier.
- void [Clear](#) ()
Clears the list of cameras and destroys their corresponding reference counted objects.
- void [Add](#) ([CameraPtr](#) camera)
Adds a copy of an camera object.
- void [Remove](#) ([CameraPtr](#) camera)
Removes all occurrences of an camera that is pointed to by the input camera pointer and destroys its corresponding reference counted object.
- void [RemoveByIndex](#) (unsigned int index)
Removes a camera at "index" and destroys its corresponding reference counted object.
- void [RemoveBySerial](#) (std::string serialNumber)
Removes a camera using its serial number and destroys its corresponding reference counted object.
- void [RemoveByDeviceID](#) (std::string deviceID)
Removes a camera using its unique device identifier and destroys its corresponding reference counted object.
- void [Append](#) (const [CameraList](#) &list)
Appends a copy of the camera list.

Additional Inherited Members

15.11.1 Detailed Description

Used to hold a list of camera objects.

15.11.2 Constructor & Destructor Documentation

15.11.2.1 CameraList() [1/2]

```
CameraList (  
    void )
```

Default constructor.

15.11.2.2 ~CameraList()

```
virtual ~CameraList (  
    void ) [virtual]
```

Virtual destructor.

15.11.2.3 CameraList() [2/2]

```
CameraList (  
    const CameraList & iface )
```

Copy constructor.

15.11.3 Member Function Documentation

15.11.3.1 Add()

```
void Add (  
    CameraPtr camera ) [virtual]
```

Adds a copy of an camera object.

Parameters

<i>camera</i>	A camera object to be added to this list.

Implements [ICameraList](#).

15.11.3.2 Append()

```
void Append (
    const CameraList & list ) [virtual]
```

Appends a copy of the camera list.

Parameters

<i>list</i>	Another CameraList object, whose elements are added to this list.

Implements [ICameraList](#).

15.11.3.3 Clear()

```
void Clear ( ) [virtual]
```

Clears the list of cameras and destroys their corresponding reference counted objects.

This is necessary in order to clean up the parent interface. It is important that the camera list is destroyed or is cleared before calling `system->ReleaseInstance()` or else the call to `system->ReleaseInstance()` will result in an error message thrown that a reference to the camera is still held.

See also

[System:ReleaseInstance\(\)](#)

Implements [ICameraList](#).

15.11.3.4 GetByDeviceID()

```
CameraPtr GetByDeviceID (
    std::string deviceID ) const [virtual]
```

Returns a pointer to a camera object with the specified device identifier.

This function will return a NULL [CameraPtr](#) if no matching device identifier is found.

Parameters

<i>deviceId</i>	The unique device identifier of the camera object to retrieve

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.11.3.5 GetByIndex()

```
CameraPtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to a camera object at the "index".

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to retrieve the camera object

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.11.3.6 GetBySerial()

```
CameraPtr GetBySerial (
    std::string serialNumber ) const [virtual]
```

Returns a pointer to a camera object with the specified serial number.

This function will return a NULL [CameraPtr](#) if no matching camera serial is found.

Parameters

<i>serialNumber</i>	The serial number of the camera object to retrieve

Returns

A pointer to a camera object.

Implements [ICameraList](#).

15.11.3.7 GetSize()

```
unsigned int GetSize ( ) const [virtual]
```

Returns the size of the camera list.

The size is the number of [Camera](#) objects stored in the list.

Returns

An integer that represents the list size.

Implements [ICameraList](#).

15.11.3.8 operator=()

```
CameraList& operator= (
    const CameraList & iface )
```

Assignment operator.

15.11.3.9 operator[]()

```
CameraPtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [ICameraList](#).

15.11.3.10 Remove()

```
void Remove (
    CameraPtr camera ) [virtual]
```

Removes all occurrences of a camera that is pointed to by the input camera pointer and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with `SPINNAKER_ERR_NOT_AVAILABLE` error if no matching camera is found.

Parameters

<i>camera</i>	Pointer to the camera that is to be removed

Implements [ICameraList](#).

15.11.3.11 RemoveByDeviceID()

```
void RemoveByDeviceID (
    std::string deviceID ) [virtual]
```

Removes a camera using its unique device identifier and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching device identifier is found.

Parameters

<i>deviceID</i>	The unique device identifier of the camera object to retrieve

Implements [ICameraList](#).

15.11.3.12 RemoveByIndex()

```
void RemoveByIndex (
    unsigned int index ) [virtual]
```

Removes a camera at "index" and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to remove the Camera object

Implements [ICameraList](#).

15.11.3.13 RemoveBySerial()

```
void RemoveBySerial (
    std::string serialNumber ) [virtual]
```

Removes a camera using its serial number and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching camera serial is found.

Parameters

<i>serialNumber</i>	The serial number of the Camera object to remove

Implements [ICameraList](#).

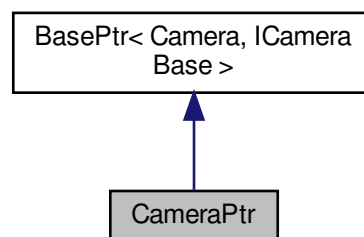
The documentation for this class was generated from the following file:

- [include/CameraList.h](#)

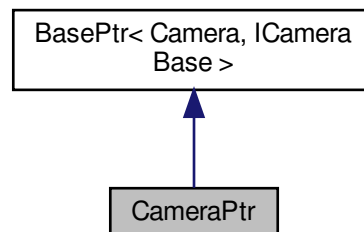
15.12 CameraPtr Class Reference

A reference tracked pointer to a camera object.

Inheritance diagram for CameraPtr:



Collaboration diagram for CameraPtr:



Public Member Functions

- [CameraPtr](#) () throw ()
Default constructor.
- [CameraPtr](#) (const int) throw ()
Default constructor.
- [CameraPtr](#) (const long) throw ()
Default constructor with argument.
- [CameraPtr](#) (const std::nullptr_t) throw ()

Additional Inherited Members

15.12.1 Detailed Description

A reference tracked pointer to a camera object.

15.12.2 Constructor & Destructor Documentation

15.12.2.1 CameraPtr() [1/4]

```
CameraPtr ( ) throw ( )    [inline]
```

Default constructor.

15.12.2.2 CameraPtr() [2/4]

```
CameraPtr (
    const int ) throw ( )    [inline]
```

Default constructor.

15.12.2.3 CameraPtr() [3/4]

```
CameraPtr (
    const long ) throw ( )    [inline]
```

Default constructor with argument.

15.12.2.4 CameraPtr() [4/4]

```
CameraPtr (
    const std::nullptr_t ) throw ( )    [inline]
```

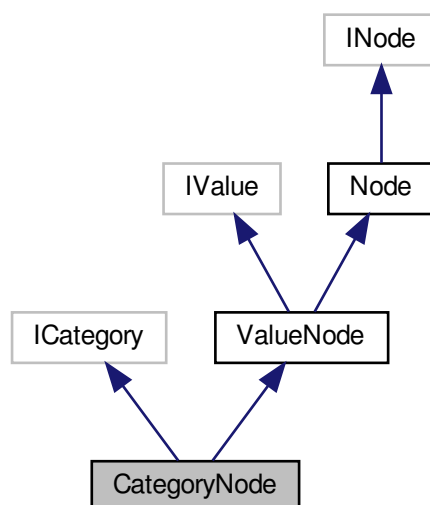
The documentation for this class was generated from the following file:

- include/[CameraPtr.h](#)

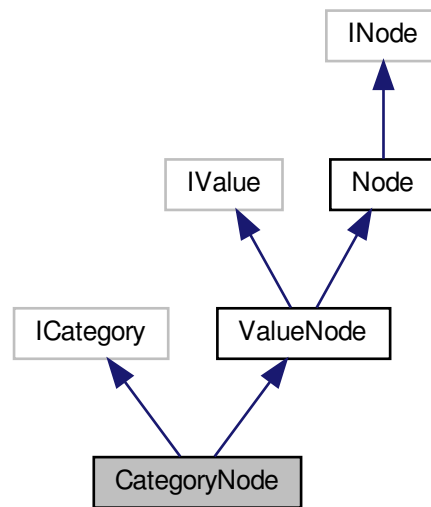
15.13 CategoryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CategoryNode:



Collaboration diagram for CategoryNode:



Public Member Functions

- [CategoryNode](#) ()
- [CategoryNode](#) (std::shared_ptr< Node::NodeImpl > pCategory)
- virtual [~CategoryNode](#) ()
- virtual void [GetFeatures](#) (FeatureList_t &Features) const
Get all features of the category (including sub-categories)
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.13.1 Detailed Description

[Interface](#) for string properties.

15.13.2 Constructor & Destructor Documentation

15.13.2.1 CategoryNode() [1/2]

[CategoryNode](#) ()

15.13.2.2 CategoryNode() [2/2]

```
CategoryNode (
    std::shared_ptr< Node::NodeImpl > pCategory )
```

15.13.2.3 ~CategoryNode()

```
virtual ~CategoryNode ( ) [virtual]
```

15.13.3 Member Function Documentation

15.13.3.1 GetFeatures()

```
virtual void GetFeatures (
    FeatureList_t & Features ) const [virtual]
```

Get all features of the category (including sub-categories)

15.13.3.2 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

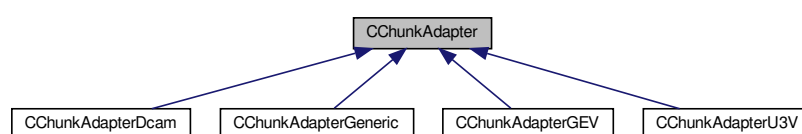
The documentation for this class was generated from the following file:

- include/SpinGenApi/[CategoryNode.h](#)

15.14 CChunkAdapter Class Reference

Connects a chunked buffer to a node map.

Inheritance diagram for CChunkAdapter:



Public Member Functions

- virtual `~CChunkAdapter ()`
Destructor.
- void `AttachNodeMap (INodeMap *pNodeMap)`
Attaches to a node map and retrieves the chunk ports.
- void `DetachNodeMap ()`
Detaches from the node map.
- virtual bool `CheckBufferLayout (uint8_t *pBuffer, int64_t BufferLength)=0`
Checks if a buffer contains chunks in a known format.
- virtual void `AttachBuffer (uint8_t *pBuffer, int64_t BufferLength, AttachStatistics_t *pAttachStatistics=NULL)=0`
Attaches a buffer to the matching ChunkPort.
- void `DetachBuffer ()`
Detaches a buffer.
- void `UpdateBuffer (uint8_t *pBaseAddress)`
Updates the base address of the buffer.
- void `ClearCaches ()`
Clears the chunk caches.

Protected Member Functions

- `CChunkAdapter (INodeMap *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)`
Serves as default constructor.

Protected Attributes

- void * `m_pChunkAdapter`

15.14.1 Detailed Description

Connects a chunked buffer to a node map.

15.14.2 Constructor & Destructor Documentation

15.14.2.1 ~CChunkAdapter()

```
virtual ~CChunkAdapter ( ) [virtual]
```

Destructor.

15.14.2.2 CChunkAdapter()

```
CChunkAdapter (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 ) [protected]
```

Serves as default constructor.

15.14.3 Member Function Documentation

15.14.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [pure virtual]
```

Attaches a buffer to the matching ChunkPort.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

15.14.3.2 AttachNodeMap()

```
void AttachNodeMap (
    INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

15.14.3.3 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [pure virtual]
```

Checks if a buffer contains chunks in a known format.

Implemented in [CChunkAdapterDcam](#), [CChunkAdapterGeneric](#), [CChunkAdapterGEV](#), and [CChunkAdapterU3V](#).

15.14.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

15.14.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

15.14.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

15.14.3.7 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

15.14.4 Member Data Documentation

15.14.4.1 m_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

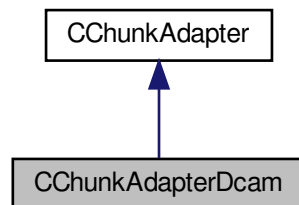
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapter.h](#)

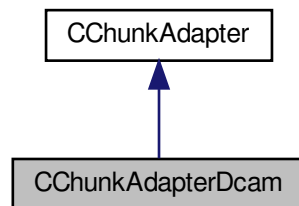
15.15 CChunkAdapterDcam Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterDcam:



Collaboration diagram for CChunkAdapterDcam:



Public Member Functions

- [CChunkAdapterDcam](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterDcam](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.
- bool [HasCRC](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if buffer has a CRC attached.
- bool [CheckCRC](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks CRC sum of buffer.

Additional Inherited Members

15.15.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

15.15.2 Constructor & Destructor Documentation

15.15.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

15.15.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam ( ) [virtual]
```

Destructor.

15.15.3 Member Function Documentation

15.15.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.15.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

15.15.3.3 CheckCRC()

```
bool CheckCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

Checks CRC sum of buffer.

15.15.3.4 HasCRC()

```
bool HasCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

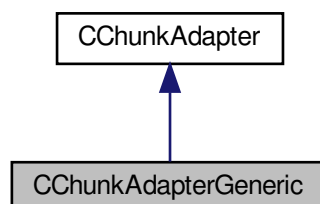
Checks if buffer has a CRC attached.

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

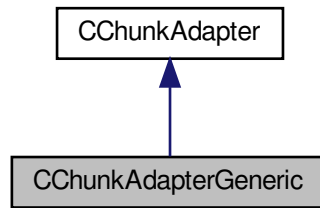
- include/SpinGenApi/[ChunkAdapterDcam.h](#)

15.16 CChunkAdapterGeneric Class Reference

Inheritance diagram for CChunkAdapterGeneric:



Collaboration diagram for CChunkAdapterGeneric:



Public Member Functions

- [CChunkAdapterGeneric](#) ([INodeMap](#) *pNodeMap=NULL, [int64_t](#) MaxChunkCacheSize=-1)
- virtual [~CChunkAdapterGeneric](#) ()
- virtual bool [CheckBufferLayout](#) ([uint8_t](#) *pBuffer, [int64_t](#) BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) ([uint8_t](#) *pBuffer, [int64_t](#) BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.
- virtual void [AttachBuffer](#) ([uint8_t](#) *pBuffer, [SingleChunkData_t](#) *ChunkData, [int64_t](#) NumChunks, [AttachStatistics_t](#) *pAttachStatistics=NULL)
- virtual void [AttachBuffer](#) ([uint8_t](#) *pBuffer, [SingleChunkDataStr_t](#) *ChunkData, [int64_t](#) NumChunks, [AttachStatistics_t](#) *pAttachStatistics=NULL)

Additional Inherited Members

15.16.1 Constructor & Destructor Documentation

15.16.1.1 CChunkAdapterGeneric()

```

CChunkAdapterGeneric (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
  
```

15.16.1.2 ~CChunkAdapterGeneric()

```

virtual ~CChunkAdapterGeneric ( ) [virtual]
  
```

15.16.2 Member Function Documentation

15.16.2.1 AttachBuffer() [1/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.16.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkData_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

15.16.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkDataStr_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

15.16.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

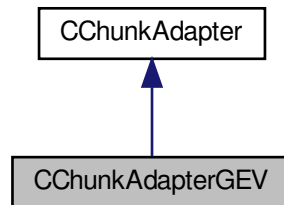
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

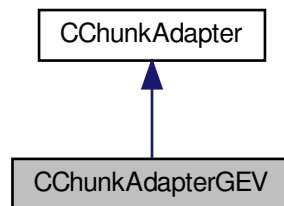
15.17 CChunkAdapterGEV Class Reference

Connects a chunked DCAM buffer to a node map.

Inheritance diagram for CChunkAdapterGEV:



Collaboration diagram for CChunkAdapterGEV:



Public Member Functions

- [CChunkAdapterGEV](#) ([INodeMap](#) *pNodeMap=NULL, [int64_t](#) MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterGEV](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) ([uint8_t](#) *pBuffer, [int64_t](#) BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) ([uint8_t](#) *pBuffer, [int64_t](#) BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

15.17.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

15.17.2 Constructor & Destructor Documentation

15.17.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

15.17.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( ) [virtual]
```

Destructor.

15.17.3 Member Function Documentation

15.17.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.17.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

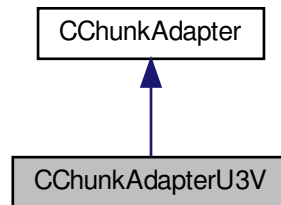
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

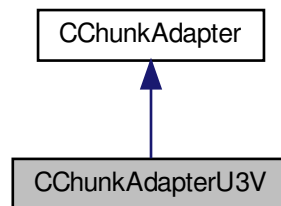
15.18 CChunkAdapterU3V Class Reference

Connects a chunked U3V buffer to a node map.

Inheritance diagram for CChunkAdapterU3V:



Collaboration diagram for CChunkAdapterU3V:



Public Member Functions

- [CChunkAdapterU3V](#) ([INodeMap](#) *pNodeMap=NULL, int64_t MaxChunkCacheSize=-1)
Constructor.
- virtual [~CChunkAdapterU3V](#) ()
Destructor.
- virtual bool [CheckBufferLayout](#) (uint8_t *pBuffer, int64_t BufferLength)
Checks if a buffer contains chunks in a known format.
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, int64_t BufferLength, [AttachStatistics_t](#) *pAttachStatistics=NULL)
Attaches a buffer to the matching ChunkPort.

Additional Inherited Members

15.18.1 Detailed Description

Connects a chunked U3V buffer to a node map.

15.18.2 Constructor & Destructor Documentation

15.18.2.1 CChunkAdapterU3V()

```
CChunkAdapterU3V (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

15.18.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

15.18.3 Member Function Documentation

15.18.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

15.18.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

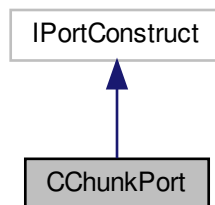
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

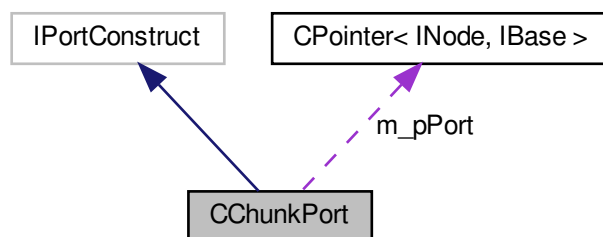
15.19 CChunkPort Class Reference

Port attachable to a chunk in a buffer.

Inheritance diagram for CChunkPort:



Collaboration diagram for CChunkPort:



Public Member Functions

- [CChunkPort](#) ([IPort](#) *pPort=NULL)
Constructor; can attach to a port.
- [~CChunkPort](#) ()
Destructor; detaches from the port.
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, [int64_t](#) [Address](#), [int64_t](#) [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, [int64_t](#) [Address](#), [int64_t](#) [Length](#))
Writes a chunk of bytes to the port.

- virtual void [SetPortImpl](#) (IPort *pPort)
Called from the port node to give the chunk port a pointer to itself.
- virtual EYesNo [GetSwapEndianess](#) ()
Determines if the port adapter must perform an endianness swap.
- void [InvalidateNode](#) ()
- bool [AttachPort](#) (::Spinnaker::GenApi::IPort *pPort)
Attaches the ChunkPort to the Port.
- void [DetachPort](#) ()
Detaches the ChunkPort to the Port.
- void [AttachChunk](#) (uint8_t *pBaseAddress, int64_t ChunkOffset, int64_t [Length](#), bool Cache)
Attaches the Chunk to the ChunkPort.
- void [DetachChunk](#) ()
Detaches the Chunk from the ChunkPort.
- int [GetChunkIDLength](#) ()
Gets the ChunkID length.
- bool [CheckChunkID](#) (uint8_t *pChunkIDBuffer, int ChunkIDLength)
Checks if a ChunkID matches.
- bool [CheckChunkID](#) (uint64_t ChunkID)
Checks if a ChunkID matches, version using uint64_t ID representation.
- void [UpdateBuffer](#) (uint8_t *pBaseAddress)
Updates the base address of the chunk.
- void [ClearCache](#) ()
Clears the chunk cache.

Protected Attributes

- [CNodePtr](#) [m_pPort](#)
- std::shared_ptr< PortAdapter > [m_pPortAdapter](#)
- void * [m_pChunkPort](#)

15.19.1 Detailed Description

Port attachable to a chunk in a buffer.

15.19.2 Constructor & Destructor Documentation

15.19.2.1 CChunkPort()

```
CChunkPort (
    IPort * pPort = NULL )
```

Constructor; can attach to a port.

15.19.2.2 ~CChunkPort()

`~CChunkPort ( )`

Destructor; detaches from the port.

15.19.3 Member Function Documentation

15.19.3.1 AttachChunk()

```
void AttachChunk (
    uint8_t * pBaseAddress,
    int64_t ChunkOffset,
    int64_t Length,
    bool Cache )
```

Attaches the Chunk to the ChunkPort.

15.19.3.2 AttachPort()

```
bool AttachPort (
    ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

15.19.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
    uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64_t ID representation.

15.19.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
    uint8_t * pChunkIDBuffer,
    int ChunkIDLength )
```

Checks if a ChunkID matches.

15.19.3.5 ClearCache()

```
void ClearCache ( )
```

Clears the chunk cache.

15.19.3.6 DetachChunk()

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

15.19.3.7 DetachPort()

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

15.19.3.8 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

15.19.3.9 GetChunkIDLength()

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

15.19.3.10 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.19.3.11 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.19.3.12 InvalidateNode()

```
void InvalidateNode ( )
```

15.19.3.13 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

15.19.3.14 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

15.19.3.15 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

15.19.3.16 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.19.4 Member Data Documentation

15.19.4.1 m_pChunkPort

`void* m_pChunkPort [protected]`

15.19.4.2 m_pPort

`CNodePtr m_pPort [protected]`

15.19.4.3 m_pPortAdapter

`std::shared_ptr<PortAdapter> m_pPortAdapter [protected]`

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

- `include/SpinGenApi/ChunkPort.h`

15.20 CCMSettings Struct Reference

Public Member Functions

- `CCMSettings ()`

Public Attributes

- `CCMColorTemperature ColorTemperature`
- `CCMType Type`
- `CCMSensor Sensor`
- `std::string CustomCCMCode`
- `CCMColorSpace ColorSpace`
- `CCMApplication Application`

15.20.1 Constructor & Destructor Documentation

15.20.1.1 CCMSettings()

`CCMSettings` () [inline]

15.20.2 Member Data Documentation

15.20.2.1 Application

`CCMApplication` Application

15.20.2.2 ColorSpace

`CCMColorSpace` ColorSpace

15.20.2.3 ColorTemperature

`CCMColorTemperature` ColorTemperature

15.20.2.4 CustomCCMCode

`std::string` CustomCCMCode

15.20.2.5 Sensor

`CCMSensor` Sensor

15.20.2.6 Type

`CCMType` Type

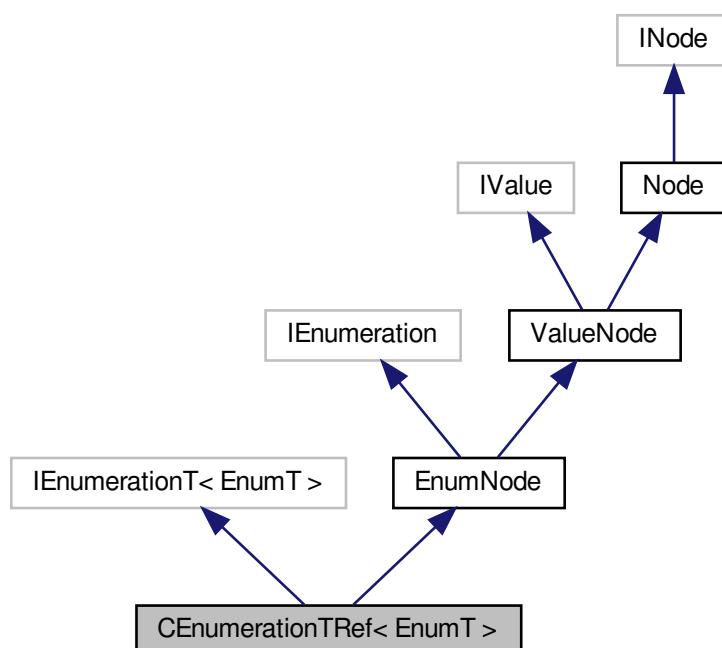
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

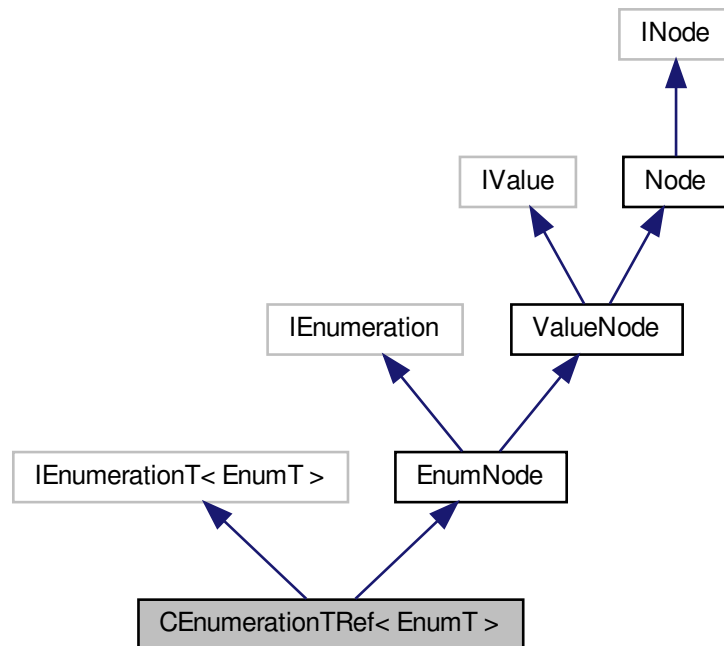
15.21 CEnumerationTRef< EnumT > Class Template Reference

[Interface](#) for string properties.

Inheritance diagram for CEnumerationTRef< EnumT >:



Collaboration diagram for CEnumerationTRef< EnumT >:



Public Member Functions

- `CEnumerationTRef` ()
- `CEnumerationTRef` (std::shared_ptr< Node::NodeImpl > pEnumeration)
- virtual `~CEnumerationTRef` ()
- virtual void `SetValue` (EnumT Value, bool `Verify`=true)
 - Set node value.*
- virtual `IEnumeration` & `operator=` (EnumT Value)
 - Set node value.*
- virtual EnumT `GetValue` (bool `Verify`=false, bool IgnoreCache=false)
 - Get node value.*
- virtual EnumT `operator()` ()
 - Get node value.*
- virtual `IEnumeration` & `operator=` (const `GenlCam::gcstring` &ValueStr)
 - Set node value.*
- virtual `IEnumEntry` * `GetEntry` (const EnumT Value)
 - returns the EnumEntry object belonging to the Value*
- virtual `IEnumEntry` * `GetEntry` (const int64_t IntValue)
 - Get an entry node by its IntValue.*
- virtual `IEnumEntry` * `GetCurrentEntry` (bool `Verify`=false, bool IgnoreCache=false)
 - Get the current entry.*
- virtual void `SetReference` (INode *pBase)
 - overload SetReference for EnumerationT*

- virtual void [SetEnumReference](#) (int Index, [GenICam::gcstring](#) Name)
sets the Enum value corresponding to a value
- virtual void [SetNumEnums](#) (int NumEnums)
sets the number of enum values

Additional Inherited Members

15.21.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

15.21.2 Constructor & Destructor Documentation

15.21.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

15.21.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

15.21.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( ) [virtual]
```

15.21.3 Member Function Documentation

15.21.3.1 GetCurrentEntry()

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented from [EnumNode](#).

15.21.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
    const EnumT Value ) [virtual]
```

returns the EnumEntry object belonging to the Value

15.21.3.3 GetEntry() [2/2]

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

15.21.3.4 GetValue()

```
virtual EnumT GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.21.3.5 operator>()

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

15.21.3.6 operator=() [1/2]

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set node value.

Note : the operator= is not inherited thus the operator= versions from IEnumeration must be implemented again

Reimplemented from [EnumNode](#).

15.21.3.7 operator=() [2/2]

```
virtual IEnumeration& operator= (
    EnumT Value ) [virtual]
```

Set node value.

15.21.3.8 SetEnumReference()

```
virtual void SetEnumReference (
    int Index,
    GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

15.21.3.9 SetNumEnums()

```
virtual void SetNumEnums (
    int NumEnums ) [virtual]
```

sets the number of enum values

15.21.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

15.21.3.11 SetValue()

```
virtual void SetValue (
    EnumT Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

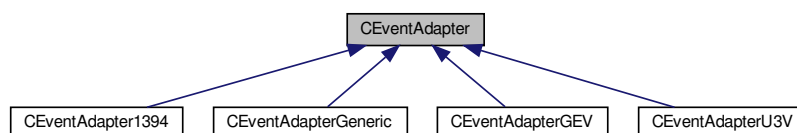
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumNodeT.h](#)

15.22 CEventAdapter Class Reference

Delivers Events to ports.

Inheritance diagram for CEventAdapter:

**Public Member Functions**

- [CEventAdapter](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapter](#) ()

Destructor.

- virtual void [AttachNodeMap](#) ([INodeMap](#) *pNodeMap)
Attaches to a node map and retrieves the chunk ports.
- virtual void [DetachNodeMap](#) ()
Detaches from the node map.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)=0
Deliver message.

Protected Attributes

- void * [m_pEventAdapter](#)

15.22.1 Detailed Description

Delivers Events to ports.

15.22.2 Constructor & Destructor Documentation

15.22.2.1 CEventAdapter()

```
CEventAdapter (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.22.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( ) [virtual]
```

Destructor.

15.22.3 Member Function Documentation

15.22.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
    INodeMap * pNodeMap ) [virtual]
```

Attaches to a node map and retrieves the chunk ports.

15.22.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [pure virtual]
```

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

15.22.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( ) [virtual]
```

Detaches from the node emap.

15.22.4 Member Data Documentation

15.22.4.1 m_pEventAdapter

```
void* m_pEventAdapter [protected]
```

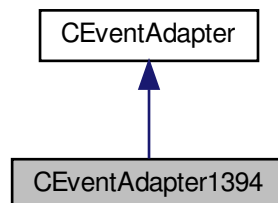
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter.h](#)

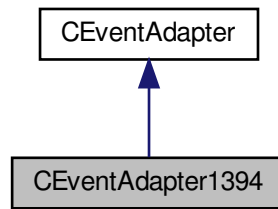
15.23 CEventAdapter1394 Class Reference

Distribute the events to the node map.

Inheritance diagram for CEventAdapter1394:



Collaboration diagram for CEventAdapter1394:



Public Member Functions

- [CEventAdapter1394](#) ([INodeMap](#) *pNodeMap=NULL)
constructor
- virtual [~CEventAdapter1394](#) ()
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (EventData1394 &Event, uint32_t numBytes)
distributes events to node map

Additional Inherited Members

15.23.1 Detailed Description

Distribute the events to the node map.

15.23.2 Constructor & Destructor Documentation

15.23.2.1 CEventAdapter1394()

```

CEventAdapter1394 (
    INodeMap * pNodeMap = NULL ) [explicit]
  
```

constructor

15.23.2.2 ~CEventAdapter1394()

```

virtual ~CEventAdapter1394 ( ) [virtual]
  
```

15.23.3 Member Function Documentation

15.23.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    EventData1394 & Event,
    uint32_t numBytes )
```

distributes events to node map

15.23.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

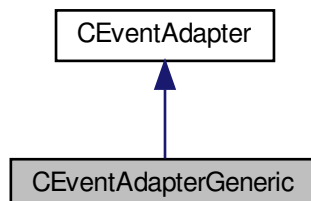
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

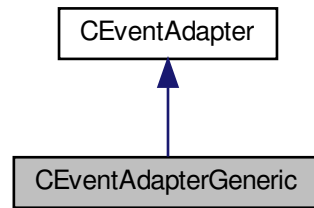
15.24 CEventAdapterGeneric Class Reference

Connects a generic event to a node map.

Inheritance diagram for CEventAdapterGeneric:



Collaboration diagram for CEventAdapterGeneric:



Public Member Functions

- [CEventAdapterGeneric](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGeneric](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, const [GenICam::gcstring](#) &EventID)
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, uint64_t EventID)

Additional Inherited Members

15.24.1 Detailed Description

Connects a generic event to a node map.

15.24.2 Constructor & Destructor Documentation

15.24.2.1 CEventAdapterGeneric()

```
CEventAdapterGeneric (  
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.24.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

15.24.3 Member Function Documentation

15.24.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

15.24.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    const GenICam::gcstring & EventID ) [virtual]
```

15.24.3.3 DeliverMessage() [3/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    uint64_t EventID ) [virtual]
```

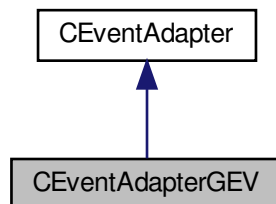
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGeneric.h](#)

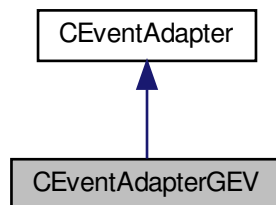
15.25 CEventAdapterGEV Class Reference

Connects a GigE Event to a node map.

Inheritance diagram for CEventAdapterGEV:



Collaboration diagram for CEventAdapterGEV:



Public Member Functions

- [CEventAdapterGEV](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGEV](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [GVCP_EVENT_REQUEST](#) *pEvent)
Delivers the Events listed in the Event packet.
- void [DeliverEventMessage](#) (const [GVCP_EVENTDATA_REQUEST](#) *pEventData)
Delivers the Event + Data listed in the EventData packet.

Additional Inherited Members

15.25.1 Detailed Description

Connects a GigE Event to a node map.

15.25.2 Constructor & Destructor Documentation

15.25.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.25.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( ) [virtual]
```

Destructor.

15.25.3 Member Function Documentation

15.25.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (
    const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the Event packet.

15.25.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (
    const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the Event + Data listed in the EventData packet.

15.25.3.3 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

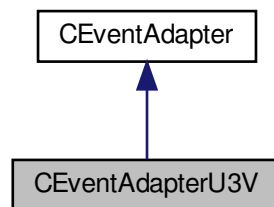
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

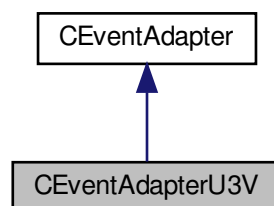
15.26 CEventAdapterU3V Class Reference

Connects a U3V Event to a node map.

Inheritance diagram for CEventAdapterU3V:



Collaboration diagram for CEventAdapterU3V:



Public Member Functions

- [CEventAdapterU3V](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterU3V](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [U3V_EVENT_MESSAGE](#) *pEventMessage)
Delivers the Event + Data listed in the packet.

Additional Inherited Members

15.26.1 Detailed Description

Connects a U3V Event to a node map.

15.26.2 Constructor & Destructor Documentation

15.26.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
    INodeMap * pNodeMap = NULL )
```

Constructor.

15.26.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( ) [virtual]
```

Destructor.

15.26.3 Member Function Documentation

15.26.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    const U3V\_EVENT\_MESSAGE * pEventMessage )
```

Delivers the Event + Data listed in the packet.

15.26.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

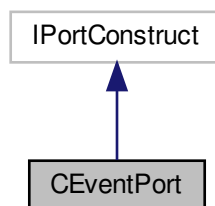
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

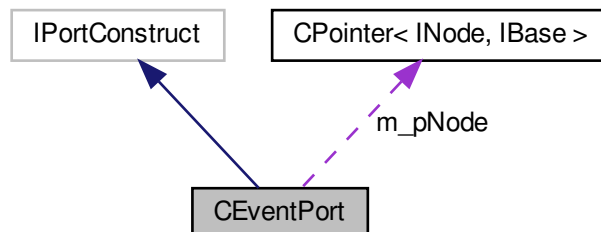
15.27 CEventPort Class Reference

Port attachable to an event.

Inheritance diagram for CEventPort:



Collaboration diagram for CEventPort:



Public Member Functions

- [CEventPort](#) (INode *pNode=NULL)
Constructor; can attach to a node.
- [~CEventPort](#) ()
Destructor; detaches from the port.
- virtual EAccessMode [GetAccessMode](#) () const
Get the access mode of the node.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t Address, int64_t Length)
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t Address, int64_t Length)
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (::Spinnaker::GenApi::IPort *pPort)
Called from the port node to give the chunk port a pointer to itself.
- virtual EYesNo [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- void [InvalidateNode](#) ()
- bool [AttachNode](#) (::Spinnaker::GenApi::INode *pNode)
Attaches to the Node.
- void [DetachNode](#) ()
Detaches from the Node.
- int [GetEventIDLength](#) ()
Gets the EventID length.
- bool [CheckEventID](#) (uint8_t *pEventIDBuffer, int EventIDLength)
Checks if a EventID matches.
- bool [CheckEventID](#) (uint64_t EventID)
Checks if a EventID matches, version using uint64_t ID representation.
- void [AttachEvent](#) (uint8_t *pBaseAddress, int64_t Length)
Attaches the an Event to the EventPort.
- void [DetachEvent](#) ()
Detaches the Event from the EventPort.

Protected Attributes

- [CNodePtr](#) m_pNode
- std::shared_ptr< PortAdapter > m_pPortAdapter
- void * m_pEventPort

15.27.1 Detailed Description

Port attachable to an event.

15.27.2 Constructor & Destructor Documentation

15.27.2.1 CEventPort()

```
CEventPort (
    INode * pNode = NULL )
```

Constructor; can attach to a node.

15.27.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

15.27.3 Member Function Documentation

15.27.3.1 AttachEvent()

```
void AttachEvent (
    uint8_t * pBaseAddress,
    int64_t Length )
```

Attaches the an Event to the EventPort.

15.27.3.2 AttachNode()

```
bool AttachNode (
    ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the [Node](#).

15.27.3.3 CheckEventID() [1/2]

```
bool CheckEventID (
    uint64_t EventID )
```

Checks if a EventID matches, version using uint64_t ID representation.

15.27.3.4 CheckEventID() [2/2]

```
bool CheckEventID (
    uint8_t * pEventIDBuffer,
    int EventIDLength )
```

Checks if a EventID matches.

15.27.3.5 DetachEvent()

```
void DetachEvent ( )
```

Detaches the Event from the EventPort.

15.27.3.6 DetachNode()

```
void DetachNode ( )
```

Detaches from the [Node](#).

15.27.3.7 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

15.27.3.8 GetEventIDLength()

```
int GetEventIDLength ( )
```

Gets the EventID length.

15.27.3.9 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.27.3.10 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.27.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

15.27.3.12 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

15.27.3.13 SetPortImpl()

```
virtual void SetPortImpl (
    ::Spinnaker::GenApi::IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

15.27.3.14 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.27.4 Member Data Documentation

15.27.4.1 m_pEventPort

```
void* m_pEventPort [protected]
```

15.27.4.2 m_pNode

```
CNodePtr m_pNode [protected]
```

15.27.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

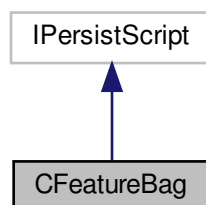
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventPort.h](#)

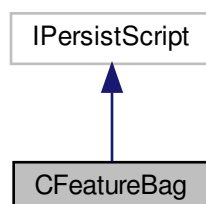
15.28 CFeatureBag Class Reference

Bag holding streamable features of a nodetree.

Inheritance diagram for CFeatureBag:



Collaboration diagram for CFeatureBag:



Public Member Functions

- [CFeatureBag](#) ()
- virtual [~CFeatureBag](#) ()
- virtual void [SetInfo](#) ([GenICam::gcstring](#) &Info)
sets information about the node map
- virtual void [PersistFeature](#) ([IValue](#) &item)
Stores a feature.
- bool [LoadFromBag](#) ([INodeMap](#) *pNodeMap, bool [Verify](#)=true, [GenICam::gcstring_vector](#) *pErrorList=NULL)
Loads the features from the bag to the node tree.
- int64_t [StoreToBag](#) ([INodeMap](#) *pNodeMap, const int MaxNumPersistSkriptEntries=-1)
Stores the streamable nodes to this feature bag.
- bool [operator==](#) (const [CFeatureBag](#) &FeatureBag) const
compares the content of two feature bags
- void * [GetFeatureBagHandle](#) ()

15.28.1 Detailed Description

Bag holding streamable features of a nodetree.

15.28.2 Constructor & Destructor Documentation

15.28.2.1 CFeatureBag()

```
CFeatureBag ( )
```

15.28.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( ) [virtual]
```

15.28.3 Member Function Documentation

15.28.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

15.28.3.2 LoadFromBag()

```
bool LoadFromBag (
    INodeMap * pNodeMap,
    bool Verify = true,
    GenICam::gcstring\_vector * pErrorList = NULL )
```

Loads the features from the bag to the node tree.

Parameters

<i>pNodeMap</i>	The node map
<i>Verify</i>	If true, all streamable features are read back
<i>pErrorList</i>	If an error occurs during loading the error message is stored in the list and the loading continues

For Verify=true the list of names in the feature bag is replayed again. If a node is a selector it's value is set to the value from the feature bag. If not the value is read from the camera and compared with the value from the feature bag.

15.28.3.3 operator==()

```
bool operator== (
    const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

15.28.3.4 PersistFeature()

```
virtual void PersistFeature (
    IValue & item ) [virtual]
```

Stores a feature.

15.28.3.5 SetInfo()

```
virtual void SetInfo (
    GenICam::gcstring & Info ) [virtual]
```

sets information about the node map

15.28.3.6 StoreToBag()

```
int64_t StoreToBag (
    INodeMap * pNodeMap,
    const int MaxNumPersistSkriptEntries = -1 )
```

Stores the streamable nodes to this feature bag.

Parameters

<i>pNodeMap</i>	The node map to persist
<i>MaxNumPersistSkriptEntries</i>	The max number of entries in the container; -1 means unlimited

Returns

number of entries in the bag

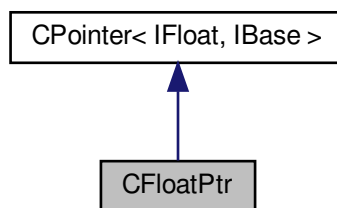
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Persistence.h](#)

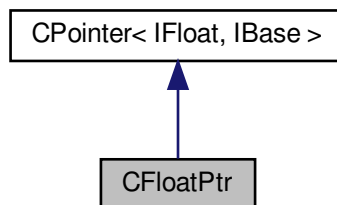
15.29 CFloatPtr Class Reference

SmartPointer for IFloat interface pointer.

Inheritance diagram for CFloatPtr:



Collaboration diagram for CFloatPtr:



Public Member Functions

- [CFloatPtr](#) () throw ()
Default constructor.
- [CFloatPtr](#) (IBase *pB)
Constructor from IBase pointer type.
- void [operator=](#) (IBase *pB)
Assign IBase Pointer.
- [Integer](#) * [GetIntAlias](#) ()
gets the interface of an integer alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an enum alias node.

Additional Inherited Members

15.29.1 Detailed Description

SmartPointer for IFloat interface pointer.

15.29.2 Constructor & Destructor Documentation

15.29.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw ( ) [inline]
```

Default constructor.

15.29.2.2 CFloatPtr() [2/2]

```
CFloatPtr (
    IBase * pB ) [inline]
```

Constructor from IBase pointer type.

15.29.3 Member Function Documentation

15.29.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( ) [inline]
```

gets the interface of an enum alias node.

15.29.3.2 GetIntAlias()

```
IInteger* GetIntAlias ( ) [inline]
```

gets the interface of an integer alias node.

15.29.3.3 operator=()

```
void operator= (
    IBase * pB ) [inline]
```

Assign IBase Pointer.

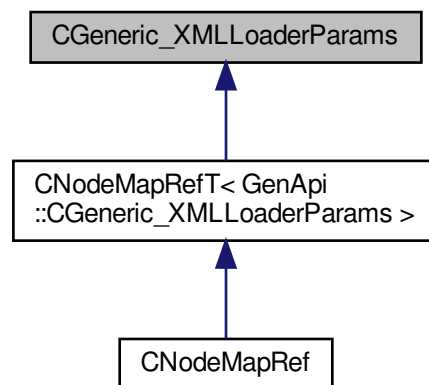
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

15.30 CGeneric_XMLLoaderParams Class Reference

Empty base class used by class [CNodeMapRef](#) as generic template argument.

Inheritance diagram for CGeneric_XMLLoaderParams:



Protected Member Functions

- virtual void [_Initialize](#) ([GenApi::INodeMap](#) *)

15.30.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

15.30.2 Member Function Documentation

15.30.2.1 _Initialize()

```
virtual void _Initialize (
    GenApi::INodeMap * ) [inline], [protected], [virtual]
```

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

- include/SpinGenApi/NodeMapRef.h

15.31 CGlobalLock Class Reference

Named global lock which can be used over process boundaries.

Public Member Functions

- [CGlobalLock](#) (const char *pszName)
Creates a global lock object name pszName.
- [CGlobalLock](#) (const [gcstring](#) &strName)
Creates a global lock object name strName.
- [~CGlobalLock](#) ()
- bool [IsValid](#) (void) const
tests whether the lock is valid
- bool [Lock](#) (unsigned int timeout_ms)
enters the lock (may block)
- bool [TryLock](#) (void)
tries to enter the lock and returns immediately when not possible
- void [Unlock](#) (void)
leaves the lock

Protected Attributes

- long [m_DebugCount](#)

15.31.1 Detailed Description

Named global lock which can be used over process boundaries.

15.31.2 Constructor & Destructor Documentation

15.31.2.1 CGlobalLock() [1/2]

```
CGlobalLock (
    const char * pszName ) [explicit]
```

Creates a global lock object name pszName.

In case an object with the same name already exists a reference to the existing object will be created. If pszName is NULL an unnamed object will be created.

15.31.2.2 CGlobalLock() [2/2]

```
CGlobalLock (
    const gcstring & strName ) [explicit]
```

Creates a global lock object name strName.

In case an object with the same name already exists a reference to the existing object will be created. If strName is empty an unnamed object will be created.

15.31.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

15.31.3 Member Function Documentation

15.31.3.1 IsValid()

```
bool IsValid (
    void ) const
```

tests whether the lock is valid

15.31.3.2 Lock()

```
bool Lock (
    unsigned int timeout_ms )
```

enters the lock (may block)

15.31.3.3 TryLock()

```
bool TryLock (
    void )
```

tries to enter the lock and returns immediately when not possible

15.31.3.4 Unlock()

```
void Unlock (
    void )
```

leaves the lock

15.31.4 Member Data Documentation

15.31.4.1 m_DebugCount

```
long m_DebugCount [mutable], [protected]
```

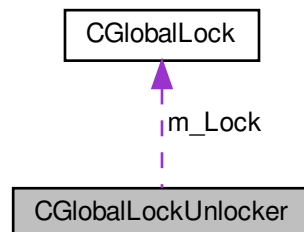
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.32 CGlobalLockUnlocker Class Reference

Unlocks the global lock object on destruction.

Collaboration diagram for CGlobalLockUnlocker:



Public Member Functions

- [CGlobalLockerUnlocker](#) ([CGlobalLocker](#) &lock)
- [~CGlobalLockerUnlocker](#) ()
- void [UnlockEarly](#) (void)

This function allows to unlock the object early before the object is destroyed.

Protected Attributes

- [CGlobalLocker](#) & [m_Lock](#)
- bool [m_enabled](#)

15.32.1 Detailed Description

Unlocks the global lock object on destruction.

This is for automatic UNLOCKING only. We can't do automatic locking here since there is no returnvalue for constructors

15.32.2 Constructor & Destructor Documentation

15.32.2.1 CGlobalLockerUnlocker()

```
CGlobalLockerUnlocker (
    CGlobalLocker & lock ) [inline]
```

15.32.2.2 ~CGlobalLockerUnlocker()

```
~CGlobalLockerUnlocker ( ) [inline]
```

15.32.3 Member Function Documentation

15.32.3.1 UnlockEarly()

```
void UnlockEarly (
    void ) [inline]
```

This function allows to unlock the object early before the object is destroyed.

15.32.4 Member Data Documentation

15.32.4.1 m_enabled

```
bool m_enabled [protected]
```

15.32.4.2 m_Lock

```
CGlobalLock& m_Lock [protected]
```

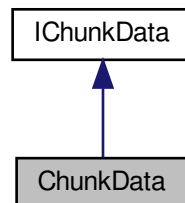
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

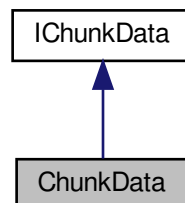
15.33 ChunkData Class Reference

The chunk data which contains additional information about an image.

Inheritance diagram for ChunkData:



Collaboration diagram for ChunkData:



Public Member Functions

- [ChunkData](#) ()
- [ChunkData](#) (const [ChunkData](#) &src)
- virtual [~ChunkData](#) (void)
- void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)
- [float64_t](#) [GetBlackLevel](#) () const
Description: Returns the black level used to capture the image.
- [int64_t](#) [GetFrameID](#) () const
Description: Returns the image frame ID.
- [float64_t](#) [GetExposureTime](#) () const
Description: Returns the exposure time used to capture the image.
- [int64_t](#) [GetCompressionMode](#) () const
Description: Returns the compression mode of the last image payload.
- [float64_t](#) [GetCompressionRatio](#) () const
Description: Returns the compression ratio of the last image payload.
- [int64_t](#) [GetTimestamp](#) () const
Description: Returns the Timestamp of the image.
- [int64_t](#) [GetExposureEndLineStatusAll](#) () const
Description: Returns the status of all the I/O lines at the end of exposure event.
- [int64_t](#) [GetWidth](#) () const
Description: Returns the width of the image included in the payload.
- [int64_t](#) [GetImage](#) () const
Description: Returns the image payload.
- [int64_t](#) [GetHeight](#) () const
Description: Returns the height of the image included in the payload.
- [float64_t](#) [GetGain](#) () const
Description: Returns the gain used to capture the image.
- [int64_t](#) [GetSequencerSetActive](#) () const
Description: Returns the index of the active set of the running sequencer included in the payload.
- [int64_t](#) [GetCRC](#) () const
Description: Returns the CRC of the image payload.
- [int64_t](#) [GetOffsetX](#) () const
Description: Returns the Offset X of the image included in the payload.
- [int64_t](#) [GetOffsetY](#) () const
Description: Returns the Offset Y of the image included in the payload.
- [int64_t](#) [GetSerialDataLength](#) () const
Description: Returns the length of the received serial data that was included in the payload.
- [int64_t](#) [GetPartSelector](#) () const
Description: Selects the part to access in chunk data in a multipart transmission.
- [int64_t](#) [GetPixelDynamicRangeMin](#) () const
Description: Returns the minimum value of dynamic range of the image included in the payload.
- [int64_t](#) [GetPixelDynamicRangeMax](#) () const
Description: Returns the maximum value of dynamic range of the image included in the payload.
- [int64_t](#) [GetTimestampLatchValue](#) () const
Description: Returns the last Timestamp latched with the TimestampLatch command.
- [int64_t](#) [GetLineStatusAll](#) () const
Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.
- [int64_t](#) [GetCounterValue](#) () const
Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.
- [float64_t](#) [GetTimerValue](#) () const

- Description: Returns the value of the selected Timer at the time of the FrameStart internal event.*

 - `int64_t GetScanLineSelector () const`

Description: Index for vector representation of one chunk value per line in an image.
- `int64_t GetEncoderValue () const`

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.
- `int64_t GetLinePitch () const`

Description: Returns the LinePitch of the image included in the payload.
- `int64_t GetTransferBlockID () const`

Description: Returns the unique identifier of the transfer block used to transport the payload.
- `int64_t GetTransferQueueCurrentBlockCount () const`

Description: Returns the current number of blocks in the transfer queue.
- `int64_t GetStreamChannelID () const`

Description: Returns identifier of the stream channel used to carry the block.
- `float64_t GetScan3dCoordinateScale () const`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dCoordinateOffset () const`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dInvalidDataValue () const`

Description: Returns the Invalid Data Value used for the image included in the payload.
- `float64_t GetScan3dAxisMin () const`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dAxisMax () const`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.
- `float64_t GetScan3dTransformValue () const`

Description: Returns the transform value.
- `float64_t GetScan3dCoordinateReferenceValue () const`

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.
- `int64_t GetInferenceFrameId () const`

Description: Returns the frame ID associated with the most recent inference result.
- `int64_t GetInferenceResult () const`

Description: Returns the chunk data inference result.
- `float64_t GetInferenceConfidence () const`

Description: Returns the chunk data inference confidence percentage.
- `InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const`

Description: Returns the chunk inference bounding box result data.

Friends

- class `ChunkDataImpl`

Additional Inherited Members

15.33.1 Detailed Description

The chunk data which contains additional information about an image.

15.33.2 Constructor & Destructor Documentation

15.33.2.1 ChunkData() [1/2]

```
ChunkData ( )
```

15.33.2.2 ChunkData() [2/2]

```
ChunkData (
    const ChunkData & src )
```

15.33.2.3 ~ChunkData()

```
virtual ~ChunkData (
    void ) [virtual]
```

15.33.3 Member Function Documentation

15.33.3.1 GetBlackLevel()

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

15.33.3.2 GetCompressionMode()

```
int64_t GetCompressionMode ( ) const [virtual]
```

Description: Returns the compression mode of the last image payload.

Visibility:

Implements [IChunkData](#).

15.33.3.3 GetCompressionRatio()

```
float64_t GetCompressionRatio ( ) const [virtual]
```

Description: Returns the compression ratio of the last image payload.

Visibility:

Implements [IChunkData](#).

15.33.3.4 GetCounterValue()

```
int64_t GetCounterValue ( ) const [virtual]
```

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.5 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

15.33.3.6 GetEncoderValue()

```
int64_t GetEncoderValue ( ) const [virtual]
```

Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.7 GetExposureEndLineStatusAll()

```
int64_t GetExposureEndLineStatusAll ( ) const [virtual]
```

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

Implements [IChunkData](#).

15.33.3.8 GetExposureTime()

```
float64_t GetExposureTime ( ) const [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

15.33.3.9 GetFrameID()

```
int64_t GetFrameID ( ) const [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

15.33.3.10 GetGain()

```
float64_t GetGain ( ) const [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

15.33.3.11 GetHeight()

```
int64_t GetHeight ( ) const [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.12 GetImage()

```
int64_t GetImage ( ) const [virtual]
```

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

15.33.3.13 GetInferenceBoundingBoxResult()

```
InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [virtual]
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.14 GetInferenceConfidence()

```
float64\_t GetInferenceConfidence ( ) const [virtual]
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.15 GetInferenceFrameId()

```
int64_t GetInferenceFrameId ( ) const [virtual]
```

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.16 GetInferenceResult()

```
int64_t GetInferenceResult ( ) const [virtual]
```

Description: Returns the chunk data inference result.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.17 GetLinePitch()

```
int64_t GetLinePitch ( ) const [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.18 GetLineStatusAll()

```
int64_t GetLineStatusAll ( ) const [virtual]
```

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.19 GetOffsetX()

```
int64_t GetOffsetX ( ) const [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.20 GetOffsetY()

```
int64_t GetOffsetY ( ) const [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.21 GetPartSelector()

```
int64_t GetPartSelector ( ) const [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.22 GetPixelDynamicRangeMax()

```
int64_t GetPixelDynamicRangeMax ( ) const [virtual]
```

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.23 GetPixelDynamicRangeMin()

```
int64_t GetPixelDynamicRangeMin ( ) const [virtual]
```

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.24 GetScan3dAxisMax()

```
float64_t GetScan3dAxisMax ( ) const [virtual]
```

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.25 GetScan3dAxisMin()

```
float64_t GetScan3dAxisMin ( ) const [virtual]
```

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.26 GetScan3dCoordinateOffset()

```
float64_t GetScan3dCoordinateOffset ( ) const [virtual]
```

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.27 GetScan3dCoordinateReferenceValue()

```
float64_t GetScan3dCoordinateReferenceValue ( ) const [virtual]
```

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.28 GetScan3dCoordinateScale()

```
float64_t GetScan3dCoordinateScale ( ) const [virtual]
```

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.29 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.30 GetScan3dTransformValue()

```
float64_t GetScan3dTransformValue ( ) const [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.31 GetScanLineSelector()

```
int64_t GetScanLineSelector ( ) const [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.32 GetSequencerSetActive()

```
int64_t GetSequencerSetActive ( ) const [virtual]
```

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.33 GetSerialDataLength()

```
int64_t GetSerialDataLength ( ) const [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.34 GetStreamChannelID()

```
int64_t GetStreamChannelID ( ) const [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.35 GetTimerValue()

```
float64_t GetTimerValue ( ) const [virtual]
```

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.36 GetTimestamp()

```
int64_t GetTimestamp ( ) const [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

15.33.3.37 GetTimestampLatchValue()

```
int64_t GetTimestampLatchValue ( ) const [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.38 GetTransferBlockID()

```
int64_t GetTransferBlockID ( ) const [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.39 GetTransferQueueCurrentBlockCount()

```
int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

15.33.3.40 GetWidth()

```
int64_t GetWidth ( ) const [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

15.33.3.41 SetChunks()

```
void SetChunks (
    GenApi::INodeMap & pNodeMap ) [virtual]
```

Implements [IChunkData](#).

15.33.4 Friends And Related Function Documentation

15.33.4.1 ChunkDataImpl

```
friend class ChunkDataImpl [friend]
```

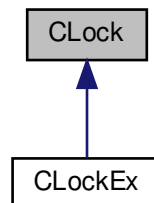
The documentation for this class was generated from the following file:

- [include/ChunkData.h](#)

15.34 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



Public Member Functions

- [CLock](#) ()
Constructor.
- [~CLock](#) ()
Destructor.
- bool [TryLock](#) ()
tries to enter the critical section; returns true if successful
- void [Lock](#) ()
enters the critical section (may block)
- void [Unlock](#) ()
leaves the critical section

15.34.1 Detailed Description

A lock class.

15.34.2 Constructor & Destructor Documentation

15.34.2.1 CLock()

[CLock](#) ()

Constructor.

15.34.2.2 ~CLOCK()

`~CLOCK ( )`

Destructor.

15.34.3 Member Function Documentation

15.34.3.1 Lock()

`void Lock ( )`

enters the critical section (may block)

15.34.3.2 TryLock()

`bool TryLock ( )`

tries to enter the critical section; returns true if successful

15.34.3.3 Unlock()

`void Unlock ( )`

leaves the critical section

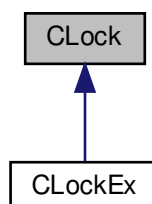
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.35 CLock Class Reference

A lock class.

Inheritance diagram for CLock:



Public Member Functions

- [CLock](#) ()
Constructor.
- [CLock](#) (void *pLock)
Constructor.
- [~CLock](#) ()
Destructor.
- bool [TryLock](#) ()
tries to enter the critical section; returns true if successful
- void [Lock](#) ()
enters the critical section (may block)
- void [Unlock](#) ()
leaves the critical section

Protected Attributes

- void * [m_lock](#)
- bool [m_bOwnLock](#)

Friends

- class [NodeMap](#)

15.35.1 Detailed Description

A lock class.

15.35.2 Constructor & Destructor Documentation

15.35.2.1 CLock() [1/2]

[CLock](#) ()

Constructor.

15.35.2.2 CLock() [2/2]

[CLock](#) (
 void * pLock)

Constructor.

15.35.2.3 ~CLOCK()

`~CLOCK ( )`

Destructor.

15.35.3 Member Function Documentation

15.35.3.1 Lock()

`void Lock ( )`

enters the critical section (may block)

15.35.3.2 TryLock()

`bool TryLock ( )`

tries to enter the critical section; returns true if successful

15.35.3.3 Unlock()

`void Unlock ( )`

leaves the critical section

15.35.4 Friends And Related Function Documentation

15.35.4.1 NodeMap

`friend class NodeMap [friend]`

15.35.5 Member Data Documentation

15.35.5.1 m_bOwnLock

```
bool m_bOwnLock [protected]
```

15.35.5.2 m_lock

```
void* m_lock [protected]
```

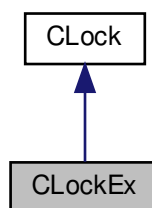
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

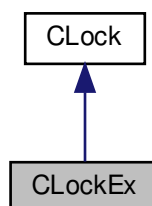
15.36 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Additional Inherited Members

15.36.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

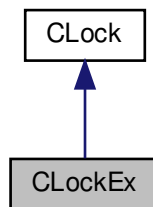
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

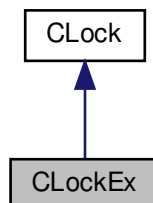
15.37 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Protected Attributes

- void * [m_lockEx](#)

Additional Inherited Members

15.37.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

15.37.2 Member Data Documentation

15.37.2.1 m_lockEx

```
void* m_lockEx [protected]
```

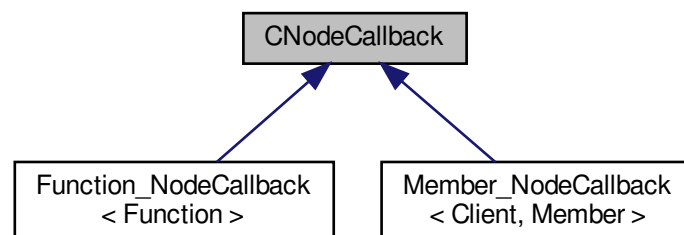
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

15.38 CNodeCallback Class Reference

callback body instance for INode pointers

Inheritance diagram for CNodeCallback:



Public Member Functions

- [CNodeCallback](#) ([INode](#) *pNode, [ECallbackType](#) CallbackType)
- virtual [~CNodeCallback](#) ()
virtual destructor
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const =0
fires the callback if th type is right
- virtual void [Destroy](#) ()=0
destroys the object
- [INode](#) * [GetNode](#) ()
returns the node the callback is registered to
- [ECallbackType](#) [GetCallbackType](#) ()

Protected Attributes

- [INode](#) * [m_pNode](#)
the node were the callback is installed
- [ECallbackType](#) [m_CallbackType](#)
the type of the callback

15.38.1 Detailed Description

callback body instance for INode pointers

15.38.2 Constructor & Destructor Documentation

15.38.2.1 CNodeCallback()

```
CNodeCallback (
    INode * pNode,
    ECallbackType CallbackType ) [inline]
```

15.38.2.2 ~CNodeCallback()

```
virtual ~CNodeCallback ( ) [inline], [virtual]
```

virtual destructor

15.38.3 Member Function Documentation

15.38.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

15.38.3.2 GetCallbackType()

```
ECallbackType GetCallbackType ( ) [inline]
```

15.38.3.3 GetNode()

```
INode* GetNode ( ) [inline]
```

returns the node the callback is registered to

15.38.3.4 operator>()()

```
virtual void operator() (
    ECallbackType CallbackType ) const [pure virtual]
```

fires the callback if th type is right

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

15.38.4 Member Data Documentation

15.38.4.1 m_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

15.38.4.2 m_pNode

```
INode* m_pNode [protected]
```

the node were the callback is installed

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

- include/SpinGenApi/NodeCallback.h

15.39 CNodeMapFactory Class Reference

The node map factory is used for creating node maps from camera description files.

Classes

- struct [NodeStatistics_t](#)

Public Member Functions

- [CNodeMapFactory](#) ()
Creates an empty node map factory for assigning a non-empty node map factory later.
- virtual [~CNodeMapFactory](#) ()
Destroys the node map factory data if all references to the data have been released.
- [CNodeMapFactory](#) (const [CNodeMapFactory](#) &)
Creates another reference to the node map factory data.
- [CNodeMapFactory](#) & operator= (const [CNodeMapFactory](#) &)
Creates another reference to the assigned node map factory data.
- [CNodeMapFactory](#) (EContentType_t FileType, const [GenICam::gcstring](#) &FileName, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false)
Creates the node map factory and simply stores the full path to the provided camera description file data.
- [CNodeMapFactory](#) (EContentType_t ContentType, const void *pData, size_t DataSize, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false)
Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.
- [CNodeMapFactory](#) (const [GenICam::gcstring](#) &XmlData, ECacheUsage_t CacheUsage=CacheUsage_Automatic, bool SuppressStringsOnLoad=false)
Creates the node map factory and copies the provided camera description file string.
- bool [IsEmpty](#) () const
Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.
- void [AddInjectionData](#) ([CNodeMapFactory](#) &injectionData)
Adds a node map factory representing a camera description file to inject.
- void [LoadAndInject](#) ()
Advanced: Loads, Parses, and Injects the camera description files recursively.
- bool [IsLoaded](#) () const
Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.
- [CNodeMapFactory](#) [ExtractSubtree](#) (const [GenICam::gcstring](#) &SubTreeRootNodeName, bool doRename←ToRoot=false)
The name of the node that represents the root of the subtree that shall be extracted.

- void [Preprocess](#) ()
Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the CNodeMap (not part of the public interface).
- bool [IsPreprocessed](#) () const
Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.
- void [ReleaseCameraDescriptionFileData](#) ()
Advanced: Releases any in constructors provided camera description file data buffers or files.
- bool [IsCameraDescriptionFileDataReleased](#) () const
Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.
- [INodeMap](#) * [CreateNodeMap](#) (const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)
Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- [INodeMap](#) * [CreateNodeMap](#) (CLOCK &UserProvidedLock, const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)
Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- void [GetSupportedSchemaVersions](#) ([GenlCam::gcstring_vector](#) &SchemaVersions) const
- [GenlCam::gcstring](#) [ToString](#) () const
Outputs the pre-processed node map in string form (for debug purpose)
- [GenlCam::gcstring](#) [ToXml](#) () const
Outputs the pre-processed node map in XML form (mainly for debug purpose)
- void [GetNodeStatistics](#) ([NodeStatistics_t](#) &NodeStatistics)
- const [GenlCam::gcstring](#) [ApplyStyleSheet](#) (const [GenlCam::gcstring](#) &StyleSheetFileName)
Applies a style sheet to the pre-processed node map.

Static Public Member Functions

- static [INodeMap](#) * [CreateEmptyNodeMap](#) ()
Creates an empty node map usable as placeholder, e.g.
- static bool [ClearCache](#) ()
Deletes all preprocessed camera description files from the cache.
- static [CNodeDataMap](#) * [CreateNodeDataFromNodeMap](#) ([INodeMap](#) *pNodeMap)

15.39.1 Detailed Description

The node map factory is used for creating node maps from camera description files.

Examples

```
// Simple node map creation from buffer, downloaded from a device for instance.
CNodeMapFactory cameraNodeMapFactory( ContentType_ZippedXml, buffer, bufferSize);
// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).
// Node map creation with injecting additional xml fragments.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(ContentType_Xml, filename2));
// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).
// Node map creation and additional extraction of a category subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
// Extract a subtree for later chunk parsing.
CNodeMapFactory chunkDataNodeMapFactory = cameraParameters.ExtractSubtree("ChunkData");
// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraParameters.CreateNodeMap();
```

```
// The next step is attaching the device port (not shown).
// Node map creation with injecting additional xml fragments and additional extraction of a category
subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(ContentType_Xml, filename2));
CNodeMapFactory chunkDataNodeMapFactory = cameraNodeMapFactory.ExtractSubtree("ChunkData");
// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).
// A node map factory can create multiple node maps from the provided camera description file(s).
for(int i = 0; i < 20; ++i)
{
    INodeMap* pNodeMapChunks = chunkDataNodeMapFactory.CreateNodeMap();
    //...
}
```

Attention

The is [CNodeMapFactory](#) not thread-safe.

You need to take care when camera description file data can be actually be freed, see method documentation of the node map factory for more detail.

15.39.2 Constructor & Destructor Documentation

15.39.2.1 CNodeMapFactory() [1/5]

```
CNodeMapFactory ( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

15.39.2.2 ~CNodeMapFactory()

```
virtual ~CNodeMapFactory ( ) [virtual]
```

Destroys the node map factory data if all references to the data have been released.

15.39.2.3 CNodeMapFactory() [2/5]

```
CNodeMapFactory (
    const CNodeMapFactory & )
```

Creates another reference to the node map factory data.

No data is copied.

15.39.2.4 CNodeMapFactory() [3/5]

```
CNodeMapFactory (
    EContentType_t FileType,
    const GenICam::gcstring & FileName,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the full path to the provided camera description file data.

Parameters

in	<i>FileType</i>	Defines how the camera description file is stored, e.g. as zipped XML text.
in	<i>FileName</i>	The full path of the camera description file to process.
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if *FileName* is empty. Throws if environment variables in *FileName* cannot be resolved.

Attention

The given file must be readable until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

15.39.2.5 CNodeMapFactory() [4/5]

```
CNodeMapFactory (
    EContentType_t ContentType,
    const void * pData,
    size_t DataSize,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.

Parameters

in	<i>ContentType</i>	Defines how the camera description file is stored, e.g. as zipped XML text.
in	<i>pData</i>	The pointer to the camera description file data.
in	<i>DataSize</i>	The size of the camera description file data.
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if *pData* is NULL or *DataSize* is 0.

Attention

The given buffer must not be freed or changed until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

15.39.2.6 CNodeMapFactory() [5/5]

```
CNodeMapFactory (
    const GenICam::gcstring & XmlData,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and copies the provided camera description file string.

Parameters

in	<i>XmlData</i>	The camera description file data as XML text. The provided text is copied. You can use the overloaded constructor accepting a buffer to avoid that. <pre>gcstring cdfData; //... fill cdfData ... CNodeMapFactory factory(ContentType_Xml, cdfData.c_str(), cdfData.size()); // Create the node map. The node map can be destroyed using the IDestroy interface later. INodeMap* pNodeMap = factory.CreateNodeMap(); // The next step is attaching the device port (not shown).</pre>
in	<i>CacheUsage</i>	Defines if and how to use the cache for preprocessed camera description files.
in	<i>SuppressStringsOnLoad</i>	Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint.

Throws an invalid argument exception if XmlData is empty.

15.39.3 Member Function Documentation

15.39.3.1 AddInjectionData()

```
void AddInjectionData (
    CNodeMapFactory & injectionData )
```

Adds a node map factory representing a camera description file to inject.

Parameters

in	<i>injectionData</i>	A node map factory representing a camera description file to inject.

The injected files are injected in the order they are added. InjectionData must not be preprocessed. The `IsPreprocessed()` method can be used to check if preprocessing has been done before. The cache usage of injection data is ignored.

15.39.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
    const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

15.39.3.3 ClearCache()

```
static bool ClearCache ( ) [static]
```

Deletes all preprocessed camera description files from the cache.

15.39.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( ) [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

15.39.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
    INodeMap * pNodeMap ) [static]
```

15.39.3.6 CreateNodeMap() [1/2]

```
INodeMap* CreateNodeMap (
    CLock & UserProvidedLock,
    const GenICam::gcstring & DeviceName = "Device",
    bool DoReleaseCameraDescriptionFileData = true )
```

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if `DoReleaseCameraDescriptionFileData` is true. This method can be called multiple times to create multiple instances of a node map. This method allows to provide an external lock to avoid using too many locks in an application.

Attention

The provided lock must not be destroyed before the created node map.

15.39.3.7 CreateNodeMap() [2/2]

```
INodeMap* CreateNodeMap (
    const GenICam::gcstring & DeviceName = "Device",
    bool DoReleaseCameraDescriptionFileData = true )
```

Creates a node map from the preprocessed memory internal representation of the camera description file(s).

[Preprocess\(\)](#) is automatically called if needed. The preprocess step can be omitted by the factory depending on the cache mode setting when a cache file is available, then the cache file is read and converted directly into a node map. [ReleaseCameraDescriptionFileData\(\)](#) is called if `DoReleaseCameraDescriptionFileData` is true. This method can be called multiple times to create multiple instances of a node map.

15.39.3.8 ExtractSubtree()

```
CNodeMapFactory ExtractSubtree (
    const GenICam::gcstring & SubTreeRootNodeName,
    bool doRenameToRoot = false )
```

The name of the node that represents the root of the subtree that shall be extracted.

Parameters

in	<i>SubTreeRootNodeName</i>	The root of the branch to extract, e.g. "ChunkData".
in	<i>doRenameToRoot</i>	Renames the extracted subtree root node SubTreeRootNodeName to "Root", sets the IsFeature property. Preprocess() is automatically called if needed to create the memory internal representation of the camera description file(s). The preprocessed result can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting.

15.39.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
    NodeStatistics_t & NodeStatistics )
```

15.39.3.10 GetSupportedSchemaVersions()

```
void GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) const
```

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

15.39.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

15.39.3.12 IsEmpty()

```
bool IsEmpty ( ) const
```

Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.

when the node map factory has been created with the default constructor.

15.39.3.13 IsLoaded()

```
bool IsLoaded ( ) const
```

Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.

Returns true if [IsPreprocessed\(\)](#) returns true (Preprocessed Data has been loaded from cache).

15.39.3.14 IsPreprocessed()

```
bool IsPreprocessed ( ) const
```

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

15.39.3.15 LoadAndInject()

```
void LoadAndInject ( )
```

Advanced: Loads, Parses, and Injects the camera description files recursively.

The result is a memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Prevents cache read if called manually.

15.39.3.16 operator=()

```
CNodeMapFactory& operator= (
    const CNodeMapFactory & )
```

Creates another reference to the assigned node map factory data.

Destroys the "overwritten" node map factory data if all references to the data have been released.

15.39.3.17 Preprocess()

```
void Preprocess ( )
```

Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Preprocessed data can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. By calling this method directly direct cache load is suppressed, see [CreateNodeMap\(\)](#) for more information.

15.39.3.18 ReleaseCameraDescriptionFileData()

```
void ReleaseCameraDescriptionFileData ( )
```

Advanced: Releases any in constructors provided camera description file data buffers or files.

This step is usually done automatically. All references to added injection data are dropped in this step to free the data. After this step any in constructors provided buffers can be freed or any in constructors given files can be deleted.

15.39.3.19 ToString()

```
GenICam::gcstring ToString ( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

15.39.3.20 ToXml()

```
GenICam::gcstring ToXml ( ) const
```

Outputs the pre-processed node map in XML form (mainly for debug purpose)

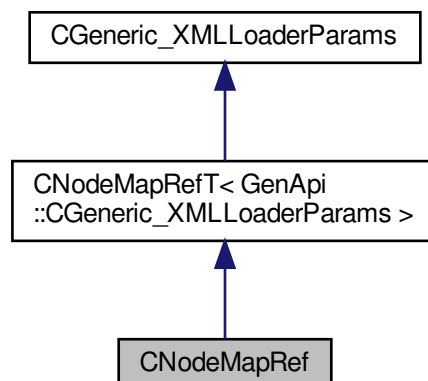
The documentation for this class was generated from the following file:

- [include/SpinGenApi/NodeMapFactory.h](#)

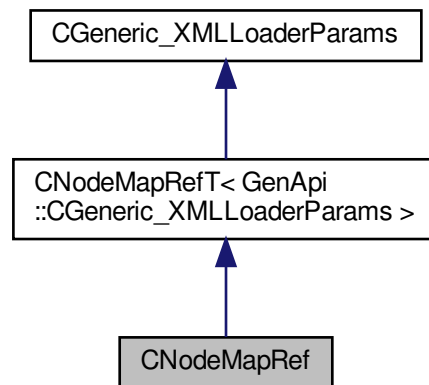
15.40 CNodeMapRef Class Reference

Smartpointer for NodeMaps with create function.

Inheritance diagram for CNodeMapRef:



Collaboration diagram for CNodeMapRef:



Public Member Functions

- **CNodeMapRef** (const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (INodeMap *pNodeMap, const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (const **CNodeMapRef** &Them)
Copy constructor.
- **CNodeMapRef** & **operator=** (const **CNodeMapRef** &Them)
Assignment.
- **CNodeMapRef** & **operator=** (INodeMap *pNodeMap)
Assignment of an INodeMap.*

Additional Inherited Members

15.40.1 Detailed Description

Smartpointer for NodeMaps with create function.

Note

This class is a simple typedef definition. The class syntax is only used, because Doxygen has to generate a useful documentation.

15.40.2 Constructor & Destructor Documentation

15.40.2.1 CNodeMapRef() [1/3]

```
CNodeMapRef (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.40.2.2 CNodeMapRef() [2/3]

```
CNodeMapRef (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.40.2.3 CNodeMapRef() [3/3]

```
CNodeMapRef (
    const CNodeMapRef & Them ) [inline]
```

Copy constructor.

15.40.3 Member Function Documentation

15.40.3.1 operator=() [1/2]

```
CNodeMapRef& operator= (
    const CNodeMapRef & Them ) [inline]
```

Assignment.

15.40.3.2 operator=() [2/2]

```
CNodeMapRef& operator= (
    INodeMap * pNodeMap ) [inline]
```

Assignment of an INodeMap*.

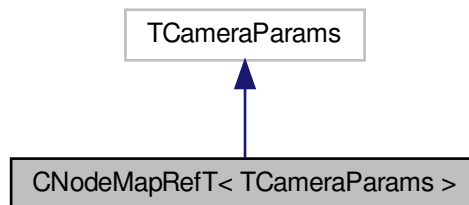
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

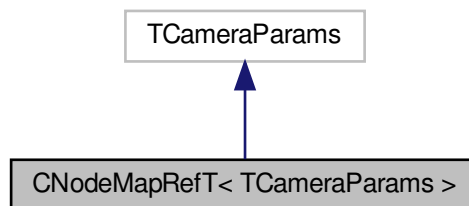
15.41 CNodeMapRefT< TCameraParams > Class Template Reference

Smartpointer template for NodeMaps with create function.

Inheritance diagram for CNodeMapRefT< TCameraParams >:



Collaboration diagram for CNodeMapRefT< TCameraParams >:



Public Member Functions

- `CNodeMapRefT` (const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (`INodeMap` *pNodeMap, const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (const `CNodeMapRefT` &Them)
Copy constructor.
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)
Assignment.
- `CNodeMapRefT` & `operator=` (`INodeMap` *pNodeMap)
Assignment of an INodeMap.*
- virtual `~CNodeMapRefT` ()
Destructor.
- void `_Destroy` ()

- Destroys the node map.*
- void `_LoadXMLFromFile` (const `GenICam::gcstring` &FileName)
 - Creates the object from a XML file with given file name.*
- void `_LoadXMLFromZIPFile` (const `GenICam::gcstring` &ZipFileName)
 - Creates the object from a ZIP'd XML file with given file name.*
- void `_LoadXMLFromZIPData` (const void *zipData, size_t zipSize)
 - Creates the object from a ZIP'd XML file given in a string.*
- void `_LoadXMLFromFileInject` (const `GenICam::gcstring` &TargetFileName, const `GenICam::gcstring` &InjectFileName)
 - Creates the object from a XML target and an inject file with given file name.*
- void `_LoadXMLFromString` (const `GenICam::gcstring` &XMLData)
 - Creates the object from XML data given in a string.*
- void `_LoadXMLFromStringInject` (const `GenICam::gcstring` &TargetXMLData, const `GenICam::gcstring` &InjectXMLData)
 - Creates the object from XML data given in a string with injection.*
- virtual void `_GetSupportedSchemaVersions` (`GenICam::gcstring_vector` &SchemaVersions)
 - Gets a list of supported schema versions.*
- virtual `GenICam::gcstring` `_GetDeviceName` ()
 - Get device name.*
- virtual void `_Poll` (int64_t ElapsedTime)
 - Fires nodes which have a polling time.*
- virtual void `_GetNodes` (`NodeList_t` &Nodes)
 - Retrieves all nodes in the node map.*
- virtual `Inode *` `_GetNode` (const `GenICam::gcstring` &key)
 - Retrieves the node from the central map by name.*
- virtual void `_InvalidateNodes` ()
 - Invalidates all nodes.*
- virtual bool `_Connect` (`IPort *`pPort, const `GenICam::gcstring` &PortName)
 - Connects a port to a port node with given name.*
- virtual bool `_Connect` (`IPort *`pPort)
 - Connects a port to the standard port "Device".*

Static Public Member Functions

- static bool `_ClearXMLCache` ()
 - Clears the cache of the camera description files.*

Public Attributes

- `InodeMap *` `_Ptr`
 - Pointer to the `NodeMap`.*

15.41.1 Detailed Description

```
template<class TCameraParams>
class Spinnaker::GenApi::CNodeMapRefT< TCameraParams >
```

Smartpointer template for NodeMaps with create function.

Parameters

<i>TCameraParams</i>	The camera specific parameter class (auto generated from camera xml file)

15.41.2 Constructor & Destructor Documentation

15.41.2.1 CNodeMapRefT() [1/3]

```
CNodeMapRefT (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.41.2.2 CNodeMapRefT() [2/3]

```
CNodeMapRefT (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

15.41.2.3 CNodeMapRefT() [3/3]

```
CNodeMapRefT (
    const CNodeMapRefT< TCameraParams > & Them )
```

Copy constructor.

15.41.2.4 ~CNodeMapRefT()

```
~CNodeMapRefT [inline], [virtual]
```

Destructor.

15.41.3 Member Function Documentation

15.41.3.1 `_ClearXMLCache()`

```
static bool _ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

15.41.3.2 `_Connect()` [1/2]

```
virtual bool _Connect (
    IPort * pPort ) [virtual]
```

Connects a port to the standard port "Device".

15.41.3.3 `_Connect()` [2/2]

```
virtual bool _Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) [virtual]
```

Connects a port to a port node with given name.

15.41.3.4 `_Destroy()`

```
void _Destroy [inline]
```

Destroys the node map.

15.41.3.5 `_GetDeviceName()`

```
virtual GenICam::gcstring _GetDeviceName ( ) [virtual]
```

Get device name.

15.41.3.6 `_GetNode()`

```
virtual INode* _GetNode (
    const GenICam::gcstring & key ) [virtual]
```

Retrieves the node from the central map by name.

15.41.3.7 _GetNodes()

```
virtual void _GetNodes (
    NodeList_t & Nodes ) [virtual]
```

Retrieves all nodes in the node map.

15.41.3.8 _GetSupportedSchemaVersions()

```
virtual void _GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

15.41.3.9 _InvalidateNodes()

```
virtual void _InvalidateNodes ( ) [virtual]
```

Invalidates all nodes.

15.41.3.10 _LoadXMLFromFile()

```
void _LoadXMLFromFile (
    const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

15.41.3.11 _LoadXMLFromFileInject()

```
void _LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

15.41.3.12 _LoadXMLFromString()

```
void _LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

15.41.3.13 _LoadXMLFromStringInject()

```
void _LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

15.41.3.14 _LoadXMLFromZIPData()

```
void _LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

15.41.3.15 _LoadXMLFromZIPFile()

```
void _LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

15.41.3.16 _Poll()

```
virtual void _Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

15.41.3.17 operator=() [1/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    const CNodeMapRefT< TCameraParams > & Them )
```

Assignment.

15.41.3.18 operator=() [2/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    INodeMap * pNodeMap )
```

Assignment of an INodeMap*.

15.41.4 Member Data Documentation**15.41.4.1 _Ptr**

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

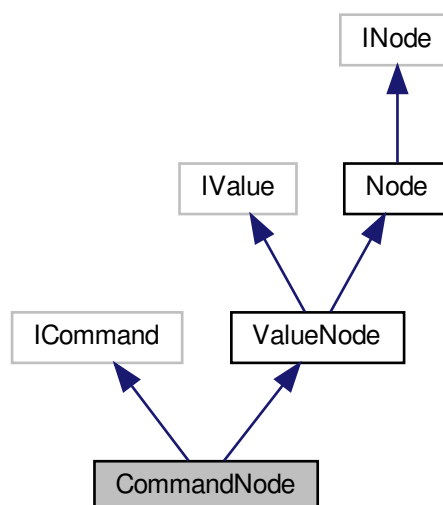
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

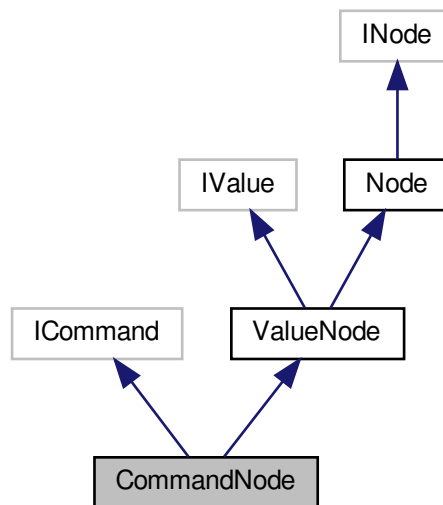
15.42 CommandNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CommandNode:



Collaboration diagram for CommandNode:



Public Member Functions

- [CommandNode](#) ()
- [CommandNode](#) (std::shared_ptr< Node::NodeImpl > pCommand)
- virtual [~CommandNode](#) ()
- virtual void [Execute](#) (bool [Verify](#)=true)
Execute the command.
- virtual void [operator\(\)](#) ()
Execute the command.
- virtual bool [IsDone](#) (bool [Verify](#)=true)
Query whether the command is executed.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.42.1 Detailed Description

[Interface](#) for string properties.

15.42.2 Constructor & Destructor Documentation

15.42.2.1 CommandNode() [1/2]

```
CommandNode ( )
```

15.42.2.2 CommandNode() [2/2]

```
CommandNode (
    std::shared_ptr< Node::NodeImpl > pCommand )
```

15.42.2.3 ~CommandNode()

```
virtual ~CommandNode ( ) [virtual]
```

15.42.3 Member Function Documentation

15.42.3.1 Execute()

```
virtual void Execute (
    bool Verify = true ) [virtual]
```

Execute the command.

Parameters

<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.42.3.2 IsDone()

```
virtual bool IsDone (
    bool Verify = true ) [virtual]
```

Query whether the command is executed.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked

Returns

True if the Execute command has finished; false otherwise

15.42.3.3 operator>()()

```
virtual void operator() ( ) [virtual]
```

Execute the command.

15.42.3.4 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

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

- include/SpinGenApi/[CommandNode.h](#)

15.43 CompressedImageInfo Struct Reference

Public Member Functions

- [CompressedImageInfo](#) (const string &[fileName](#), size_t imageSize, size_t width, size_t height, size_t xOffset, size_t yOffset, [PixelFormatEnums](#) pixelFormat)

Public Attributes

- string [fileName](#)
- size_t [compressedImageSize](#)
- size_t [imageWidth](#)
- size_t [imageHeight](#)
- size_t [imageXOffset](#)
- size_t [imageYOffset](#)
- [PixelFormatEnums](#) [imagePixelFormat](#)

15.43.1 Constructor & Destructor Documentation

15.43.1.1 CompressedImageInfo()

```
CompressedImageInfo (
    const string & fileName,
    size_t imageSize,
    size_t width,
    size_t height,
    size_t xOffset,
    size_t yOffset,
    PixelFormatEnums pixelFormat ) [inline]
```

15.43.2 Member Data Documentation

15.43.2.1 compressedImageSize

size_t compressedImageSize

15.43.2.2 fileName

string fileName

15.43.2.3 imageHeight

size_t imageHeight

15.43.2.4 imagePixelFormat

PixelFormatEnums imagePixelFormat

15.43.2.5 imageWidth

size_t imageWidth

15.43.2.6 imageXOffset

```
size_t imageXOffset
```

15.43.2.7 imageYOffset

```
size_t imageYOffset
```

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

- src/Compression/[Compression.cpp](#)

15.44 Counter Class Reference

Definition of a simple [Counter](#) class.

Public Member Functions

- [Counter](#) ()
- unsigned int [GetValue](#) () const
- unsigned int [operator++](#) ()
- unsigned int [operator++](#) (int)
- unsigned int [operator--](#) (int)
- unsigned int [operator--](#) ()
- [operator unsigned int](#) ()
- bool [IsZero](#) ()

15.44.1 Detailed Description

Definition of a simple [Counter](#) class.

15.44.2 Constructor & Destructor Documentation

15.44.2.1 Counter()

```
Counter ( ) [inline]
```

15.44.3 Member Function Documentation

15.44.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

15.44.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

15.44.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

15.44.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

15.44.3.5 operator++() [2/2]

```
unsigned int operator++ (
    int ) [inline]
```

15.44.3.6 operator--() [1/2]

```
unsigned int operator-- ( ) [inline]
```

15.44.3.7 operator--() [2/2]

```
unsigned int operator-- (
    int ) [inline]
```

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

- include/SpinGenApi/[Counter.h](#)

15.45 CPointer< T, B > Class Template Reference

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

Public Member Functions

- [CPointer](#) (void)
Default constructor.
- [CPointer](#) (B *pB)
Constructor from INode pointer type.
- virtual [~CPointer](#) (void)
- void [operator=](#) (B *pB)
Assign INode Pointer.
- [operator T*](#) (void) const
Dereferencing.
- T & [operator*](#) (void) const
Dereferencing.
- T & [operator\(\)](#) (void) const
Dereferencing.
- T * [operator->](#) (void) const
Dereferencing.
- bool [IsValid](#) () const throw ()
true if the pointer is valid
- [operator bool](#) (void) const throw ()
true if the pointer is valid
- bool [operator==](#) (T *pT) const
pointer equal
- bool [operator==](#) (const [CPointer](#)< T, B > &rT) const
pointer equal
- bool [operator==](#) (int nMustBeNull) const
pointer equal
- bool [operator!=](#) (const [CPointer](#)< T, B > &rT) const
pointer unequal
- bool [operator!=](#) (T *pT) const
pointer unequal
- bool [operator!=](#) (const long int nMustBeNull) const
pointer unequal
- bool [operator!=](#) (const int nMustBeNull) const
pointer unequal
- bool [operator!=](#) (const std::nullptr_t nullptr) const
pointer unequal

Protected Attributes

- T * [m_pT](#)
Underlying raw pointer.

15.45.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

15.45.2 Constructor & Destructor Documentation

15.45.2.1 CPointer() [1/2]

```
CPointer (
    void ) [inline]
```

Default constructor.

15.45.2.2 CPointer() [2/2]

```
CPointer (
    B * pB ) [inline]
```

Constructor from INode pointer type.

15.45.2.3 ~CPointer()

```
virtual ~CPointer (
    void ) [inline], [virtual]
```

15.45.3 Member Function Documentation

15.45.3.1 IsValid()

```
bool IsValid ( ) const throw ( ) [inline]
```

true if the pointer is valid

15.45.3.2 operator bool()

```
operator bool (
    void ) const throw ( )    [inline]
```

true if the pointer is valid

15.45.3.3 operator T*()

```
operator T* (
    void ) const    [inline]
```

Dereferencing.

15.45.3.4 operator"!=() [1/5]

```
bool operator!= (
    const CPointer< T, B > & rT ) const    [inline]
```

pointer unequal

15.45.3.5 operator"!=() [2/5]

```
bool operator!= (
    const int nMustBeNull ) const    [inline]
```

pointer unequal

15.45.3.6 operator"!=() [3/5]

```
bool operator!= (
    const long int nMustBeNull ) const    [inline]
```

pointer unequal

15.45.3.7 operator"!=() [4/5]

```
bool operator!= (
    const std::nullptr_t nullptr ) const    [inline]
```

pointer unequal

15.45.3.8 operator"!=()" [5/5]

```
bool operator!= (
    T * pT ) const [inline]
```

pointer unequal

15.45.3.9 operator>()()

```
T& operator() (
    void ) const [inline]
```

Dereferencing.

15.45.3.10 operator*()

```
T& operator* (
    void ) const [inline]
```

Dereferencing.

15.45.3.11 operator->()

```
T* operator-> (
    void ) const [inline]
```

Dereferencing.

15.45.3.12 operator=()

```
void operator= (
    B * pB ) [inline]
```

Assign INode Pointer.

15.45.3.13 operator==([1/3]

```
bool operator== (
    const CPointer< T, B > & rT ) const [inline]
```

pointer equal

15.45.3.14 operator==() [2/3]

```
bool operator== (
    int nMustBeNull ) const [inline]
```

pointer equal

15.45.3.15 operator==() [3/3]

```
bool operator== (
    T * pT ) const [inline]
```

pointer equal

15.45.4 Member Data Documentation**15.45.4.1 m_pT**

```
T* m_pT [protected]
```

Underlying raw pointer.

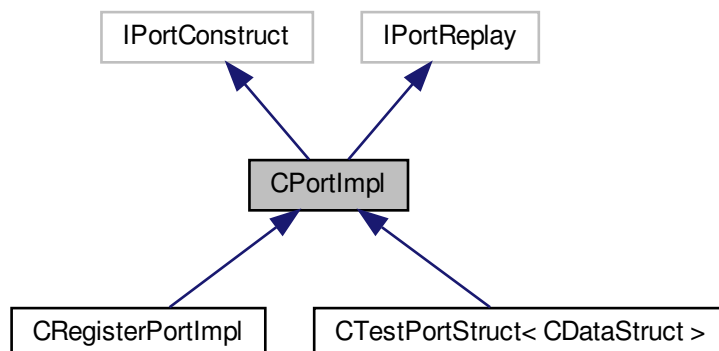
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Pointer.h](#)

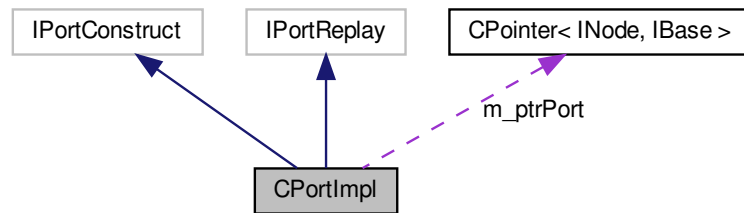
15.46 CPortImpl Class Reference

Standard implementation for a port.

Inheritance diagram for CPortImpl:



Collaboration diagram for CPortImpl:



Public Member Functions

- [CPortImpl](#) ()
Constructor.
- virtual [~CPortImpl](#) ()
Destructor.
- virtual [EAccessMode](#) [GetAccessMode](#) () const =0
Get the access mode of the node.
- virtual void [Read](#) (void *pBuffer, [int64_t](#) [Address](#), [int64_t](#) [Length](#))=0
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, [int64_t](#) [Address](#), [int64_t](#) [Length](#))=0
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) ([IPort](#) *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- void [InvalidateNode](#) ()

Protected Attributes

- [CNodePtr](#) [m_ptrPort](#)
Pointer to the node holding a reference to this implementation.

15.46.1 Detailed Description

Standard implementation for a port.

15.46.2 Constructor & Destructor Documentation

15.46.2.1 CPortImpl()

```
CPortImpl ( ) [inline]
```

Constructor.

15.46.2.2 ~CPortImpl()

```
virtual ~CPortImpl ( ) [inline], [virtual]
```

Destructor.

15.46.3 Member Function Documentation

15.46.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [pure virtual]
```

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.46.3.2 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.46.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```


15.46.3.4 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads a chunk of bytes from the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.46.3.5 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [inline], [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

15.46.3.6 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

15.46.3.7 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes a chunk of bytes to the port.

Implemented in [CRegisterPortImpl](#), and [CTestPortStruct< CDataStruct >](#).

15.46.4 Member Data Documentation

15.46.4.1 m_ptrPort

`CNodePtr m_ptrPort` [protected]

Pointer to the node holding a reference to this implementation.

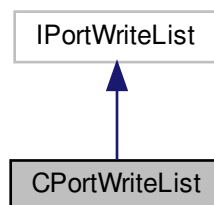
The documentation for this class was generated from the following file:

- `include/SpinGenApi/PortImpl.h`

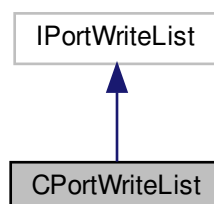
15.47 CPortWriteList Class Reference

Container holding a list of port write commands.

Inheritance diagram for CPortWriteList:



Collaboration diagram for CPortWriteList:



Public Member Functions

- [CPortWriteList](#) ()
Constructor.
- [~CPortWriteList](#) ()
Destructor.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [Replay](#) (IPort *pPort)
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()
Gets the cookie a port implementation may have set for caching a command list.
- void * [GetPortWriteListHandle](#) ()

Protected Attributes

- void * [m_pWriteList](#)

15.47.1 Detailed Description

Container holding a list of port write commands.

15.47.2 Constructor & Destructor Documentation

15.47.2.1 CPortWriteList()

```
CPortWriteList ( )
```

Constructor.

15.47.2.2 ~CPortWriteList()

```
~CPortWriteList ( )
```

Destructor.

15.47.3 Member Function Documentation

15.47.3.1 GetCookie()

```
virtual int64_t GetCookie ( ) [virtual]
```

Gets the cookie a port implementation may have set for caching a command list.

15.47.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ( )
```

15.47.3.3 Replay()

```
virtual void Replay (
    IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

15.47.3.4 SetCookie()

```
virtual void SetCookie (
    const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

15.47.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

15.47.4 Member Data Documentation

15.47.4.1 m_pWriteList

```
void* m_pWriteList [protected]
```

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

- include/SpinGenApi/[PortWriteList.h](#)

15.48 CpuUsageInfo Struct Reference

Public Attributes

- bool [dummy](#)

15.48.1 Member Data Documentation

15.48.1.1 dummy

```
bool dummy
```

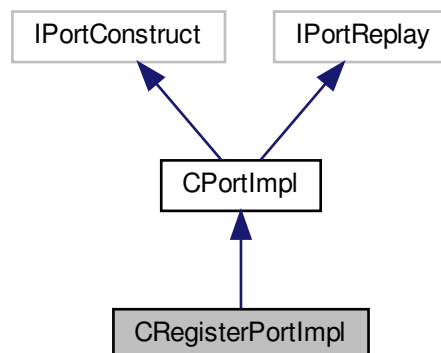
The documentation for this struct was generated from the following file:

- src/GigEVisionPerformance/[CpuUtil.h](#)

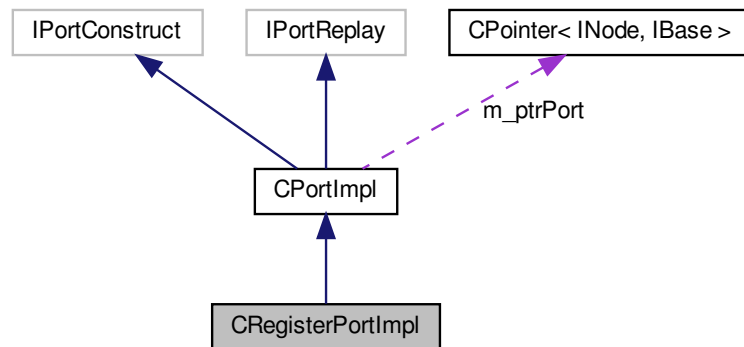
15.49 CRegisterPortImpl Class Reference

Standard implementation for a port using a register based transport layer.

Inheritance diagram for CRegisterPortImpl:



Collaboration diagram for CRegisterPortImpl:



Public Member Functions

- [CRegisterPortImpl](#) (int MaxNumQuadlets=1, bool TransportLayerSwapsEndianness=false)
Constructor.
- virtual [~CRegisterPortImpl](#) ()
Destructor.
- virtual EAccessMode [GetAccessMode](#) () const =0
Get the access mode of the node.
- virtual void [ReadRegister](#) (uint32_t *pRegisters, int64_t [Address](#), int64_t [Length](#))=0
Reads an array of quadlets from the port.
- virtual void [WriteRegister](#) (const uint32_t *pRegisters, int64_t [Address](#), int64_t [Length](#))=0
Writes an array of quadlets to the port.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.

Additional Inherited Members

15.49.1 Detailed Description

Standard implementation for a port using a register based transport layer.

15.49.2 Constructor & Destructor Documentation

15.49.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
    int MaxNumQuadlets = 1,
    bool TransportLayerSwapsEndianness = false ) [inline]
```

Constructor.

15.49.2.2 ~CRegisterPortImpl()

```
virtual ~CRegisterPortImpl ( ) [inline], [virtual]
```

Destructor.

15.49.3 Member Function Documentation

15.49.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [pure virtual]
```

Get the access mode of the node.

Driver closed => NI, Driver open => RW, analyzing a struct, RO

Implements [CPortImpl](#).

15.49.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

15.49.3.3 ReadRegister()

```
virtual void ReadRegister (
    uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads an array of quadlets from the port.

15.49.3.4 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

15.49.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

15.49.3.6 WriteRegister()

```
virtual void WriteRegister (
    const uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes an array of quadlets to the port.

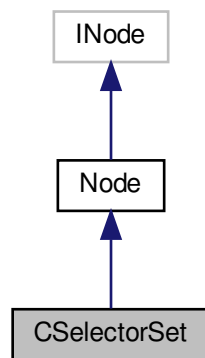
The documentation for this class was generated from the following file:

- [include/SpinGenApi/RegisterPortImpl.h](#)

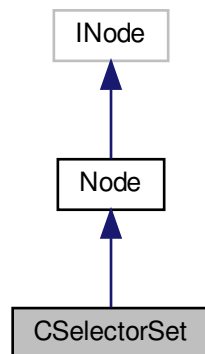
15.50 CSelectorSet Class Reference

The set of selectors selecting a given node.

Inheritance diagram for CSelectorSet:



Collaboration diagram for CSelectorSet:



Public Member Functions

- [CSelectorSet](#) ([IBase](#) *pBase)
Constructor.
- [~CSelectorSet](#) ()
Destructor.
- bool [IsEmpty](#) ()

returns true if no selectors are present

- virtual bool [SetFirst](#) ()
- virtual bool [SetNext](#) (bool Tick=true)
- virtual void [Restore](#) ()
- virtual [GenICam::gcstring ToString](#) ()
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)

Additional Inherited Members

15.50.1 Detailed Description

The set of selectors selecting a given node.

15.50.2 Constructor & Destructor Documentation

15.50.2.1 CSelectorSet()

```
CSelectorSet (
    IBase * pBase )
```

Constructor.

Parameters

<i>pBase</i>	> Feature selected by the selector set

15.50.2.2 ~CSelectorSet()

```
~CSelectorSet ( )
```

Destructor.

15.50.3 Member Function Documentation

15.50.3.1 GetSelectorList()

```
virtual void GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [virtual]
```

15.50.3.2 IsEmpty()

```
bool IsEmpty ( )
```

returns true if no selectors are present

15.50.3.3 Restore()

```
virtual void Restore ( ) [virtual]
```

15.50.3.4 SetFirst()

```
virtual bool SetFirst ( ) [virtual]
```

15.50.3.5 SetNext()

```
virtual bool SetNext (
    bool Tick = true ) [virtual]
```

15.50.3.6 ToString()

```
virtual GenICam::gcstring ToString ( ) [virtual]
```

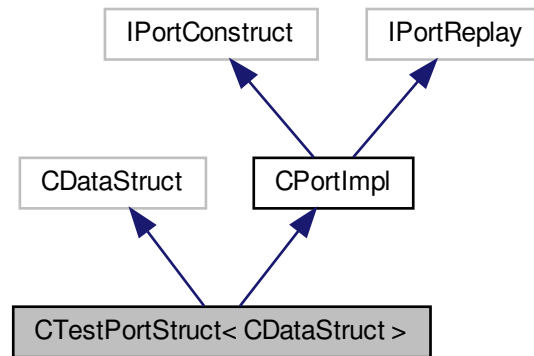
The documentation for this class was generated from the following file:

- `include/SpinGenApi/SelectorSet.h`

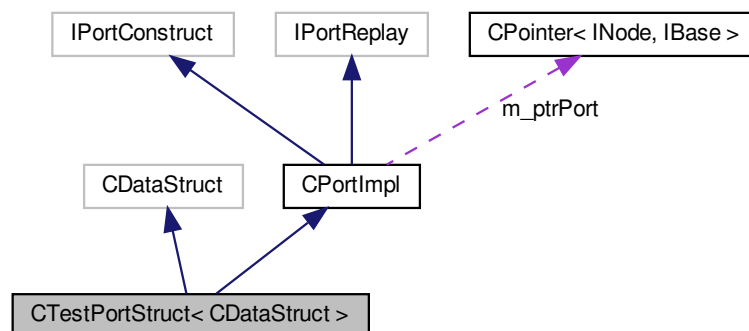
15.51 CTestPortStruct< CDataStruct > Class Template Reference

Implements a register spaces based on a C++ struct.

Inheritance diagram for CTestPortStruct< CDataStruct >:



Collaboration diagram for CTestPortStruct< CDataStruct >:



Public Member Functions

- [CTestPortStruct](#) (int64_t BaseAddress=0)
- virtual EAccessMode [GetAccessMode](#) () const
Get the access mode of the node.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t Address, int64_t Length)
Reads a chunk of bytes from the port.

- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- void [MemSet](#) (const char FillValue)
- void [ResetStatistics](#) ()
Resets the read/write statistics.
- int64_t [GetNumReads](#) ()
Returns the number of reads since lastReset Statistics.
- int64_t [GetNumWrites](#) ()
Returns the number of writes since lastReset Statistics.

Protected Attributes

- int64_t [m_NumReads](#)
Number of reads since last reset.
- int64_t [m_NumWrites](#)
Number of writes since last reset.
- int64_t [m_BaseAddress](#)
the base address used for the struct

15.51.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

15.51.2 Constructor & Destructor Documentation

15.51.2.1 CTestPortStruct()

```
CTestPortStruct (
    int64_t BaseAddress = 0 ) [inline]
```

15.51.3 Member Function Documentation

15.51.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Get the access mode of the node.

Implements [CPortImpl](#).

15.51.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

15.51.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

15.51.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

15.51.3.5 MemSet()

```
void MemSet (
    const char FillValue ) [inline]
```

15.51.3.6 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

15.51.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

15.51.3.8 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Implements [CPortImpl](#).

15.51.4 Member Data Documentation

15.51.4.1 m_BaseAddress

```
int64_t m_BaseAddress [protected]
```

the base address used for the struct

15.51.4.2 m_NumReads

```
int64_t m_NumReads [protected]
```

Number of reads since last reset.

15.51.4.3 m_NumWrites

```
int64_t m_NumWrites [protected]
```

Number of writes since last reset.

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

- include/SpinGenApi/[StructPort.h](#)

15.52 DCAM_CHECKSUM Struct Reference

Public Attributes

- uint32_t [CRCChecksum](#)

15.52.1 Member Data Documentation

15.52.1.1 CRCChecksum

uint32_t CRCChecksum

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

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

15.53 DCAM_CHUNK_TRAILER Struct Reference

Public Attributes

- SPIN_GUID [ChunkID](#)
- uint32_t [ChunkLength](#)
- uint32_t [InverseChunkLength](#)

15.53.1 Member Data Documentation

15.53.1.1 ChunkID

SPIN_GUID ChunkID

15.53.1.2 ChunkLength

uint32_t ChunkLength

15.53.1.3 InverseChunkLength

uint32_t InverseChunkLength

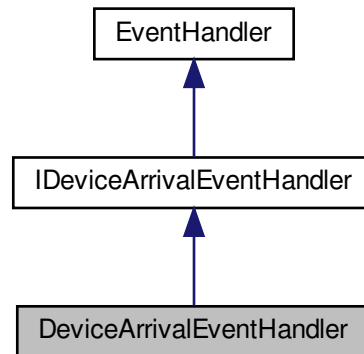
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

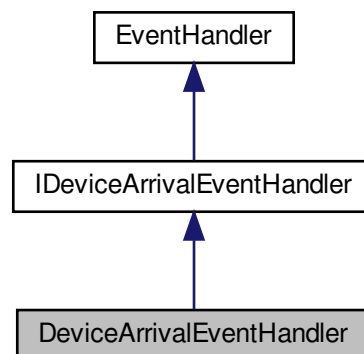
15.54 DeviceArrivalEventHandler Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for DeviceArrivalEventHandler:



Collaboration diagram for DeviceArrivalEventHandler:



Public Member Functions

- [DeviceArrivalEventHandler](#) ()
Default constructor.
- virtual [~DeviceArrivalEventHandler](#) ()
Virtual destructor.
- virtual void [OnDeviceArrival](#) ([CameraPtr](#) pCamera)=0
Callback to the device arrival event.

Protected Member Functions

- [DeviceArrivalEventHandler](#) & `operator=` (const [DeviceArrivalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.54.1 Detailed Description

An event handler for capturing the device arrival event.

15.54.2 Constructor & Destructor Documentation

15.54.2.1 DeviceArrivalEventHandler()

```
DeviceArrivalEventHandler ( )
```

Default constructor.

15.54.2.2 ~DeviceArrivalEventHandler()

```
virtual ~DeviceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

15.54.3 Member Function Documentation

15.54.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    CameraPtr pCamera ) [pure virtual]
```

Callback to the device arrival event.

Parameters

<i>pCamera</i>	Reference tracked CameraPtr object of the camera attached to the system

Implements [IDeviceArrivalEventHandler](#).

15.54.3.2 operator=()

```
DeviceArrivalEventHandler& operator= (  
    const DeviceArrivalEventHandler & ) [protected]
```

Assignment operator.

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

- include/[DeviceArrivalEventHandler.h](#)

15.55 DeviceEventExposureEndData Struct Reference

Data Fields for Device Event payload for EventExposureEnd.

Public Attributes

- uint64_t [frameID](#)

15.55.1 Detailed Description

Data Fields for Device Event payload for EventExposureEnd.

15.55.2 Member Data Documentation

15.55.2.1 frameID

```
uint64_t frameID
```

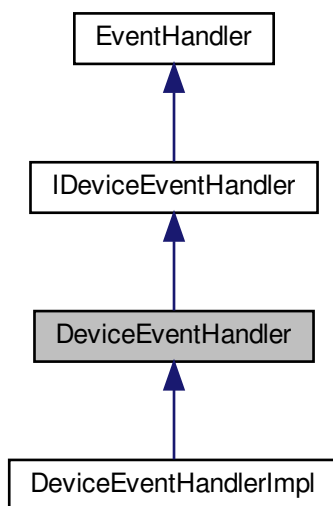
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

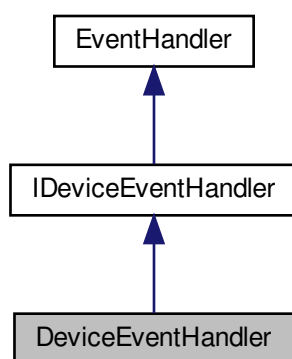
15.56 DeviceEventHandler Class Reference

A handler to device events.

Inheritance diagram for DeviceEventHandler:



Collaboration diagram for DeviceEventHandler:



Public Member Functions

- [DeviceEventHandler](#) ()

Default constructor.

- virtual `~DeviceEventHandler ()`

Virtual destructor.

- virtual void `OnDeviceEvent (Spinnaker::GenICam::gcstring eventName)=0`

Device event callback.

- uint64_t `GetDeviceEventId () const`

Get the ID of the device event.

- `GenICam::gcstring GetDeviceEventName () const`

Get the name of the device event.

Protected Member Functions

- `DeviceEventHandler & operator= (const DeviceEventHandler &)`

Assignment operator.

Additional Inherited Members

15.56.1 Detailed Description

A handler to device events.

15.56.2 Constructor & Destructor Documentation

15.56.2.1 DeviceEventHandler()

```
DeviceEventHandler ( )
```

Default constructor.

15.56.2.2 ~DeviceEventHandler()

```
virtual ~DeviceEventHandler ( ) [virtual]
```

Virtual destructor.

15.56.3 Member Function Documentation

15.56.3.1 GetDeviceEventId()

```
uint64_t GetDeviceEventId ( ) const [virtual]
```

Get the ID of the device event.

Returns

The device event ID

Implements [IDeviceEventHandler](#).

15.56.3.2 GetDeviceEventName()

```
GenICam::gcstring GetDeviceEventName ( ) const [virtual]
```

Get the name of the device event.

Returns

The device event name

Implements [IDeviceEventHandler](#).

15.56.3.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Device event callback.

Parameters

<i>eventName</i>	The name of the event

Implements [IDeviceEventHandler](#).

Implemented in [DeviceEventHandlerImpl](#).

15.56.3.4 operator=()

```
DeviceEventHandler& operator= (
    const DeviceEventHandler & ) [protected]
```

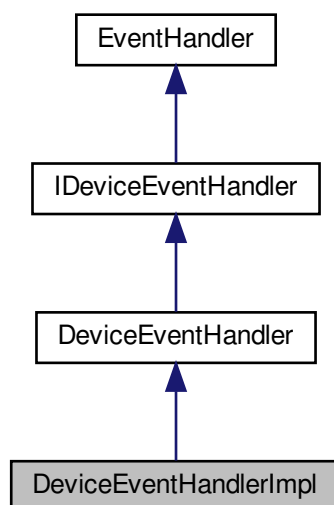
Assignment operator.

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

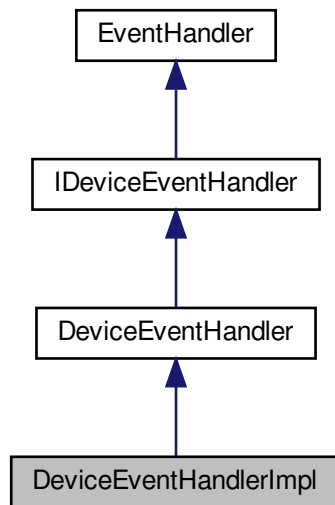
- [include/DeviceEventHandler.h](#)

15.57 DeviceEventHandlerImpl Class Reference

Inheritance diagram for DeviceEventHandlerImpl:



Collaboration diagram for DeviceEventHandlerImpl:



Public Member Functions

- [DeviceEventHandlerImpl](#) ([gcstring](#) eventName)
- [~DeviceEventHandlerImpl](#) ()
- void [OnDeviceEvent](#) ([gcstring](#) eventName)
Device event callback.

Additional Inherited Members

15.57.1 Constructor & Destructor Documentation

15.57.1.1 DeviceEventHandlerImpl()

```
DeviceEventHandlerImpl (  
    gcstring eventName ) [inline]
```

15.57.1.2 ~DeviceEventHandlerImpl()

```
~DeviceEventHandlerImpl ( ) [inline]
```


15.57.2 Member Function Documentation

15.57.2.1 OnDeviceEvent()

```
void OnDeviceEvent (
    gcstring eventName ) [inline], [virtual]
```

Device event callback.

Parameters

<i>eventName</i>	The name of the event

Implements [DeviceEventHandler](#).

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

- src/DeviceEvents/[DeviceEvents.cpp](#)

15.58 DeviceEventInferenceData Struct Reference

Data fields for Device Event payload for EventInference.

Public Attributes

- uint32_t [result](#)
- float [confidence](#)
- uint64_t [frameID](#)

15.58.1 Detailed Description

Data fields for Device Event payload for EventInference.

15.58.2 Member Data Documentation

15.58.2.1 confidence

```
float confidence
```

15.58.2.2 frameID

```
uint64_t frameID
```

15.58.2.3 result

```
uint32_t result
```

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

- [include/SpinnakerDefs.h](#)

15.59 DeviceEventUtility Class Reference

Static Public Member Functions

- static void [ParseDeviceEventInference](#) (const uint8_t *payloadData, const size_t payloadSize, [DeviceEventInferenceData](#) &eventData)
Parse the EventInference device event payload data.
- static void [ParseDeviceEventExposureEnd](#) (const uint8_t *payloadData, const size_t payloadSize, [DeviceEventExposureEndData](#) &eventData)
Parse the EventExposureEnd device event payload data.

15.59.1 Member Function Documentation

15.59.1.1 ParseDeviceEventExposureEnd()

```
static void ParseDeviceEventExposureEnd (
    const uint8_t * payloadData,
    const size_t payloadSize,
    DeviceEventExposureEndData & eventData ) [static]
```

Parse the EventExposureEnd device event payload data.

Parameters

in	<i>payloadData</i>	Event payload data
in	<i>payloadSize</i>	Event payload data size
out	<i>eventData</i>	Parsed ExposureEnd payload data returned by reference

See also

[DeviceEventExposureEndData](#)

15.59.1.2 ParseDeviceEventInference()

```
static void ParseDeviceEventInference (
    const uint8_t * payloadData,
    const size_t payloadSize,
    DeviceEventInferenceData & eventData ) [static]
```

Parse the EventInference device event payload data.

Parameters

in	<i>payloadData</i>	Event payload data
in	<i>payloadSize</i>	Event payload data size
out	<i>eventData</i>	Parsed EventInference payload data

See also

[DeviceEventInferenceData](#)

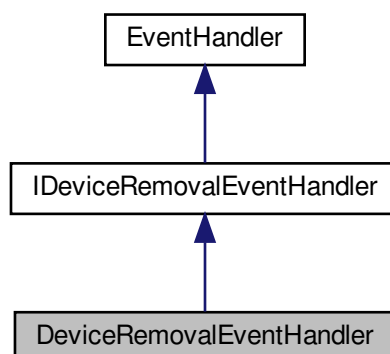
The documentation for this class was generated from the following file:

- include/[DeviceEventUtility.h](#)

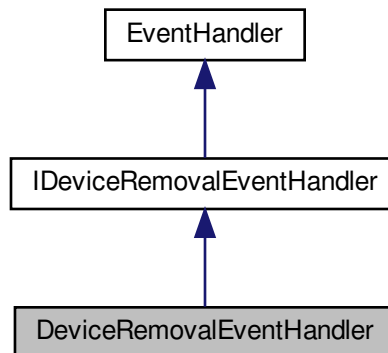
15.60 DeviceRemovalEventHandler Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for DeviceRemovalEventHandler:



Collaboration diagram for DeviceRemovalEventHandler:



Public Member Functions

- [DeviceRemovalEventHandler](#) ()
Default Constructor.
- virtual [~DeviceRemovalEventHandler](#) ()
Virtual Destructor.
- virtual void [OnDeviceRemoval](#) ([CameraPtr](#) pCamera)=0
Device removal event callback.

Protected Member Functions

- [DeviceRemovalEventHandler](#) & [operator=](#) (const [DeviceRemovalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.60.1 Detailed Description

An event handler for capturing the device removal event.

15.60.2 Constructor & Destructor Documentation

15.60.2.1 DeviceRemovalEventHandler()

`DeviceRemovalEventHandler ( )`

Default Constructor.

15.60.2.2 ~DeviceRemovalEventHandler()

`virtual ~DeviceRemovalEventHandler ( ) [virtual]`

Virtual Destructor.

15.60.3 Member Function Documentation

15.60.3.1 OnDeviceRemoval()

`virtual void OnDeviceRemoval (
 CameraPtr pCamera) [pure virtual]`

Device removal event callback.

Parameters

<code>pCamera</code>	Reference tracked CameraPtr object of the camera removed from the system

Implements [IDeviceRemovalEventHandler](#).

15.60.3.2 operator=()

`DeviceRemovalEventHandler& operator= (
 const DeviceRemovalEventHandler &) [protected]`

Assignment operator.

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

- include/[DeviceRemovalEventHandler.h](#)

15.61 double_autovector_t Class Reference

Vector of doubles with reference counting.

Public Member Functions

- [double_autovector_t](#) ()
- [double_autovector_t](#) (const [double_autovector_t](#) &obj)
- [double_autovector_t](#) (size_t n)
- virtual [~double_autovector_t](#) (void)
- [double_autovector_t](#) & [operator=](#) (const [double_autovector_t](#) &obj)
- void [operator delete](#) (void *pWhere)
- void * [operator new](#) (size_t uiSize)
- double & [operator\[\]](#) (size_t uiIndex)
- const double & [operator\[\]](#) (size_t uiIndex) const
- size_t [size](#) () const

Protected Attributes

- std::vector< double > * [_pv](#)
- ATOMIC_VARIABLE * [_pCount](#)

15.61.1 Detailed Description

Vector of doubles with reference counting.

15.61.2 Constructor & Destructor Documentation

15.61.2.1 [double_autovector_t](#)() [1/3]

```
double\_autovector\_t ( )
```

15.61.2.2 [double_autovector_t](#)() [2/3]

```
double\_autovector\_t (
    const double\_autovector\_t & obj )
```

15.61.2.3 [double_autovector_t](#)() [3/3]

```
double\_autovector\_t (
    size_t n ) [explicit]
```

15.61.2.4 ~double_autovector_t()

```
virtual ~double_autovector_t (
    void ) [virtual]
```

15.61.3 Member Function Documentation

15.61.3.1 operator delete()

```
void operator delete (
    void * pWhere )
```

15.61.3.2 operator new()

```
void* operator new (
    size_t uiSize )
```

15.61.3.3 operator=()

```
double_autovector_t& operator= (
    const double_autovector_t & obj )
```

15.61.3.4 operator[]() [1/2]

```
double& operator[] (
    size_t uiIndex )
```

15.61.3.5 operator[]() [2/2]

```
const double& operator[] (
    size_t uiIndex ) const
```

15.61.3.6 size()

```
size_t size ( ) const
```

15.61.4 Member Data Documentation

15.61.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

15.61.4.2 _pv

```
std::vector<double>* _pv [protected]
```

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

- include/SpinGenApi/[Autovector.h](#)

15.62 EAccessModeClass Class Reference

Holds conversion methods for the access mode enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EAccessMode *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EAccessMode *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EAccessMode Value)
Converts a string to an int32_t property.

15.62.1 Detailed Description

Holds conversion methods for the access mode enumeration.

15.62.2 Member Function Documentation

15.62.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to enum value.

15.62.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EAccessMode Value ) [static]
```

Converts a string to an int32_t property.

15.62.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.63 ECachingModeClass Class Reference

Holds conversion methods for the caching mode enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, ECachingMode *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, ECachingMode *pValue)
- static [GenICam::gcstring ToString](#) (ECachingMode Value)
Converts a string to an int32_t property.

15.63.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

15.63.2 Member Function Documentation

15.63.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

Converts a string to enum value.

15.63.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ECachingMode Value ) [static]
```

Converts a string to an int32_t property.

15.63.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

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

- include/SpinGenApi/[EnumClasses.h](#)

15.64 EDisplayNotationClass Class Reference

Holds conversion methods for the notation type of floats.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EDisplayNotation *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EDisplayNotation *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EDisplayNotation Value)
Converts a string to an int32_t property.

15.64.1 Detailed Description

Holds conversion methods for the notation type of floats.

15.64.2 Member Function Documentation

15.64.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EDisplayNotation * pValue ) [static]
```

Converts a string to enum value.

15.64.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    EDisplayNotation Value ) [static]
```

Converts a string to an int32_t property.

15.64.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EDisplayNotation * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.65 EEndianessClass Class Reference

Holds conversion methods for the endianness enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EEndianness *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EEndianness *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EEndianness Value)
Converts a string to an int32_t property.

15.65.1 Detailed Description

Holds conversion methods for the endianness enumeration.

15.65.2 Member Function Documentation

15.65.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EEndianness * pValue ) [static]
```

Converts a string to enum value.

15.65.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    EEndianness Value ) [static]
```

Converts a string to an int32_t property.

15.65.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EEndianness * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.66 EGenApiSchemaVersionClass Class Reference

helper class converting EGenApiSchemaVersion from and to string

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EGenApiSchemaVersion *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EGenApiSchemaVersion *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EGenApiSchemaVersion Value)
Converts a string to an int32_t property.

15.66.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

15.66.2 Member Function Documentation

15.66.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to enum value.

15.66.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EGenApiSchemaVersion Value ) [static]
```

Converts a string to an int32_t property.

15.66.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.67 EInputDirectionClass Class Reference

Holds conversion methods for the notation type of floats.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EInputDirection *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EInputDirection *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EInputDirection Value)
Converts a string to an int32_t property.

15.67.1 Detailed Description

Holds conversion methods for the notation type of floats.

15.67.2 Member Function Documentation

15.67.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to enum value.

15.67.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EInputDirection Value ) [static]
```

Converts a string to an int32_t property.

15.67.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.68 ENamespaceClass Class Reference

Holds conversion methods for the namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, ENamespace *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, ENamespace *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (ENamespace Value)
Converts a string to an int32_t property.

15.68.1 Detailed Description

Holds conversion methods for the namespace enumeration.

15.68.2 Member Function Documentation

15.68.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    ENamespace * pValue ) [static]
```

Converts a string to enum value.

15.68.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    ENamespace Value ) [static]
```

Converts a string to an int32_t property.

15.68.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    ENamespace * pValue ) [static]
```

Converts a string to an int32_t property.

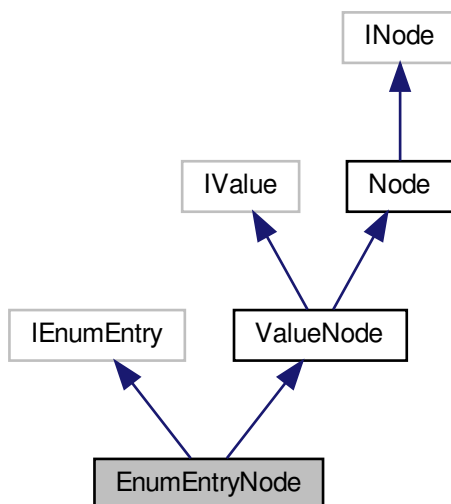
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

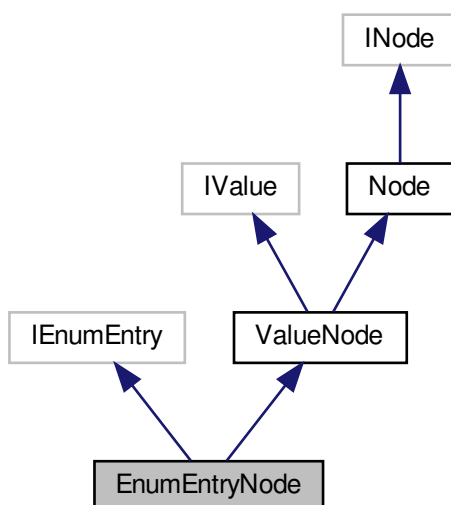
15.69 EnumEntryNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for EnumEntryNode:



Collaboration diagram for EnumEntryNode:



Public Member Functions

- [EnumEntryNode](#) ()
- [EnumEntryNode](#) (std::shared_ptr< Node::NodeImpl > pEnumEntry)
- virtual [~EnumEntryNode](#) ()
- virtual int64_t [GetValue](#) ()
Get numeric enum value.
- virtual [GenlCam::gcstring](#) [GetSymbolic](#) () const
Get symbolic enum value.
- virtual double [GetNumericValue](#) ()
Get double number associated with the entry.
- virtual bool [IsSelfClearing](#) ()
Indicates if the corresponding EnumEntry is self clearing.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for EnumEntry

Additional Inherited Members

15.69.1 Detailed Description

[Interface](#) for string properties.

15.69.2 Constructor & Destructor Documentation

15.69.2.1 EnumEntryNode() [1/2]

```
EnumEntryNode ( )
```

15.69.2.2 EnumEntryNode() [2/2]

```
EnumEntryNode (
    std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

15.69.2.3 ~EnumEntryNode()

```
virtual ~EnumEntryNode ( ) [virtual]
```

15.69.3 Member Function Documentation

15.69.3.1 GetNumericValue()

```
virtual double GetNumericValue ( ) [virtual]
```

Get double number associated with the entry.

15.69.3.2 GetSymbolic()

```
virtual GenICam::gcstring GetSymbolic ( ) const [virtual]
```

Get symbolic enum value.

15.69.3.3 GetValue()

```
virtual int64_t GetValue ( ) [virtual]
```

Get numeric enum value.

15.69.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( ) [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

15.69.3.5 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

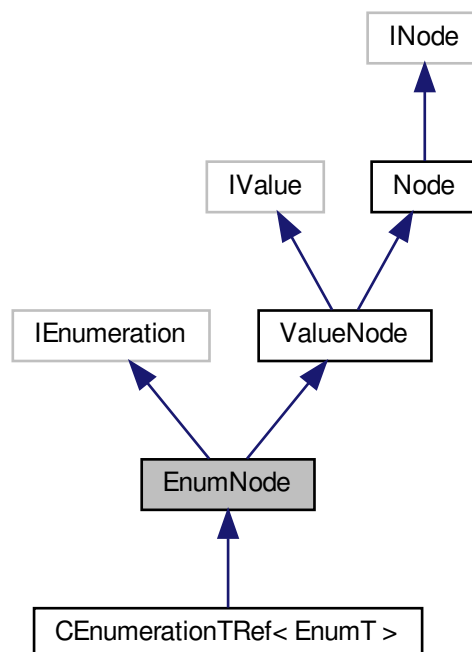
The documentation for this class was generated from the following file:

- [include/SpinGenApi/EnumEntryNode.h](#)

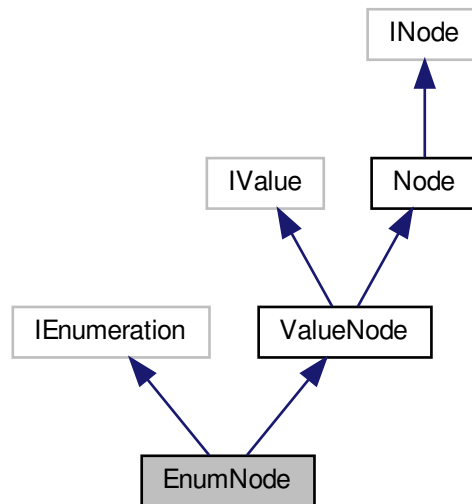
15.70 EnumNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for EnumNode:



Collaboration diagram for EnumNode:



Public Member Functions

- [EnumNode](#) ()
- [EnumNode](#) (std::shared_ptr< Node::NodeImpl > pEnumeration)
- virtual [~EnumNode](#) ()
- virtual void [GetSymbolics](#) (StringList_t &Symbolics)
Get list of symbolic Values.
- virtual void [GetEntries](#) (NodeList_t &Entries)
Get list of entry nodes.
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool [Verify](#)=true)
Set integer node value.
- virtual [GenICam::gcstring](#) [operator*](#) ()
Get string node value.
- virtual int64_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get integer node value.
- virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)
Get an entry node by name.
- virtual [IEnumEntry](#) * [GetEntry](#) (const int64_t IntValue)
Get an entry node by its IntValue.
- virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get the current entry.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Enumeration

Protected Attributes

- `std::shared_ptr< Node::NodeImpl > m_pEnumeration`

15.70.1 Detailed Description

[Interface](#) for string properties.

15.70.2 Constructor & Destructor Documentation

15.70.2.1 EnumNode() [1/2]

```
EnumNode ( )
```

15.70.2.2 EnumNode() [2/2]

```
EnumNode (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

15.70.2.3 ~EnumNode()

```
virtual ~EnumNode ( ) [virtual]
```

15.70.3 Member Function Documentation

15.70.3.1 GetCurrentEntry()

```
virtual IEnumEntry* GetCurrentEntry (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.70.3.2 GetEntries()

```
virtual void GetEntries (
    NodeList_t & Entries ) [virtual]
```

Get list of entry nodes.

15.70.3.3 GetEntry()

```
virtual IEnumEntry* GetEntry (
    const int64_t IntValue ) [virtual]
```

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.70.3.4 GetEntryByName()

```
virtual IEnumEntry* GetEntryByName (
    const GenICam::gcstring & Symbolic ) [virtual]
```

Get an entry node by name.

15.70.3.5 GetIntValue()

```
virtual int64_t GetIntValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get integer node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.70.3.6 GetSymbolics()

```
virtual void GetSymbolics (
    StringList_t & Symbolics ) [virtual]
```

Get list of symbolic Values.

15.70.3.7 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get string node value.

15.70.3.8 operator=()

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set string node value.

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.70.3.9 SetIntValue()

```
virtual void SetIntValue (
    int64_t Value,
    bool Verify = true ) [virtual]
```

Set integer node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.70.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

15.70.4 Member Data Documentation

15.70.4.1 m_pEnumeration

```
std::shared_ptr<Node::NodeImpl> m_pEnumeration [protected]
```

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

- include/SpinGenApi/[EnumNode.h](#)

15.71 ERepresentationClass Class Reference

Holds conversion methods for the representation enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, ERepresentation *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, ERepresentation *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (ERepresentation Value)
Converts a string to an int32_t property.

15.71.1 Detailed Description

Holds conversion methods for the representation enumeration.

15.71.2 Member Function Documentation

15.71.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to enum value.

15.71.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ERepresentation Value ) [static]
```

Converts a string to an int32_t property.

15.71.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.72 ESignClass Class Reference

Holds conversion methods for the sign enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, ESign *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, ESign *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (ESign Value)
Converts a string to an int32_t property.

15.72.1 Detailed Description

Holds conversion methods for the sign enumeration.

15.72.2 Member Function Documentation

15.72.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to enum value.

15.72.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ESign Value ) [static]
```

Converts a string to an int32_t property.

15.72.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.73 ESlopeClass Class Reference

Holds conversion methods for the converter formulas.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, ESlope *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, ESlope *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (ESlope Value)
Converts a string to an int32_t property.

15.73.1 Detailed Description

Holds conversion methods for the converter formulas.

15.73.2 Member Function Documentation

15.73.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    ESlope * pValue ) [static]
```

Converts a string to enum value.

15.73.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    ESlope Value ) [static]
```

Converts a string to an int32_t property.

15.73.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    ESlope * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.74 EStandardNameSpaceClass Class Reference

Holds conversion methods for the standard namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EStandardNameSpace *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EStandardNameSpace *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EStandardNameSpace Value)
Converts a string to an int32_t property.

15.74.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

15.74.2 Member Function Documentation

15.74.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EStandardNameSpace * pValue ) [static]
```

Converts a string to enum value.

15.74.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    EStandardNameSpace Value ) [static]
```

Converts a string to an int32_t property.

15.74.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EStandardNameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

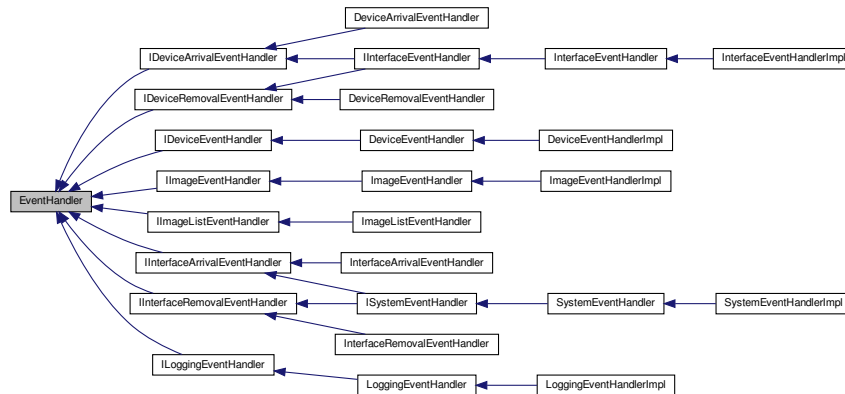
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

15.75 EventHandler Class Reference

The base class for all event handler types.

Inheritance diagram for EventHandler:



Public Member Functions

- virtual [~EventHandler](#) ()
Virtual Destructor.
- void [SetEventType](#) (EventType eventType)
Sets the event type.
- EventType [GetEventType](#) ()
Gets the event type.
- const uint8_t * [GetEventPayloadData](#) ()
Gets the event payload data.
- const size_t [GetEventPayloadDataSize](#) ()
Gets the event payload data size.

Protected Member Functions

- [EventHandler](#) ()
- [EventHandler](#) & [operator=](#) (const [EventHandler](#) &)
- void [SetEventPayload](#) (uint8_t *offset, size_t length)

Protected Attributes

- EventData * [m_pEventData](#)

Friends

- class [CallbackHandlerInternal](#)
- class [EventProcessor](#)
- class [IDataStream](#)
- class [Stream](#)
- class [ImageSetInspector](#)

15.75.1 Detailed Description

The base class for all event handler types.

15.75.2 Constructor & Destructor Documentation

15.75.2.1 ~EventHandler()

```
virtual ~EventHandler ( ) [virtual]
```

Virtual Destructor.

15.75.2.2 EventHandler()

```
EventHandler ( ) [protected]
```

15.75.3 Member Function Documentation

15.75.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData ( )
```

Gets the event payload data.

Returns

The event payload data

15.75.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

Returns

The event payload data size

15.75.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

Returns

The event type

15.75.3.4 operator=()

```
EventHandler& operator= (
    const EventHandler & ) [protected]
```

15.75.3.5 SetEventPayload()

```
void SetEventPayload (
    uint8_t * offset,
    size_t length ) [protected]
```

15.75.3.6 SetEventType()

```
void SetEventType (
    EventType eventType )
```

Sets the event type.

Parameters

<i>eventType</i>	The event type

15.75.4 Friends And Related Function Documentation

15.75.4.1 CallbackHandlerInternal

```
friend class CallbackHandlerInternal [friend]
```

15.75.4.2 EventProcessor

```
friend class EventProcessor [friend]
```

15.75.4.3 IDataStream

```
friend class IDataStream [friend]
```

15.75.4.4 ImageSetInspector

```
friend class ImageSetInspector [friend]
```

15.75.4.5 Stream

```
friend class Stream [friend]
```

15.75.5 Member Data Documentation

15.75.5.1 m_pEventData

```
EventData* m_pEventData [protected]
```

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

- include/[EventHandler.h](#)

15.76 EVisibilityClass Class Reference

Holds conversion methods for the visibility enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EVisibility *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EVisibility *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring](#) [ToString](#) (EVisibility Value)
Converts a string to an int32_t property.

15.76.1 Detailed Description

Holds conversion methods for the visibility enumeration.

15.76.2 Member Function Documentation

15.76.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to enum value.

15.76.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EVisibility Value ) [static]
```

Converts a string to an int32_t property.

15.76.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to an int32_t property.

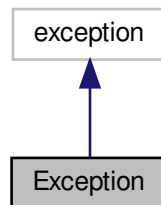
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

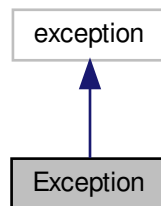
15.77 Exception Class Reference

The [Exception](#) object represents an error that is returned from the library.

Inheritance diagram for Exception:



Collaboration diagram for Exception:



Public Member Functions

- [Exception](#) ()
Default constructor.
- [Exception](#) (int line, const char *fileName, const char *funcName, const char *errMsg, [Error](#) err)
Message constructor.
- [Exception](#) (int line, const char *fileName, const char *funcName, const char *buildDate, const char *buildTime, const char *errMsg, [Error](#) err)
Message constructor.
- [Exception](#) (const [Exception](#) &except)
Copy constructor.
- virtual [~Exception](#) () throw ()
Default destructor.
- [Exception](#) & [operator=](#) (const [Exception](#) &except)
Assignment operator.

- bool `operator==` (const `Error` err) const
Equality operator.
- bool `operator!=` (const `Error` err) const
Inequality operator.
- virtual const char * `what` () const throw ()
virtual override for `what()`.
- const char * `GetFullErrorMessage` () const
Gets the error code and full error message including the line, file, function, build date, and time.
- const char * `GetErrorMessage` () const
Accessor Functions.
- const char * `GetFileName` () const
- const char * `GetFunctionName` () const
- const char * `GetBuildDate` () const
- const char * `GetBuildTime` () const
- int `GetLineNumber` () const
- `Error` `GetError` () const

15.77.1 Detailed Description

The `Exception` object represents an error that is returned from the library.

Overloaded operators allow comparisons against other `Exception` objects.

15.77.2 Constructor & Destructor Documentation

15.77.2.1 `Exception()` [1/4]

```
Exception ( )
```

Default constructor.

15.77.2.2 `Exception()` [2/4]

```
Exception (
    int line,
    const char * fileName,
    const char * funcName,
    const char * errMsg,
    Error err )
```

Message constructor.

Parameters

<i>line</i>	Line number where the exception is thrown
<i>fileName</i>	Name of the file called
<i>funcName</i>	Name of the function called
<i>errMsg</i>	A pointer to the exception message string
<i>err</i>	Error code

15.77.2.3 Exception() [3/4]

```
Exception (
    int line,
    const char * fileName,
    const char * funcName,
    const char * buildDate,
    const char * buildTime,
    const char * errMsg,
    Error err )
```

Message constructor.

Parameters

<i>line</i>	Line number where the exception is thrown
<i>fileName</i>	Name of the file called
<i>funcName</i>	Name of the function called
<i>buildDate</i>	Build date
<i>buildTime</i>	Build time
<i>errMsg</i>	A pointer to the exception message string
<i>err</i>	Error code

15.77.2.4 Exception() [4/4]

```
Exception (
    const Exception & except )
```

Copy constructor.

15.77.2.5 ~Exception()

```
virtual ~Exception ( ) throw ( )    [virtual]
```

Default destructor.

15.77.3 Member Function Documentation

15.77.3.1 GetBuildDate()

```
const char* GetBuildDate ( ) const
```

15.77.3.2 GetBuildTime()

```
const char* GetBuildTime ( ) const
```

15.77.3.3 GetError()

```
Error GetError ( ) const
```

15.77.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

15.77.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

15.77.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

15.77.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

15.77.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

15.77.3.9 operator"!="()

```
bool operator!= (
    const Error err ) const
```

Inequality operator.

15.77.3.10 operator=()

```
Exception& operator= (
    const Exception & except )
```

Assignment operator.

15.77.3.11 operator==()

```
bool operator== (
    const Error err ) const
```

Equality operator.

15.77.3.12 what()

```
virtual const char* what ( ) const throw ( ) [virtual]
```

virtual override for [what\(\)](#).

Gets the error code and error message.

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

- include/[Exception.h](#)

15.78 EYesNoClass Class Reference

Holds conversion methods for the standard namespace enumeration.

Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EYesNo *pValue)
Converts a string to enum value.
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EYesNo *pValue)
Converts a string to an int32_t property.
- static [GenICam::gcstring ToString](#) (EYesNo Value)
Converts a string to an int32_t property.

15.78.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

15.78.2 Member Function Documentation

15.78.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EYesNo * pValue ) [static]
```

Converts a string to enum value.

15.78.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (  
    EYesNo Value ) [static]
```

Converts a string to an int32_t property.

15.78.2.3 ToString() [2/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EYesNo * pValue ) [static]
```

Converts a string to an int32_t property.

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

- include/SpinGenApi/[EnumClasses.h](#)

15.79 FileProtocolAdapter Class Reference

Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.

Public Member Functions

- [FileProtocolAdapter](#) ()
Constructor.
- virtual [~FileProtocolAdapter](#) ()
- bool [attach](#) (::Spinnaker::GenApi::INodeMap *pInterface)
attach file protocol adapter to [NodeMap](#)
- bool [openFile](#) (const char *pFileName, std::ios_base::openmode mode)
open a file on the device
- bool [closeFile](#) (const char *pFileName)
close a file on the device
- std::streamsize [write](#) (const char *buf, int64_t offs, int64_t len, const char *pFileName)
writes data into a file.
- std::streamsize [read](#) (char *buf, int64_t offs, std::streamsize len, const char *pFileName)
read data from the device into a buffer
- int64_t [getBufSize](#) (const char *pFileName, std::ios_base::openmode mode)
fetch max FileAccessBuffer length for a file
- bool [deleteFile](#) (const char *pFileName)
Delete the content of the file.

15.79.1 Detailed Description

Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.

The adapter assumes, that the features provide stdio file access compatible semantic

15.79.2 Constructor & Destructor Documentation

15.79.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ( )
```

Constructor.

15.79.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter ( ) [virtual]
```


15.79.3 Member Function Documentation

15.79.3.1 attach()

```
bool attach (
    ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to [NodeMap](#)

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached

Returns

true if attach was successful, false if not

15.79.3.2 closeFile()

```
bool closeFile (
    const char * pFileName )
```

close a file on the device

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

15.79.3.3 deleteFile()

```
bool deleteFile (
    const char * pFileName )
```

Delete the content of the file.

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector

Returns

true on success, false on error

15.79.3.4 getBufSize()

```
int64_t getBufSize (
    const char * pFileName,
    std::ios_base::openmode mode )
```

fetch max FileAccessBuffer length for a file

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector
<i>mode</i>	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

max length of FileAccessBuffer in the given mode on the given file

15.79.3.5 openFile()

```
bool openFile (
    const char * pFileName,
    std::ios_base::openmode mode )
```

open a file on the device

Parameters

<i>pFileName</i>	filename of the file to open. The filename must exist in the Enumeration FileSelector
<i>mode</i>	mode to open the file. The mode must exist in the Enumeration FileOpenMode

Returns

true on success, false on error

15.79.3.6 read()

```
std::streamsize read (
    char * buf,
    int64_t offs,
    std::streamsize len,
    const char * pFileName )
```

read data from the device into a buffer

Parameters

<i>buf</i>	target buffer
<i>offs</i>	offset in the device file to read from
<i>len</i>	count of bytes to read
<i>pFileName</i>	filename of the file to write into The filename must exist in the Enumeration FileSelector

Returns

count of bytes successfully read

15.79.3.7 write()

```
std::streamsize write (
    const char * buf,
    int64_t offs,
    int64_t len,
    const char * pFileName )
```

writes data into a file.

Parameters

<i>buf</i>	source buffer
<i>offs</i>	offset into the device file
<i>len</i>	count of bytes to write
<i>pFileName</i>	filename of the file to write into The filename must exist in the Enumeration FileSelector

Returns

count of bytes written

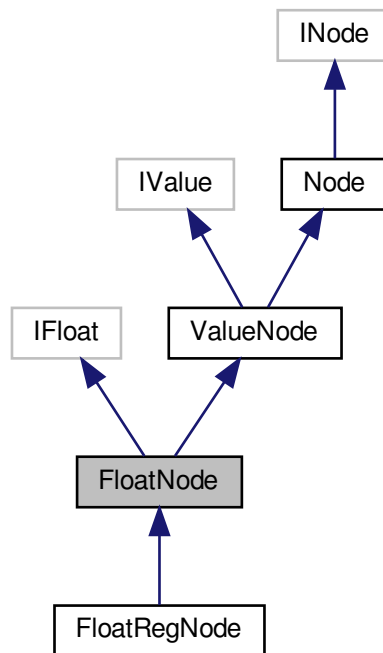
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Filestream.h](#)

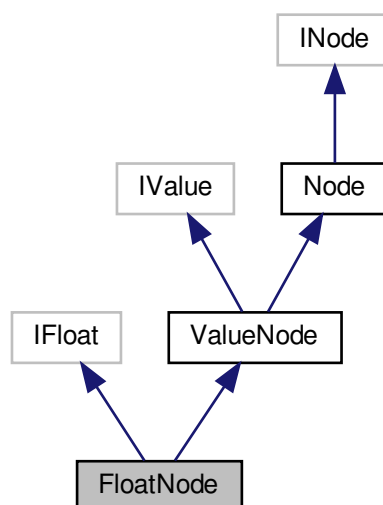
15.80 FloatNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatNode:



Collaboration diagram for FloatNode:



Public Member Functions

- [FloatNode](#) ()
- [FloatNode](#) (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatNode](#) ()
- virtual void [SetValue](#) (double Value, bool [Verify](#)=true)
Set node value.
- virtual [IFloat](#) & [operator=](#) (double Value)
Set node value.
- virtual double [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get node value.
- virtual double [operator\(\)](#) ()
Get node value.
- virtual double [operator*](#) ()
Get node value.
- virtual double [GetMin](#) ()
Get minimum value allowed.
- virtual double [GetMax](#) ()
Get maximum value allowed.
- virtual bool [HasInc](#) ()
True if the float has a constant increment.
- virtual EIncMode [GetIncMode](#) ()
Get increment mode.
- virtual double [GetInc](#) ()
Get the constant increment if there is any.
- virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
Get list of valid value.
- virtual ERepresentation [GetRepresentation](#) ()
Get recommended representation.
- virtual [GenlCam::gcstring](#) [GetUnit](#) () const
Get the physical unit name.
- virtual EDisplayNotation [GetDisplayNotation](#) () const
Get the way the float should be converted to a string.
- virtual int64_t [GetDisplayPrecision](#) () const
Get the precision to be used when converting the float to a string.
- [Integer](#) * [GetIntAlias](#) ()
gets the interface of an alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (double Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)
Restrict maximum value.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Float

Additional Inherited Members

15.80.1 Detailed Description

[Interface](#) for string properties.

15.80.2 Constructor & Destructor Documentation

15.80.2.1 FloatNode() [1/2]

```
FloatNode ( )
```

15.80.2.2 FloatNode() [2/2]

```
FloatNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

15.80.2.3 ~FloatNode()

```
virtual ~FloatNode ( ) [virtual]
```

15.80.3 Member Function Documentation

15.80.3.1 GetDisplayNotation()

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

Get the way the float should be converted to a string.

15.80.3.2 GetDisplayPrecision()

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

Get the precision to be used when converting the float to a string.

15.80.3.3 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

15.80.3.4 GetInc()

```
virtual double GetInc ( ) [virtual]
```

Get the constant increment if there is any.

15.80.3.5 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

15.80.3.6 GetIntAlias()

```
IInteger* GetIntAlias ( )
```

gets the interface of an alias node.

15.80.3.7 GetListOfValidValues()

```
virtual double_autovector_t GetListOfValidValues (
    bool bounded = true ) [virtual]
```

Get list of valid value.

15.80.3.8 GetMax()

```
virtual double GetMax ( ) [virtual]
```

Get maximum value allowed.

15.80.3.9 GetMin()

```
virtual double GetMin ( ) [virtual]
```

Get minimum value allowed.

15.80.3.10 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

15.80.3.11 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) const [virtual]
```

Get the physical unit name.

15.80.3.12 GetValue()

```
virtual double GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.80.3.13 HasInc()

```
virtual bool HasInc ( ) [virtual]
```

True if the float has a constant increment.

15.80.3.14 ImposeMax()

```
virtual void ImposeMax (
    double Value ) [virtual]
```

Restrict maximum value.

15.80.3.15 `ImposeMin()`

```
virtual void ImposeMin (
    double Value ) [virtual]
```

Restrict minimum value.

15.80.3.16 `operator>()()`

```
virtual double operator() ( ) [virtual]
```

Get node value.

15.80.3.17 `operator*()`

```
virtual double operator* ( ) [virtual]
```

Get node value.

15.80.3.18 `operator=()`

```
virtual IFloat& operator= (
    double Value ) [virtual]
```

Set node value.

15.80.3.19 `SetReference()`

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

15.80.3.20 `SetValue()`

```
virtual void SetValue (
    double Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

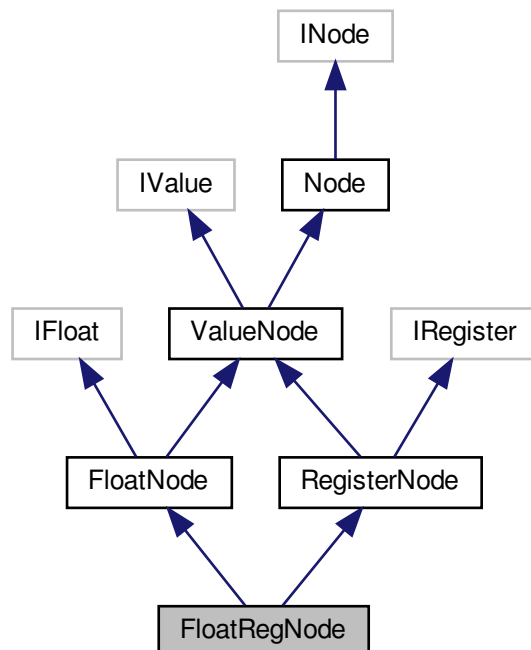
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatNode.h](#)

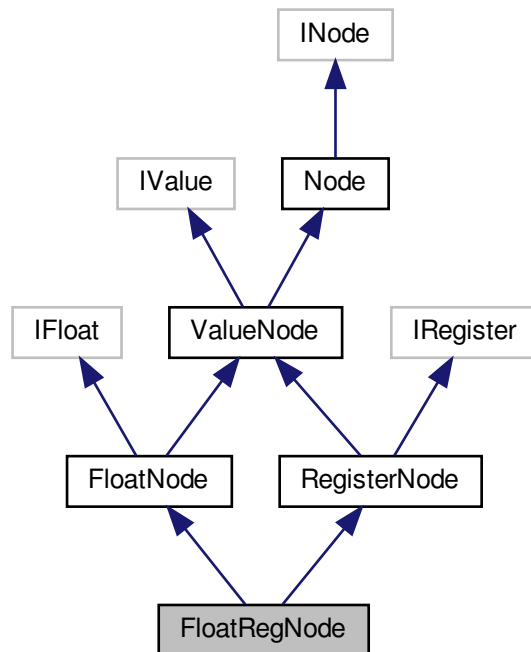
15.81 FloatRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatRegNode:



Collaboration diagram for FloatRegNode:



Public Member Functions

- [FloatRegNode](#) ()
- [FloatRegNode](#) (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.81.1 Detailed Description

[Interface](#) for string properties.

15.81.2 Constructor & Destructor Documentation

15.81.2.1 FloatRegNode() [1/2]

```
FloatRegNode ( )
```

15.81.2.2 FloatRegNode() [2/2]

```
FloatRegNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

15.81.2.3 ~FloatRegNode()

```
virtual ~FloatRegNode ( ) [virtual]
```

15.81.3 Member Function Documentation

15.81.3.1 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [FloatNode](#).

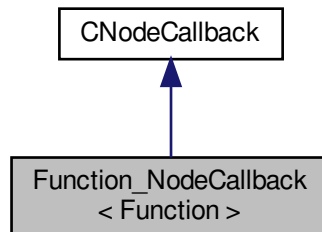
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatRegNode.h](#)

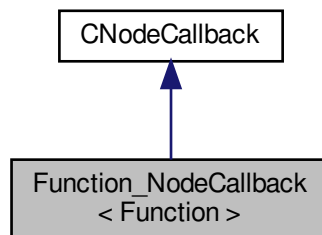
15.82 Function_NodeCallback< Function > Class Template Reference

Container for a function pointer.

Inheritance diagram for Function_NodeCallback< Function >:



Collaboration diagram for Function_NodeCallback< Function >:



Public Member Functions

- [Function_NodeCallback](#) (`INode *pNode`, `const Function &function`, `ECallbackType CallbackType`)
Constructor.
- virtual void [operator\(\)](#) (`ECallbackType CallbackType`) `const`
execute operation: call the function
- virtual void [Destroy](#) ()
destroys teh object

Additional Inherited Members

15.82.1 Detailed Description

```

template<class Function>
class Spinnaker::GenApi::Function_NodeCallback< Function >

```

Container for a function pointer.

15.82.2 Constructor & Destructor Documentation

15.82.2.1 `Function_NodeCallback()`

```
Function_NodeCallback (
    INode * pNode,
    const Function & function,
    ECallbackType CallbackType ) [inline]
```

Constructor.

15.82.3 Member Function Documentation

15.82.3.1 `Destroy()`

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys teh object

Implements [CNodeCallback](#).

15.82.3.2 `operator>()()`

```
virtual void operator() (
    ECallbackType CallbackType ) const [inline], [virtual]
```

execute operation: call the function

Implements [CNodeCallback](#).

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

- `include/SpinGenApi/NodeCallback.h`

15.83 gcstring Class Reference

Public Member Functions

- [gcstring](#) ()
- [gcstring](#) (const char *pc)
- [gcstring](#) (const char *pc, size_t n)
- [gcstring](#) (size_t count, char ch)
- [gcstring](#) (const [gcstring](#) &str)
- virtual [~gcstring](#) (void)
- virtual [gcstring](#) & [append](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [append](#) (size_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const [gcstring](#) &str)
- virtual [gcstring](#) & [assign](#) (size_t count, char ch)
- virtual [gcstring](#) & [assign](#) (const char *pc)
- virtual [gcstring](#) & [assign](#) (const char *pc, size_t n)
- virtual int [compare](#) (const [gcstring](#) &str) const
- virtual const char * [c_str](#) (void) const
- virtual bool [empty](#) (void) const
- virtual size_t [find](#) (char ch, size_t offset=0) const
- virtual size_t [find](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual size_t [find](#) (const [gcstring](#) &str, size_t offset, size_t count) const
- virtual size_t [find](#) (const char *pc, size_t offset=0) const
- virtual size_t [find](#) (const char *pc, size_t offset, size_t count) const
- virtual size_t [length](#) (void) const
- virtual size_t [size](#) (void) const
- virtual void [resize](#) (size_t n)
- virtual size_t [max_size](#) () const
- virtual [gcstring](#) [substr](#) (size_t offset=0, size_t count=[GCSTRING_NPOS](#)) const
- virtual size_t [find_first_of](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual size_t [find_first_not_of](#) (const [gcstring](#) &str, size_t offset=0) const
- virtual void [swap](#) ([gcstring](#) &Right)
- bool [operator!=](#) (const [gcstring](#) &str) const
- bool [operator!=](#) (const char *pc) const
- [gcstring](#) & [operator+=](#) (const [gcstring](#) &str)
- [gcstring](#) [operator+=](#) (const [gcstring](#) &str) const
- [gcstring](#) & [operator+=](#) (const char *pc)
- [gcstring](#) & [operator+=](#) (char ch)
- [gcstring](#) [operator+=](#) (char ch) const
- [gcstring](#) & [operator=](#) (const [gcstring](#) &str)
- bool [operator==](#) (const [gcstring](#) &str) const
- bool [operator==](#) (const char *pc) const
- bool [operator<](#) (const [gcstring](#) &str) const
- bool [operator>](#) (const [gcstring](#) &str) const
- [operator](#) const char * (void) const
- void [operator delete](#) (void *pWhere)
- void [operator delete](#) (void *pWhere, void *pNewWhere)
- void * [operator new](#) (size_t uiSize)
- void * [operator new](#) (size_t uiSize, void *pWhere)

Static Public Member Functions

- static size_t [_npos](#) (void)

Static Public Attributes

- static const size_t `npos`

Friends

- `SPINNAKER_API` friend `gcstring operator+` (const `gcstring` &left, const `gcstring` &right)
- `SPINNAKER_API` friend `gcstring operator+` (const `gcstring` &left, const char *right)
- `SPINNAKER_API` friend `gcstring operator+` (const char *left, const `gcstring` &right)

15.83.1 Constructor & Destructor Documentation

15.83.1.1 `gcstring()` [1/5]

```
gcstring ( )
```

15.83.1.2 `gcstring()` [2/5]

```
gcstring (
    const char * pc )
```

15.83.1.3 `gcstring()` [3/5]

```
gcstring (
    const char * pc,
    size_t n )
```

15.83.1.4 `gcstring()` [4/5]

```
gcstring (
    size_t count,
    char ch )
```

15.83.1.5 `gcstring()` [5/5]

```
gcstring (
    const gcstring & str )
```

15.83.1.6 `~gcstring()`

```
virtual ~gcstring (
    void ) [virtual]
```

15.83.2 Member Function Documentation

15.83.2.1 `_npos()`

```
static size_t _npos (
    void ) [static]
```

15.83.2.2 `append()` [1/2]

```
virtual gcstring& append (
    const gcstring & str ) [virtual]
```

15.83.2.3 `append()` [2/2]

```
virtual gcstring& append (
    size_t count,
    char ch ) [virtual]
```

15.83.2.4 `assign()` [1/4]

```
virtual gcstring& assign (
    const char * pc ) [virtual]
```

15.83.2.5 assign() [2/4]

```
virtual gcstring& assign (  
    const char * pc,  
    size_t n ) [virtual]
```

15.83.2.6 assign() [3/4]

```
virtual gcstring& assign (  
    const gcstring & str ) [virtual]
```

15.83.2.7 assign() [4/4]

```
virtual gcstring& assign (  
    size_t count,  
    char ch ) [virtual]
```

15.83.2.8 c_str()

```
virtual const char* c_str (  
    void ) const [virtual]
```

15.83.2.9 compare()

```
virtual int compare (  
    const gcstring & str ) const [virtual]
```

15.83.2.10 empty()

```
virtual bool empty (  
    void ) const [virtual]
```

15.83.2.11 find() [1/5]

```
virtual size_t find (  
    char ch,  
    size_t offset = 0 ) const [virtual]
```

15.83.2.12 find() [2/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset,  
    size_t count ) const [virtual]
```

15.83.2.13 find() [3/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset = 0 ) const [virtual]
```

15.83.2.14 find() [4/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset,  
    size_t count ) const [virtual]
```

15.83.2.15 find() [5/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

15.83.2.16 find_first_not_of()

```
virtual size_t find_first_not_of (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

15.83.2.17 find_first_of()

```
virtual size_t find_first_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

15.83.2.18 length()

```
virtual size_t length (
    void ) const [virtual]
```

15.83.2.19 max_size()

```
virtual size_t max_size ( ) const [virtual]
```

15.83.2.20 operator const char *()

```
operator const char * (
    void ) const
```

15.83.2.21 operator delete() [1/2]

```
void operator delete (
    void * pWhere )
```

15.83.2.22 operator delete() [2/2]

```
void operator delete (
    void * pWhere,
    void * pNewWhere )
```

15.83.2.23 operator new() [1/2]

```
void* operator new (
    size_t uiSize )
```

15.83.2.24 operator new() [2/2]

```
void* operator new (
    size_t uiSize,
    void * pWhere )
```

15.83.2.25 operator"!=() [1/2]

```
bool operator!= (
    const char * pc ) const
```

15.83.2.26 operator"!=() [2/2]

```
bool operator!= (
    const gcstring & str ) const
```

15.83.2.27 operator+=() [1/5]

```
gcstring& operator+= (
    char ch )
```

15.83.2.28 operator+=() [2/5]

```
gcstring operator+= (
    char ch ) const
```

15.83.2.29 operator+=() [3/5]

```
gcstring& operator+= (
    const char * pc )
```

15.83.2.30 operator+=() [4/5]

```
gcstring& operator+= (
    const gcstring & str )
```

15.83.2.31 operator+=() [5/5]

```
gcstring operator+= (
    const gcstring & str ) const
```

15.83.2.32 operator<()

```
bool operator< (
    const gcstring & str ) const
```

15.83.2.33 operator=()

```
gcstring& operator= (
    const gcstring & str )
```

15.83.2.34 operator==() [1/2]

```
bool operator== (
    const char * pc ) const
```

15.83.2.35 operator==() [2/2]

```
bool operator== (
    const gcstring & str ) const
```

15.83.2.36 operator>()

```
bool operator> (
    const gcstring & str ) const
```

15.83.2.37 resize()

```
virtual void resize (
    size_t n ) [virtual]
```

15.83.2.38 size()

```
virtual size_t size (
    void ) const [virtual]
```

15.83.2.39 substr()

```
virtual gcstring substr (
    size_t offset = 0,
    size_t count = GCSTRING_NPOS ) const [virtual]
```

15.83.2.40 swap()

```
virtual void swap (
    gcstring & Right ) [virtual]
```

15.83.3 Friends And Related Function Documentation**15.83.3.1 operator+ [1/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const char * left,
    const gcstring & right ) [friend]
```

15.83.3.2 operator+ [2/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const char * right ) [friend]
```

15.83.3.3 operator+ [3/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const gcstring & right ) [friend]
```


15.83.4 Member Data Documentation

15.83.4.1 npos

```
const size_t npos [static]
```

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

- [include/SpinGenApi/GCString.h](#)

15.84 GrabInfo Struct Reference

Public Member Functions

- [GrabInfo](#) (const string &deviceSerial)

Public Attributes

- unsigned int [numImagesGrabbed](#)
- unsigned int [numIncompleteImages](#)
- unsigned int [numRemovals](#)
- std::shared_ptr< [ImageEventHandlerImpl](#) > [imageEventHandler](#)

15.84.1 Constructor & Destructor Documentation

15.84.1.1 GrabInfo()

```
GrabInfo (  
    const string & deviceSerial ) [inline]
```

15.84.2 Member Data Documentation

15.84.2.1 imageEventHandler

```
std::shared_ptr<ImageEventHandlerImpl> imageEventHandler
```

15.84.2.2 numImagesGrabbed

unsigned int numImagesGrabbed

15.84.2.3 numIncompleteImages

unsigned int numIncompleteImages

15.84.2.4 numRemovals

unsigned int numRemovals

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

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)

15.85 GVCP_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

15.85.1 Detailed Description

header of a GVCP request packet

15.85.2 Member Data Documentation

15.85.2.1 ChunkID

uint32_t ChunkID

15.85.2.2 ChunkLength

uint32_t ChunkLength

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

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

15.86 GVCP_EVENT_ITEM Struct Reference

layout of a GVCP event item (Extended ID flag not set)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)
- uint16_t [StreamChannelId](#)
- uint16_t [BlockId](#)
- uint32_t [TimestampHigh](#)
- uint32_t [TimestampLow](#)

15.86.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

15.86.2 Member Data Documentation

15.86.2.1 BlockId

uint16_t BlockId

15.86.2.2 EventId

uint16_t EventId

15.86.2.3 ReservedOrEventSize

uint16_t ReservedOrEventSize

15.86.2.4 StreamChannelId

uint16_t StreamChannelId

15.86.2.5 TimestampHigh

uint32_t TimestampHigh

15.86.2.6 TimestampLow

uint32_t TimestampLow

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.87 GVCP_EVENT_ITEM_BASIC Struct Reference

layout of a GVCP event item (common to all types)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)

15.87.1 Detailed Description

layout of a GVCP event item (common to all types)

15.87.2 Member Data Documentation

15.87.2.1 EventId

uint16_t EventId

15.87.2.2 ReservedOrEventSize

uint16_t ReservedOrEventSize

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.88 GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference

layout of a GVCP event item (Extended ID flag set)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)
- uint16_t [StreamChannelId](#)
- uint16_t [BlockId](#)
- uint32_t [BlockId64High](#)
- uint32_t [BlockId64Low](#)
- uint32_t [TimestampHigh](#)
- uint32_t [TimestampLow](#)

15.88.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

15.88.2 Member Data Documentation

15.88.2.1 BlockId

uint16_t BlockId

15.88.2.2 BlockId64High

`uint32_t BlockId64High`

15.88.2.3 BlockId64Low

`uint32_t BlockId64Low`

15.88.2.4 EventId

`uint16_t EventId`

15.88.2.5 ReservedOrEventSize

`uint16_t ReservedOrEventSize`

15.88.2.6 StreamChannelId

`uint16_t StreamChannelId`

15.88.2.7 TimestampHigh

`uint32_t TimestampHigh`

15.88.2.8 TimestampLow

`uint32_t TimestampLow`

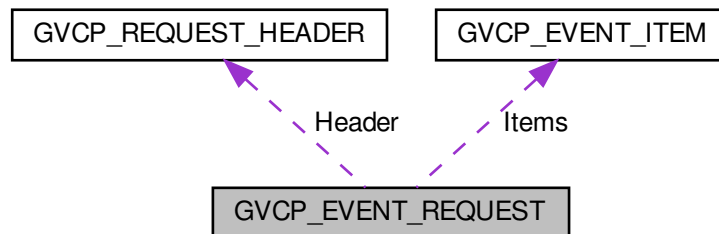
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterGEV.h`

15.89 GVCP_EVENT_REQUEST Struct Reference

Layout of a GVCP event request packet (Extended ID flag not set)

Collaboration diagram for GVCP_EVENT_REQUEST:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM](#) Items [1]

15.89.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

15.89.2 Member Data Documentation

15.89.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

15.89.2.2 Items

[GVCP_EVENT_ITEM](#) Items [1]

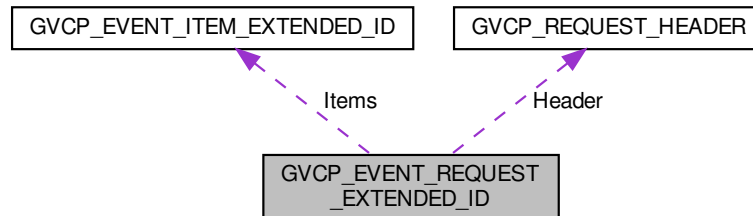
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.90 GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference

Layout of a GVCP event request packet (Extended ID flag set)

Collaboration diagram for GVCP_EVENT_REQUEST_EXTENDED_ID:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM_EXTENDED_ID](#) Items [1]

15.90.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

15.90.2 Member Data Documentation

15.90.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

15.90.2.2 Items

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Items [1]

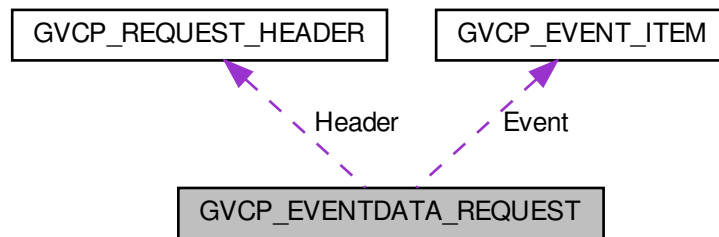
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.91 GVCP_EVENTDATA_REQUEST Struct Reference

Layout of a GVCP event data request packet (Extended ID flag not set)

Collaboration diagram for GVCP_EVENTDATA_REQUEST:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM](#) Event
- `uint32_t` [Data](#) [1]

15.91.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

15.91.2 Member Data Documentation

15.91.2.1 Data

```
uint32_t Data[1]
```

15.91.2.2 Event

[GVCP_EVENT_ITEM](#) Event

15.91.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

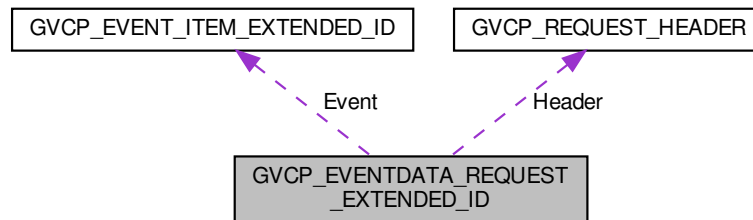
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterGEV.h`

15.92 GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference

Layout of a GVCP event data request packet (Extended ID flag set)

Collaboration diagram for GVCP_EVENTDATA_REQUEST_EXTENDED_ID:



Public Attributes

- [GVCP_REQUEST_HEADER](#) Header
- [GVCP_EVENT_ITEM_EXTENDED_ID](#) Event
- `uint32_t Data [1]`

15.92.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

15.92.2 Member Data Documentation

15.92.2.1 Data

`uint32_t Data[1]`

15.92.2.2 Event

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Event

15.92.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

15.93 GVCP_REQUEST_HEADER Struct Reference

header of a GVCP request packet

Public Attributes

- uint8_t [Magic](#)
- uint8_t [Flags](#)
- uint16_t [Command](#)
- uint16_t [Length](#)
- uint16_t [ReqId](#)

15.93.1 Detailed Description

header of a GVCP request packet

15.93.2 Member Data Documentation

15.93.2.1 Command

uint16_t Command

15.93.2.2 Flags

uint8_t Flags

15.93.2.3 Length

`uint16_t` Length

15.93.2.4 Magic

`uint8_t` Magic

15.93.2.5 ReqId

`uint16_t` ReqId

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

- `include/SpinGenApi/EventAdapterGEV.h`

15.94 H264Option Struct Reference

Options for saving H264 files.

Public Member Functions

- [H264Option](#) ()

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [bitrate](#)
Bit-rate to encode at.
- unsigned int [reserved](#) [256]
Reserved for future use.

15.94.1 Detailed Description

Options for saving H264 files.

15.94.2 Constructor & Destructor Documentation

15.94.2.1 H264Option()

```
H264Option ( ) [inline]
```

15.94.3 Member Data Documentation

15.94.3.1 bitrate

```
unsigned int bitrate
```

Bit-rate to encode at.

15.94.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

15.94.3.3 height

```
unsigned int height
```

Height of source image.

15.94.3.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

15.94.3.5 width

unsigned int width

Width of source image.

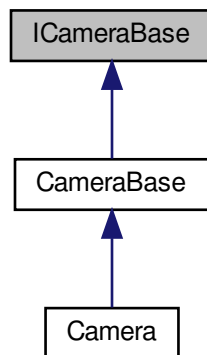
The documentation for this struct was generated from the following file:

- include/[SpinVideoDefs.h](#)

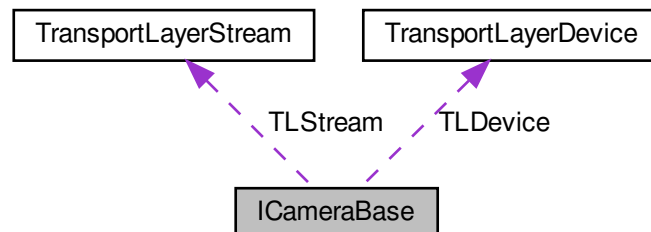
15.95 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:



Collaboration diagram for ICameraBase:



Public Member Functions

- virtual [~ICameraBase](#) (void)
- virtual void [Init](#) ()=0
- virtual void [DelInit](#) ()=0
- virtual bool [IsInitialized](#) ()=0
- virtual bool [IsValid](#) ()=0
- virtual [GenApi::INodeMap](#) & [GetNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetTLDeviceNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetTLStreamNodeMap](#) (uint64_t streamIndex) const =0
- virtual [GenApi::EAccessMode](#) [GetAccessMode](#) () const =0
- virtual void [ReadPort](#) (uint64_t iAddress, void *pBuffer, size_t iSize)=0
- virtual void [WritePort](#) (uint64_t iAddress, const void *pBuffer, size_t iSize)=0
- virtual void [BeginAcquisition](#) ()=0
- virtual void [EndAcquisition](#) ()=0
- virtual [BufferOwnership](#) [GetBufferOwnership](#) () const =0
- virtual void [SetBufferOwnership](#) (const [BufferOwnership](#) mode)=0
- virtual uint64_t [GetUserBufferCount](#) () const =0
- virtual uint64_t [GetUserBufferSize](#) () const =0
- virtual uint64_t [GetUserBufferTotalSize](#) () const =0
- virtual void [SetUserBuffers](#) (void *const pMemBuffers, uint64_t totalSize)=0
- virtual void [SetUserBuffers](#) (void **const ppMemBuffers, const uint64_t bufferCount, const uint64_t bufferSize)=0
- virtual [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout=[EVENT_TIMEOUT_INFINITE](#), uint64_t streamIndex=0)=0
- virtual [GenICam::gcstring](#) [GetUniqueID](#) ()=0
- virtual bool [IsStreaming](#) () const =0
- virtual [GenICam::gcstring](#) [GetGuiXml](#) () const =0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)=0
- virtual void [RegisterEventHandler](#) ([DeviceEventHandler](#) &evtHandlerToRegister, const [GenICam::gcstring](#) &eventName)=0
- virtual void [RegisterEventHandler](#) ([ImageEventHandler](#) &evtHandlerToRegister, uint64_t streamIndex)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual unsigned int [GetNumImagesInUse](#) ()=0
- virtual unsigned int [GetNumDataStreams](#) ()=0
- virtual unsigned int [DiscoverMaxPacketSize](#) ()=0
- virtual void [ForceIP](#) ()=0

Public Attributes

- [TransportLayerDevice](#) [TLDevice](#)
Gets vital camera information by connecting to the camera's bootstrap registers.
- [TransportLayerStream](#) [TLStream](#)
Gets information about the stream data by connecting to the camera's bootstrap registers.

Protected Member Functions

- [ICameraBase](#) ()
- [ICameraBase](#) (const [ICameraBase](#) &)
- [ICameraBase](#) & [operator=](#) (const [ICameraBase](#) &)

Protected Attributes

- CameraBaseData * [m_pCameraBaseData](#)

Friends

- class [CameraInternal](#)
- class [InterfaceImpl](#)

15.95.1 Detailed Description

The interface file for base class for the camera object.

15.95.2 Constructor & Destructor Documentation

15.95.2.1 ~ICameraBase()

```
virtual ~ICameraBase (
    void ) [inline], [virtual]
```

15.95.2.2 ICameraBase() [1/2]

```
ICameraBase ( ) [inline], [protected]
```

15.95.2.3 ICameraBase() [2/2]

```
ICameraBase (
    const ICameraBase & ) [inline], [protected]
```

15.95.3 Member Function Documentation

15.95.3.1 BeginAcquisition()

```
virtual void BeginAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.2 DeInit()

```
virtual void DeInit ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.4 EndAcquisition()

```
virtual void EndAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.5 ForceIP()

```
virtual void ForceIP ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.8 GetGuiXml()

```
virtual GenICam::gcstring GetGuiXml ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE,
    uint64_t streamIndex = 0 ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.10 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.14 GetTLStreamNodeMap()

```
virtual GenApi::INodeMap& GetTLStreamNodeMap (
    uint64_t streamIndex ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.15 GetUniqueID()

```
virtual GenICam::gcstring GetUniqueID ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.16 GetUserBufferCount()

```
virtual uint64_t GetUserBufferCount ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.17 GetUserBufferSize()

```
virtual uint64_t GetUserBufferSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.18 GetUserBufferTotalSize()

```
virtual uint64_t GetUserBufferTotalSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.19 Init()

```
virtual void Init ( ) [pure virtual]
```

Implemented in [CameraBase](#), and [Camera](#).

15.95.3.20 IsInitialized()

```
virtual bool IsInitialized ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.21 IsStreaming()

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.22 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.23 operator=()

```
ICameraBase& operator= (
    const ICameraBase & ) [protected]
```

15.95.3.24 ReadPort()

```
virtual void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.25 RegisterEventHandler() [1/3]

```
virtual void RegisterEventHandler (
    DeviceEventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.26 RegisterEventHandler() [2/3]

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.27 RegisterEventHandler() [3/3]

```
virtual void RegisterEventHandler (
    ImageEventHandler & evtHandlerToRegister,
    uint64_t streamIndex ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.28 SetBufferOwnership()

```
virtual void SetBufferOwnership (
    const BufferOwnership mode ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.29 SetUserBuffers() [1/2]

```
virtual void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.30 SetUserBuffers() [2/2]

```
virtual void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.31 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.3.32 WritePort()

```
virtual void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

15.95.4 Friends And Related Function Documentation

15.95.4.1 CameraInternal

```
friend class CameraInternal [friend]
```

15.95.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

15.95.5 Member Data Documentation

15.95.5.1 m_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData [protected]
```

15.95.5.2 TLDevice

[TransportLayerDevice](#) TLDevice

Gets vital camera information by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera.

15.95.5.3 TLStream

[TransportLayerStream](#) TLStream

Gets information about the stream data by connecting to the camera's bootstrap registers.

These nodes also access host software modules and the nodes can be used without having to call [Init\(\)](#) on the camera. In cameras with multiple streams, will always represent Stream channel 0.

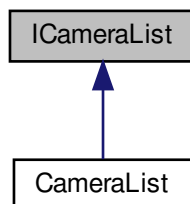
The documentation for this class was generated from the following file:

- [include/Interface/ICameraBase.h](#)

15.96 ICameraList Class Reference

Used to hold a list of camera objects.

Inheritance diagram for ICameraList:



Public Member Functions

- virtual [~ICameraList](#) ()
- virtual [CameraPtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [CameraPtr GetByIndex](#) (unsigned int index) const =0
- virtual [CameraPtr GetBySerial](#) (std::string serialNumber) const =0
- virtual [CameraPtr GetByDeviceID](#) (std::string deviceID) const =0
- virtual void [Clear](#) ()=0
- virtual void [Add](#) ([CameraPtr](#) camera)=0
- virtual void [Remove](#) ([CameraPtr](#) camera)=0
- virtual void [RemoveBySerial](#) (std::string serialNumber)=0
- virtual void [RemoveByIndex](#) (unsigned int index)=0
- virtual void [RemoveByDeviceID](#) (std::string deviceID)=0
- virtual void [Append](#) (const [CameraList](#) &list)=0

Protected Member Functions

- [ICameraList](#) ()
- [ICameraList](#) (const [ICameraList](#) &)
- [ICameraList](#) & [operator=](#) (const [ICameraList](#) &)

Protected Attributes

- CameraListData * [m_pCameraListData](#)

Friends

- class [InterfacelImpl](#)
- class [CameraListImpl](#)

15.96.1 Detailed Description

Used to hold a list of camera objects.

15.96.2 Constructor & Destructor Documentation

15.96.2.1 ~ICameraList()

```
virtual ~ICameraList ( ) [inline], [virtual]
```

15.96.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

15.96.2.3 ICameraList() [2/2]

```
ICameraList (  
    const ICameraList & ) [inline], [protected]
```

15.96.3 Member Function Documentation

15.96.3.1 Add()

```
virtual void Add (
    CameraPtr camera ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.2 Append()

```
virtual void Append (
    const CameraList & list ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.3 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.4 GetByDeviceID()

```
virtual CameraPtr GetByDeviceID (
    std::string deviceID ) const [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.5 GetByIndex()

```
virtual CameraPtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.6 GetBySerial()

```
virtual CameraPtr GetBySerial (
    std::string serialNumber ) const [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.7 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.8 operator=()

```
ICameraList& operator= (
    const ICameraList & ) [protected]
```

15.96.3.9 operator[]()

```
virtual CameraPtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.10 Remove()

```
virtual void Remove (
    CameraPtr camera ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.11 RemoveByDeviceID()

```
virtual void RemoveByDeviceID (
    std::string deviceID ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.12 RemoveByIndex()

```
virtual void RemoveByIndex (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.3.13 RemoveBySerial()

```
virtual void RemoveBySerial (
    std::string serialNumber ) [pure virtual]
```

Implemented in [CameraList](#).

15.96.4 Friends And Related Function Documentation

15.96.4.1 CameraListImpl

```
friend class CameraListImpl [friend]
```

15.96.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

15.96.5 Member Data Documentation

15.96.5.1 m_pCameraListData

```
CameraListData* m_pCameraListData [protected]
```

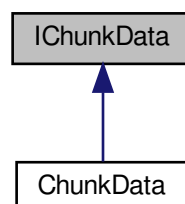
The documentation for this class was generated from the following file:

- [include/Interface/ICameraList.h](#)

15.97 IChunkData Class Reference

The [Interface](#) file for [ChunkData](#).

Inheritance diagram for IChunkData:



Public Member Functions

- virtual [~IChunkData](#) ()
- virtual void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)=0
- virtual [float64_t](#) [GetBlackLevel](#) () const =0
- virtual [int64_t](#) [GetFrameID](#) () const =0
- virtual [float64_t](#) [GetExposureTime](#) () const =0
- virtual [int64_t](#) [GetCompressionMode](#) () const =0
- virtual [float64_t](#) [GetCompressionRatio](#) () const =0
- virtual [int64_t](#) [GetTimestamp](#) () const =0
- virtual [int64_t](#) [GetExposureEndLineStatusAll](#) () const =0
- virtual [int64_t](#) [GetWidth](#) () const =0
- virtual [int64_t](#) [GetImage](#) () const =0
- virtual [int64_t](#) [GetHeight](#) () const =0
- virtual [float64_t](#) [GetGain](#) () const =0
- virtual [int64_t](#) [GetSequencerSetActive](#) () const =0
- virtual [int64_t](#) [GetCRC](#) () const =0
- virtual [int64_t](#) [GetOffsetX](#) () const =0
- virtual [int64_t](#) [GetOffsetY](#) () const =0
- virtual [int64_t](#) [GetSerialDataLength](#) () const =0
- virtual [int64_t](#) [GetPartSelector](#) () const =0
- virtual [int64_t](#) [GetPixelDynamicRangeMin](#) () const =0
- virtual [int64_t](#) [GetPixelDynamicRangeMax](#) () const =0
- virtual [int64_t](#) [GetTimestampLatchValue](#) () const =0
- virtual [int64_t](#) [GetLineStatusAll](#) () const =0
- virtual [int64_t](#) [GetCounterValue](#) () const =0
- virtual [float64_t](#) [GetTimerValue](#) () const =0
- virtual [int64_t](#) [GetScanLineSelector](#) () const =0
- virtual [int64_t](#) [GetEncoderValue](#) () const =0
- virtual [int64_t](#) [GetLinePitch](#) () const =0
- virtual [int64_t](#) [GetTransferBlockID](#) () const =0
- virtual [int64_t](#) [GetTransferQueueCurrentBlockCount](#) () const =0
- virtual [int64_t](#) [GetStreamChannelID](#) () const =0
- virtual [float64_t](#) [GetScan3dCoordinateScale](#) () const =0
- virtual [float64_t](#) [GetScan3dCoordinateOffset](#) () const =0
- virtual [float64_t](#) [GetScan3dInvalidDataValue](#) () const =0
- virtual [float64_t](#) [GetScan3dAxisMin](#) () const =0
- virtual [float64_t](#) [GetScan3dAxisMax](#) () const =0
- virtual [float64_t](#) [GetScan3dTransformValue](#) () const =0
- virtual [float64_t](#) [GetScan3dCoordinateReferenceValue](#) () const =0
- virtual [int64_t](#) [GetInferenceFrameId](#) () const =0
- virtual [int64_t](#) [GetInferenceResult](#) () const =0
- virtual [float64_t](#) [GetInferenceConfidence](#) () const =0
- virtual [InferenceBoundingBoxResult](#) [GetInferenceBoundingBoxResult](#) () const =0

Protected Member Functions

- [IChunkData](#) ()

Protected Attributes

- ChunkDataData * [m_pChunkDataData](#)

Friends

- class [ChunkDataImpl](#)
- class [ImageImpl](#)

15.97.1 Detailed Description

The [Interface](#) file for [ChunkData](#).

15.97.2 Constructor & Destructor Documentation

15.97.2.1 ~IChunkData()

```
virtual ~IChunkData ( ) [inline], [virtual]
```

15.97.2.2 IChunkData()

```
IChunkData ( ) [inline], [protected]
```

15.97.3 Member Function Documentation

15.97.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.2 GetCompressionMode()

```
virtual int64_t GetCompressionMode ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.3 GetCompressionRatio()

```
virtual float64_t GetCompressionRatio ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.4 GetCounterValue()

```
virtual int64_t GetCounterValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.5 GetCRC()

```
virtual int64_t GetCRC ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.6 GetEncoderValue()

```
virtual int64_t GetEncoderValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.7 GetExposureEndLineStatusAll()

```
virtual int64_t GetExposureEndLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.8 GetExposureTime()

```
virtual float64_t GetExposureTime ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.9 GetFrameID()

```
virtual int64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.10 GetGain()

```
virtual float64_t GetGain ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.11 GetHeight()

```
virtual int64_t GetHeight ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.12 GetImage()

```
virtual int64_t GetImage ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.13 GetInferenceBoundingBoxResult()

```
virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.14 GetInferenceConfidence()

```
virtual float64_t GetInferenceConfidence ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.15 GetInferenceFrameId()

```
virtual int64_t GetInferenceFrameId ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.16 GetInferenceResult()

```
virtual int64_t GetInferenceResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.17 GetLinePitch()

```
virtual int64_t GetLinePitch ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.18 GetLineStatusAll()

```
virtual int64_t GetLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.19 GetOffsetX()

```
virtual int64_t GetOffsetX ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.20 GetOffsetY()

```
virtual int64_t GetOffsetY ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.21 GetPartSelector()

```
virtual int64_t GetPartSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.22 GetPixelDynamicRangeMax()

```
virtual int64_t GetPixelDynamicRangeMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.23 GetPixelDynamicRangeMin()

```
virtual int64_t GetPixelDynamicRangeMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.24 GetScan3dAxisMax()

```
virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.25 GetScan3dAxisMin()

```
virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.26 GetScan3dCoordinateOffset()

```
virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.27 GetScan3dCoordinateReferenceValue()

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.28 GetScan3dCoordinateScale()

```
virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.29 GetScan3dInvalidDataValue()

```
virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.30 GetScan3dTransformValue()

```
virtual float64_t GetScan3dTransformValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.31 GetScanLineSelector()

```
virtual int64_t GetScanLineSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.32 GetSequencerSetActive()

```
virtual int64_t GetSequencerSetActive ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.33 GetSerialDataLength()

```
virtual int64_t GetSerialDataLength ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.34 GetStreamChannelID()

```
virtual int64_t GetStreamChannelID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.35 GetTimerValue()

```
virtual float64_t GetTimerValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.36 GetTimestamp()

```
virtual int64_t GetTimestamp ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.37 GetTimestampLatchValue()

```
virtual int64_t GetTimestampLatchValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.38 GetTransferBlockID()

```
virtual int64_t GetTransferBlockID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.39 GetTransferQueueCurrentBlockCount()

```
virtual int64_t GetTransferQueueCurrentBlockCount ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.40 GetWidth()

```
virtual int64_t GetWidth ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

15.97.3.41 SetChunks()

```
virtual void SetChunks (
    GenApi::INodeMap & pNodeMap ) [pure virtual]
```

Implemented in [ChunkData](#).

15.97.4 Friends And Related Function Documentation

15.97.4.1 ChunkDataImpl

```
friend class ChunkDataImpl [friend]
```

15.97.4.2 ImageImpl

```
friend class ImageImpl [friend]
```

15.97.5 Member Data Documentation

15.97.5.1 m_pChunkDataData

```
ChunkDataData* m_pChunkDataData [protected]
```

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

- [include/Interface/IChunkData.h](#)

15.98 IDataStream Class Reference

Public Member Functions

- virtual [~IDataStream](#) ()
- virtual std::string [GetStreamID](#) () const =0
- virtual uint64_t [GetStreamIndex](#) () const =0
- virtual void [ResetStreamHandle](#) (std::shared_ptr< Spinnaker::IGenTLDevice > pDevice)=0
- virtual [StreamTypeEnum](#) [GetStreamType](#) () const =0
- virtual void [AnnouncelImage](#) (size_t size)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pPrivate)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pData, void *pPrivate)=0
- virtual void [RevokelImages](#) ()=0
- virtual void [StartStream](#) (const unsigned int stream_index=0)=0
- virtual void [StopStream](#) ()=0
- virtual [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout)=0
- virtual [ImagePtr](#) [GetNextImageInternal](#) (void **ppPrivate, uint64_t grabTimeout)=0
- virtual void [ReleaseImageBuffer](#) (const BUFFER_HANDLE bufferHandle)=0
- virtual void [FlushQueueAllDiscard](#) ()=0
- virtual bool [IsStreaming](#) ()=0
- virtual void [KillBufferEvent](#) ()=0
- virtual bool [IsImageBufferInUse](#) (const BUFFER_HANDLE bufferHandle)=0
- virtual unsigned int [GetNumImagesInUse](#) () const =0
- virtual size_t [GetStreamInfoSizeType](#) (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual bool [GetStreamInfoBool8Type](#) (GenTL::STREAM_INFO_CMD iInfoCmd)=0
- virtual void * [GetBufferInfoPtrType](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual size_t [GetBufferInfoSizeType](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual uint64_t [GetBufferInfoUInt64Type](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual bool [GetBufferInfoBool8Type](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::BUFFER_INFO_CMD iInfoCmd)=0
- virtual std::shared_ptr< unsigned char > [GetSystemBuffer](#) (GenTL::BUFFER_HANDLE hBuffer)=0
- virtual void [RegisterImageEventHandler](#) (IImageEventHandler &imageEventHandler, EventPollingOptions pollingOption)=0
- virtual void [UnregisterImageEventHandler](#) (IImageEventHandler &imageEventHandler)=0
- virtual bool [IsEventHandlerRegistered](#) (IImageEventHandler &imageEventHandler)=0
- virtual bool [IsAutoPollingEventHandlerRegistered](#) ()=0
- virtual void [WaitOnImageEvent](#) (uint64_t timeout)=0
- virtual void [InitChunkAdapter](#) (GenApi::INodeMap &nodemap)=0
- virtual void [CleanupChunkAdapter](#) ()=0
- virtual GenTL::GC_ERROR [GetBufferChunkData](#) (GenTL::BUFFER_HANDLE hBuffer, GenTL::SINGLE_CHUNK_DATA *pChunkData, size_t *piNumChunks)=0
- virtual void [AttachBuffer](#) (uint8_t *pBuffer, GenApi::SingleChunkData_t *ChunkData, int64_t NumChunks)=0
- virtual bool [IsCRCCheckEnabled](#) () const =0
- virtual [GenApi::INodeMap](#) & [GetNodeMap](#) () const =0
- virtual [GenApi::INodeMap](#) * [GetDeviceNodeMap](#) () const =0
- virtual Port & [GetPort](#) () const =0
- virtual const [TransportLayerStream](#) & [TransportLayerStreamInfo](#) () const =0

Protected Member Functions

- [IDataStream](#) ()

15.98.1 Constructor & Destructor Documentation

15.98.1.1 ~IDataStream()

```
virtual ~IDataStream ( ) [inline], [virtual]
```

15.98.1.2 IDataStream()

```
IDataStream ( ) [inline], [protected]
```

15.98.2 Member Function Documentation

15.98.2.1 AnnounceImage() [1/3]

```
virtual void AnnounceImage (
    size_t size ) [pure virtual]
```

15.98.2.2 AnnounceImage() [2/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pData,
    void * pPrivate ) [pure virtual]
```

15.98.2.3 AnnounceImage() [3/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pPrivate ) [pure virtual]
```

15.98.2.4 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    GenApi::SingleChunkData_t * ChunkData,
    int64_t NumChunks ) [pure virtual]
```

15.98.2.5 CleanupChunkAdapter()

```
virtual void CleanupChunkAdapter ( ) [pure virtual]
```

15.98.2.6 FlushQueueAllDiscard()

```
virtual void FlushQueueAllDiscard ( ) [pure virtual]
```

15.98.2.7 GetBufferChunkData()

```
virtual GenTL::GC_ERROR GetBufferChunkData (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::SINGLE_CHUNK_DATA * pChunkData,
    size_t * piNumChunks ) [pure virtual]
```

15.98.2.8 GetBufferInfoBool8Type()

```
virtual bool GetBufferInfoBool8Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.9 GetBufferInfoPtrType()

```
virtual void* GetBufferInfoPtrType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.10 GetBufferInfoSizeType()

```
virtual size_t GetBufferInfoSizeType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.11 GetBufferInfoUInt64Type()

```
virtual uint64_t GetBufferInfoUInt64Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.12 GetDeviceNodeMap()

```
virtual GenApi::INodeMap* GetDeviceNodeMap ( ) const [pure virtual]
```

15.98.2.13 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout ) [pure virtual]
```

15.98.2.14 GetNextImageInternal()

```
virtual ImagePtr GetNextImageInternal (
    void ** ppPrivate,
    uint64_t grabTimeout ) [pure virtual]
```

15.98.2.15 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

15.98.2.16 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) const [pure virtual]
```


15.98.2.17 GetPort()

```
virtual Port& GetPort ( ) const [pure virtual]
```

15.98.2.18 GetStreamID()

```
virtual std::string GetStreamID ( ) const [pure virtual]
```

15.98.2.19 GetStreamIndex()

```
virtual uint64_t GetStreamIndex ( ) const [pure virtual]
```

15.98.2.20 GetStreamInfoBool8Type()

```
virtual bool GetStreamInfoBool8Type (
    GenTL::STREAM_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.21 GetStreamInfoSizeType()

```
virtual size_t GetStreamInfoSizeType (
    GenTL::STREAM_INFO_CMD iInfoCmd ) [pure virtual]
```

15.98.2.22 GetStreamType()

```
virtual StreamTypeEnum GetStreamType ( ) const [pure virtual]
```

15.98.2.23 GetSystemBuffer()

```
virtual std::shared_ptr<unsigned char> GetSystemBuffer (
    GenTL::BUFFER_HANDLE hBuffer ) [pure virtual]
```

15.98.2.24 InitChunkAdapter()

```
virtual void InitChunkAdapter (
    GenApi::INodeMap & nodemap ) [pure virtual]
```

15.98.2.25 IsAutoPollingEventHandlerRegistered()

```
virtual bool IsAutoPollingEventHandlerRegistered ( ) [pure virtual]
```

15.98.2.26 IsCRCCheckEnabled()

```
virtual bool IsCRCCheckEnabled ( ) const [pure virtual]
```

15.98.2.27 IsEventHandlerRegistered()

```
virtual bool IsEventHandlerRegistered (
    IImageEventHandler & imageEventHandler ) [pure virtual]
```

15.98.2.28 IsImageBufferInUse()

```
virtual bool IsImageBufferInUse (
    const BUFFER_HANDLE bufferHandle ) [pure virtual]
```

15.98.2.29 IsStreaming()

```
virtual bool IsStreaming ( ) [pure virtual]
```

15.98.2.30 KillBufferEvent()

```
virtual void KillBufferEvent ( ) [pure virtual]
```

15.98.2.31 RegisterImageEventHandler()

```
virtual void RegisterImageEventHandler (
    IImageEventHandler & imageEventHandler,
    EventPollingOptions pollingOption ) [pure virtual]
```

15.98.2.32 ReleaseImageBuffer()

```
virtual void ReleaseImageBuffer (
    const BUFFER_HANDLE bufferHandle ) [pure virtual]
```

15.98.2.33 ResetStreamHandle()

```
virtual void ResetStreamHandle (
    std::shared_ptr< Spinnaker::IGenTLDevice > pDevice ) [pure virtual]
```

15.98.2.34 RevokeImages()

```
virtual void RevokeImages ( ) [pure virtual]
```

15.98.2.35 StartStream()

```
virtual void StartStream (
    const unsigned int stream_index = 0 ) [pure virtual]
```

15.98.2.36 StopStream()

```
virtual void StopStream ( ) [pure virtual]
```

15.98.2.37 TransportLayerStreamInfo()

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const [pure virtual]
```

15.98.2.38 UnregisterImageEventHandler()

```
virtual void UnregisterImageEventHandler (
    IImageEventHandler & imageEventHandler ) [pure virtual]
```

15.98.2.39 WaitOnImageEvent()

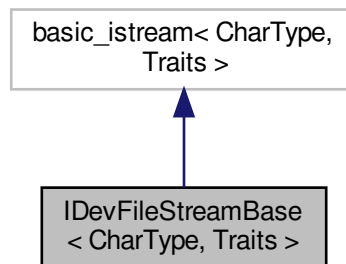
```
virtual void WaitOnImageEvent (
    uint64_t timeout ) [pure virtual]
```

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

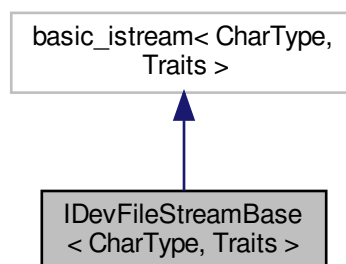
- include/Interface/IStream.h

15.99 IDevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBase< CharType, Traits >:



Collaboration diagram for IDevFileStreamBase< CharType, Traits >:



Public Types

- typedef IDevFileStreamBuf< CharType, Traits > [filebuf_type](#)
- typedef std::basic_ios< CharType, Traits > [ios_type](#)
- typedef std::basic_istream< CharType, Traits > [istream_type](#)

Public Member Functions

- [filebuf_type](#) * [rdbuf](#) () const
- bool [is_open](#) () const
- void [open](#) (Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in)
Open file on device in write mode.
- void [close](#) ()
Close the file on the device.

15.99.1 Member Typedef Documentation

15.99.1.1 filebuf_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf\_type
```

15.99.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios\_type
```

15.99.1.3 istream_type

```
typedef std::basic_istream<CharType, Traits> istream\_type
```

15.99.2 Member Function Documentation

15.99.2.1 close()

```
void close ( ) [inline]
```

Close the file on the device.

15.99.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.99.2.3 open()

```
void open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

Open file on device in write mode.

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached
<i>pFileName</i>	Name of the file to open
<i>mode</i>	open mode

15.99.2.4 rdbuf()

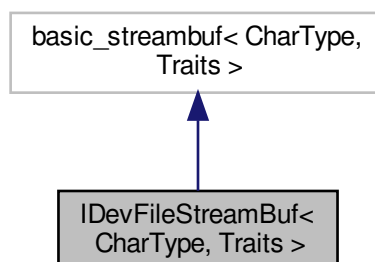
```
filebuf_type* rdbuf ( ) const [inline]
```

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

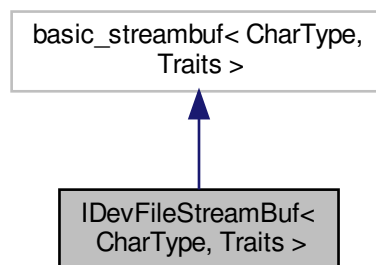
- include/SpinGenApi/[Filestream.h](#)

15.100 IDevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for IDevFileStreamBuf< CharType, Traits >:



Collaboration diagram for IDevFileStreamBuf< CharType, Traits >:



Public Member Functions

- `IDevFileStreamBuf ()`
- `~IDevFileStreamBuf ()`
- `filebuf_type * open (Spinnaker::GenApi::INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::in)`
- `bool is_open () const`
- `filebuf_type * close ()`

Protected Member Functions

- `int_type underflow ()`
- `int_type pbackfail (int_type c)`

15.100.1 Constructor & Destructor Documentation

15.100.1.1 IDevFileStreamBuf()

```
IDevFileStreamBuf ( ) [inline]
```

15.100.1.2 ~IDevFileStreamBuf()

```
~IDevFileStreamBuf ( ) [inline]
```

15.100.2 Member Function Documentation

15.100.2.1 close()

```
filebuf_type* close ( ) [inline]
```

15.100.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.100.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

15.100.2.4 pbackfail()

```
int_type pbackfail (
    int_type c ) [inline], [protected]
```

15.100.2.5 underflow()

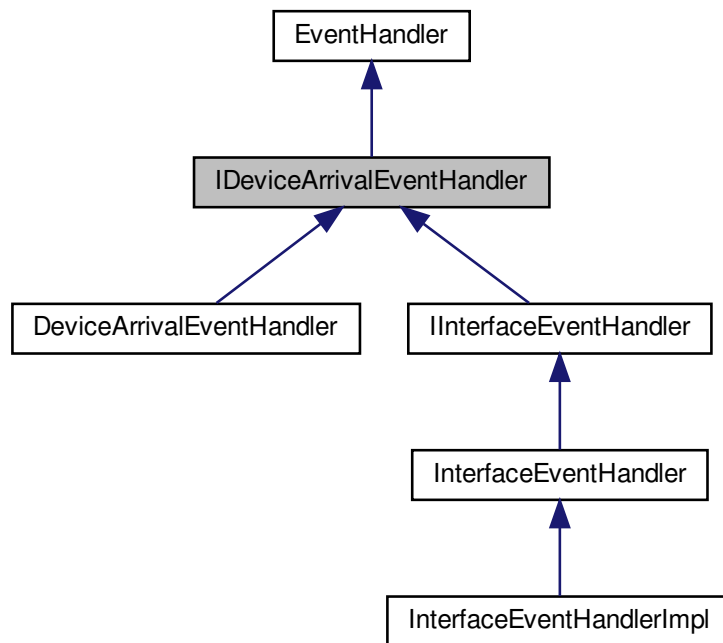
```
int_type underflow ( ) [inline], [protected]
```

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

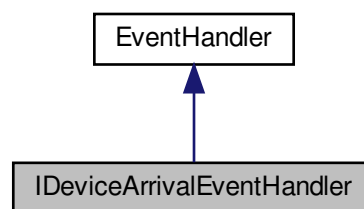
- include/SpinGenApi/[Filestream.h](#)

15.101 IDeviceArrivalEventHandler Class Reference

Inheritance diagram for IDeviceArrivalEventHandler:



Collaboration diagram for IDeviceArrivalEventHandler:



Public Member Functions

- virtual [~IDeviceArrivalEventHandler](#) ()
- virtual void [OnDeviceArrival](#) ([CameraPtr](#) pCamera)=0

Protected Member Functions

- [IDeviceArrivalEventHandler](#) ()
- [IDeviceArrivalEventHandler](#) (const [IDeviceArrivalEventHandler](#) &)
- [IDeviceArrivalEventHandler](#) & [operator=](#) (const [IDeviceArrivalEventHandler](#) &)

Additional Inherited Members

15.101.1 Constructor & Destructor Documentation

15.101.1.1 ~IDeviceArrivalEventHandler()

```
virtual ~IDeviceArrivalEventHandler ( ) [inline], [virtual]
```

15.101.1.2 IDeviceArrivalEventHandler() [1/2]

```
IDeviceArrivalEventHandler ( ) [inline], [protected]
```

15.101.1.3 IDeviceArrivalEventHandler() [2/2]

```
IDeviceArrivalEventHandler (
    const IDeviceArrivalEventHandler & ) [inline], [protected]
```

15.101.2 Member Function Documentation

15.101.2.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    CameraPtr pCamera ) [pure virtual]
```

Implemented in [DeviceArrivalEventHandler](#), [InterfaceEventHandler](#), [IInterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.101.2.2 operator=()

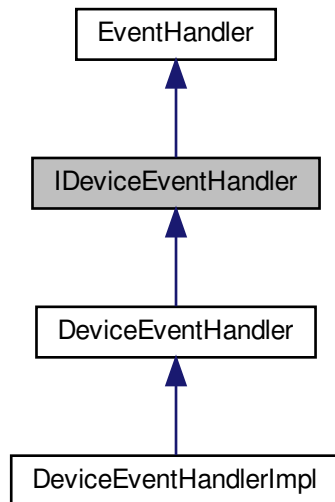
```
IDeviceArrivalEventHandler& operator= (
    const IDeviceArrivalEventHandler & ) [protected]
```

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

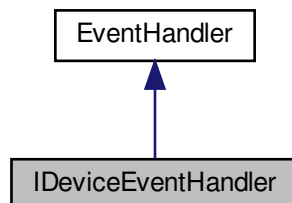
- include/Interface/[IDeviceArrivalEventHandler.h](#)

15.102 IDeviceEventHandler Class Reference

Inheritance diagram for IDeviceEventHandler:



Collaboration diagram for IDeviceEventHandler:



Public Member Functions

- virtual [~IDeviceEventHandler](#) ()
- virtual void [OnDeviceEvent](#) ([Spinnaker::GenICam::gcstring](#) eventName)=0
- virtual uint64_t [GetDeviceEventId](#) () const =0
- virtual [GenICam::gcstring](#) [GetDeviceEventName](#) () const =0

Protected Member Functions

- [IDeviceEventHandler](#) ()
- [IDeviceEventHandler](#) (const [IDeviceEventHandler](#) &)
- [IDeviceEventHandler](#) & [operator=](#) (const [IDeviceEventHandler](#) &)

Additional Inherited Members

15.102.1 Constructor & Destructor Documentation

15.102.1.1 ~IDeviceEventHandler()

[virtual ~IDeviceEventHandler](#) () [inline], [virtual]

15.102.1.2 IDeviceEventHandler() [1/2]

[IDeviceEventHandler](#) () [inline], [protected]

15.102.1.3 IDeviceEventHandler() [2/2]

[IDeviceEventHandler](#) (
 const [IDeviceEventHandler](#) &) [inline], [protected]

15.102.2 Member Function Documentation

15.102.2.1 GetDeviceEventId()

[virtual uint64_t](#) [GetDeviceEventId](#) () const [pure virtual]

Implemented in [DeviceEventHandler](#).

15.102.2.2 GetDeviceEventName()

```
virtual GenICam::gcstring GetDeviceEventName ( ) const [pure virtual]
```

Implemented in [DeviceEventHandler](#).

15.102.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (   
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Implemented in [DeviceEventHandler](#), and [DeviceEventHandlerImpl](#).

15.102.2.4 operator=()

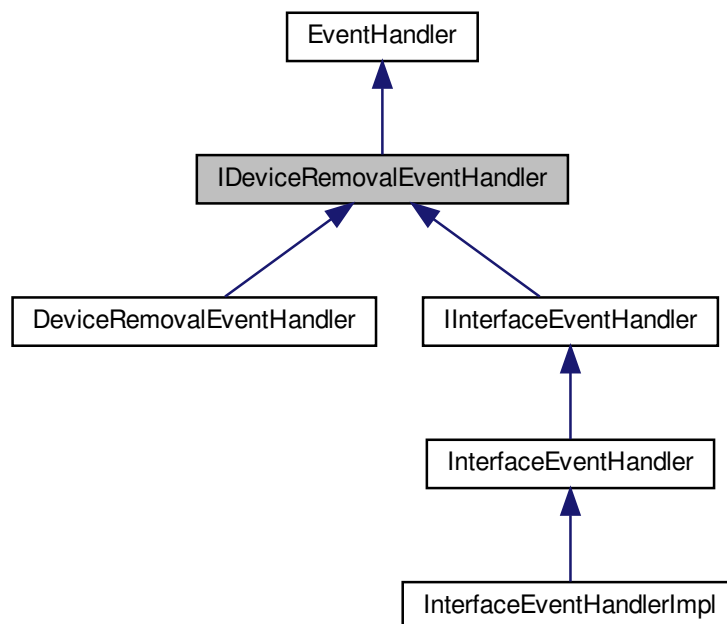
```
IDeviceEventHandler& operator= (   
    const IDeviceEventHandler & ) [protected]
```

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

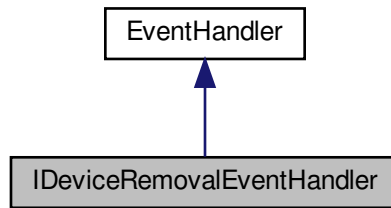
- [include/Interface/IDeviceEventHandler.h](#)

15.103 IDeviceRemovalEventHandler Class Reference

Inheritance diagram for IDeviceRemovalEventHandler:



Collaboration diagram for IDeviceRemovalEventHandler:



Public Member Functions

- virtual [~IDeviceRemovalEventHandler](#) ()
- virtual void [OnDeviceRemoval](#) ([CameraPtr](#) pCamera)=0

Protected Member Functions

- [IDeviceRemovalEventHandler](#) ()
- [IDeviceRemovalEventHandler](#) (const [IDeviceRemovalEventHandler](#) &)
- [IDeviceRemovalEventHandler](#) & [operator=](#) (const [IDeviceRemovalEventHandler](#) &)

Additional Inherited Members

15.103.1 Constructor & Destructor Documentation

15.103.1.1 ~IDeviceRemovalEventHandler()

```
virtual ~IDeviceRemovalEventHandler ( ) [inline], [virtual]
```

15.103.1.2 IDeviceRemovalEventHandler() [1/2]

```
IDeviceRemovalEventHandler ( ) [inline], [protected]
```

15.103.1.3 IDeviceRemovalEventHandler() [2/2]

```
IDeviceRemovalEventHandler (
    const IDeviceRemovalEventHandler & ) [inline], [protected]
```

15.103.2 Member Function Documentation

15.103.2.1 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    CameraPtr pCamera ) [pure virtual]
```

Implemented in [InterfaceEventHandler](#), [DeviceRemovalEventHandler](#), [IInterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.103.2.2 operator=()

```
IDeviceRemovalEventHandler& operator= (
    const IDeviceRemovalEventHandler & ) [protected]
```

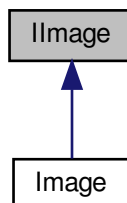
The documentation for this class was generated from the following file:

- [include/Interface/IDeviceRemovalEventHandler.h](#)

15.104 IImage Class Reference

The interface file for [Image](#).

Inheritance diagram for IImage:



Public Member Functions

- virtual [~Image](#) ()
- virtual [ColorProcessingAlgorithm GetColorProcessing](#) () const =0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat)=0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData)=0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData, [TLPayloadType](#) dataPayloadType, size_t dataSize)=0
- virtual void [Release](#) ()=0
- virtual uint64_t [GetID](#) () const =0
- virtual uint64_t [GetStreamIndex](#) () const =0
- virtual void * [GetData](#) () const =0
- virtual void * [GetPrivateData](#) () const =0
- virtual float [GetDataAbsoluteMax](#) () const =0
- virtual float [GetDataAbsoluteMin](#) () const =0
- virtual size_t [GetBufferSize](#) () const =0
- virtual void [DeepCopy](#) (const [ImagePtr](#) pSrcImage)=0
- virtual size_t [GetWidth](#) () const =0
- virtual size_t [GetHeight](#) () const =0
- virtual size_t [GetStride](#) () const =0
- virtual size_t [GetBitsPerPixel](#) () const =0
- virtual size_t [GetNumChannels](#) () const =0
- virtual size_t [GetXOffset](#) () const =0
- virtual size_t [GetYOffset](#) () const =0
- virtual size_t [GetXPadding](#) () const =0
- virtual size_t [GetYPadding](#) () const =0
- virtual uint64_t [GetFrameID](#) () const =0
- virtual size_t [GetPayloadType](#) () const =0
- virtual [TLPayloadType](#) [GetTLPayloadType](#) () const =0
- virtual uint64_t [GetTLPixelFormat](#) () const =0
- virtual [TLPixelFormatNamespace](#) [GetTLPixelFormatNamespace](#) () const =0
- virtual [GenlCam::gcstring](#) [GetPixelFormatName](#) () const =0
- virtual [PixelFormatEnums](#) [GetPixelFormat](#) () const =0
- virtual [PixelFormatIntType](#) [GetPixelFormatIntType](#) () const =0
- virtual bool [IsIncomplete](#) () const =0
- virtual size_t [GetValidPayloadSize](#) () const =0
- virtual uint64_t [GetChunkLayoutId](#) () const =0
- virtual uint64_t [GetTimeStamp](#) () const =0
- virtual void [Save](#) (const char *pFilename, [ImageFileFormat](#) format=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) const =0
- virtual void [Save](#) (const char *pFilename, [PNGOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [PPMOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [PGMOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [TIFFOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [JPEGOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [JPG2Option](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [BMPOption](#) &pOption) const =0
- virtual void [Save](#) (const char *pFilename, [SIOption](#) &pOption) const =0
- virtual const [ChunkData](#) & [GetChunkData](#) () const =0
- virtual void [CalculateStatistics](#) ([ImageStatistics](#) &pStatistics)=0
- virtual bool [HasCRC](#) () const =0
- virtual bool [HasChunkData](#) () const =0
- virtual bool [CheckCRC](#) () const =0
- virtual size_t [GetImageSize](#) () const =0
- virtual bool [IsInUse](#) ()=0
- virtual [ImageStatus](#) [GetImageStatus](#) () const =0
- virtual bool [IsCompressed](#) () const =0

Protected Member Functions

- [IImage](#) ()
- virtual ImageData * [GetImageData](#) () const =0

Friends

- class [Stream](#)
- class [ImageImpl](#)
- class [ImageListImpl](#)

15.104.1 Detailed Description

The interface file for [Image](#).

15.104.2 Constructor & Destructor Documentation

15.104.2.1 ~IImage()

```
virtual ~IImage ( ) [inline], [virtual]
```

15.104.2.2 IImage()

```
IImage ( ) [inline], [protected]
```

15.104.3 Member Function Documentation

15.104.3.1 CalculateStatistics()

```
virtual void CalculateStatistics (
    ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.2 CheckCRC()

```
virtual bool CheckCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.3 DeepCopy()

```
virtual void DeepCopy (
    const ImagePtr pSrcImage ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.4 GetBitsPerPixel()

```
virtual size_t GetBitsPerPixel ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.5 GetBufferSize()

```
virtual size_t GetBufferSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.6 GetChunkData()

```
virtual const ChunkData& GetChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.7 GetChunkLayoutId()

```
virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.8 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.9 GetData()

```
virtual void* GetData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.10 GetDataAbsoluteMax()

```
virtual float GetDataAbsoluteMax ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.11 GetDataAbsoluteMin()

```
virtual float GetDataAbsoluteMin ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.12 GetFrameID()

```
virtual uint64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.13 GetHeight()

```
virtual size_t GetHeight ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.14 GetID()

```
virtual uint64_t GetID ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.15 GetImageData()

```
virtual ImageData* GetImageData ( ) const [protected], [pure virtual]
```

Implemented in [Image](#).

15.104.3.16 GetImageSize()

```
virtual size_t GetImageSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.17 GetImageStatus()

```
virtual ImageStatus GetImageStatus ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.18 GetNumChannels()

```
virtual size_t GetNumChannels ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.19 GetPayloadType()

```
virtual size_t GetPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.20 GetPixelFormat()

```
virtual PixelFormatEnums GetPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.21 GetPixelFormatIntType()

```
virtual PixelFormatIntType GetPixelFormatIntType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.22 GetPixelFormatName()

```
virtual GenICam::gcstring GetPixelFormatName ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.23 GetPrivateData()

```
virtual void* GetPrivateData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.24 GetStreamIndex()

```
virtual uint64_t GetStreamIndex ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.25 GetStride()

```
virtual size_t GetStride ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.26 GetTimeStamp()

```
virtual uint64_t GetTimeStamp ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.27 GetTLPayloadType()

```
virtual TLPayloadType GetTLPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.28 GetTLPixelFormat()

```
virtual uint64_t GetTLPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.29 GetTLPixelFormatNamespace()

```
virtual TLPixelFormatNamespace GetTLPixelFormatNamespace ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.30 GetValidPayloadSize()

```
virtual size_t GetValidPayloadSize ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.31 GetWidth()

```
virtual size_t GetWidth ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.32 GetXOffset()

```
virtual size_t GetXOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.33 GetXPadding()

```
virtual size_t GetXPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.34 GetYOffset()

```
virtual size_t GetYOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.35 GetYPadding()

```
virtual size_t GetYPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.36 HasChunkData()

```
virtual bool HasChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.37 HasCRC()

```
virtual bool HasCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.38 IsCompressed()

```
virtual bool IsCompressed ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.39 IsIncomplete()

```
virtual bool IsIncomplete ( ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.40 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.41 Release()

```
virtual void Release ( ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.42 ResetImage() [1/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.43 ResetImage() [2/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.44 ResetImage() [3/3]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    TLPayloadType dataPayloadType,
    size_t dataSize ) [pure virtual]
```

Implemented in [Image](#).

15.104.3.45 Save() [1/9]

```
virtual void Save (
    const char * pFilename,
    BMPOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.46 Save() [2/9]

```
virtual void Save (
    const char * pFilename,
    ImageFileFormat format = SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT ) const [pure
virtual]
```

Implemented in [Image](#).

15.104.3.47 Save() [3/9]

```
virtual void Save (  
    const char * pFilename,  
    JPEGOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.48 Save() [4/9]

```
virtual void Save (  
    const char * pFilename,  
    JPG2Option & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.49 Save() [5/9]

```
virtual void Save (  
    const char * pFilename,  
    PGMOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.50 Save() [6/9]

```
virtual void Save (  
    const char * pFilename,  
    PNGOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.51 Save() [7/9]

```
virtual void Save (  
    const char * pFilename,  
    PPMOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.52 Save() [8/9]

```
virtual void Save (  
    const char * pFilename,  
    SIOption & option ) const [pure virtual]
```

Implemented in [Image](#).

15.104.3.53 Save() [9/9]

```
virtual void Save (  
    const char * pFilename,  
    TIFFOption & pOption ) const [pure virtual]
```

Implemented in [Image](#).

15.104.4 Friends And Related Function Documentation**15.104.4.1 ImageImpl**

```
friend class ImageImpl [friend]
```

15.104.4.2 ImageListImpl

```
friend class ImageListImpl [friend]
```

15.104.4.3 Stream

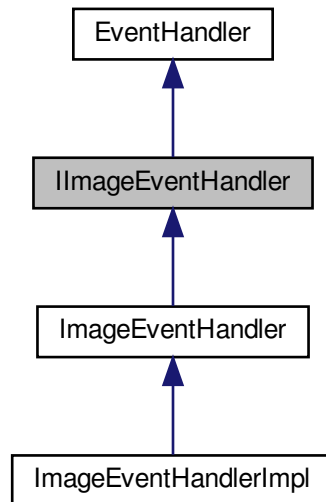
```
friend class Stream [friend]
```

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

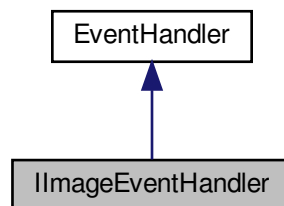
- [include/Interface/IImage.h](#)

15.105 IImageEventHandler Class Reference

Inheritance diagram for IImageEventHandler:



Collaboration diagram for IImageEventHandler:



Public Member Functions

- virtual [~IImageEventHandler](#) ()

Protected Member Functions

- [IImageEventHandler](#) ()
- [IImageEventHandler](#) (const [IImageEventHandler](#) &)
- [IImageEventHandler](#) & [operator=](#) (const [IImageEventHandler](#) &)
- virtual void [OnImageEvent](#) ([ImagePtr](#) image)=0

Friends

- class [Stream](#)

Additional Inherited Members

15.105.1 Constructor & Destructor Documentation

15.105.1.1 ~IImageEventHandler()

```
virtual ~IImageEventHandler ( ) [inline], [virtual]
```

15.105.1.2 IImageEventHandler() [1/2]

```
IImageEventHandler ( ) [inline], [protected]
```

15.105.1.3 IImageEventHandler() [2/2]

```
IImageEventHandler (
    const IImageEventHandler & ) [inline], [protected]
```

15.105.2 Member Function Documentation

15.105.2.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [protected], [pure virtual]
```

Implemented in [ImageEventHandler](#), [ImageEventHandlerImpl](#), and [ImageEventHandlerImpl](#).

15.105.2.2 operator=()

```
IImageEventHandler& operator= (
    const IImageEventHandler & ) [protected]
```

15.105.3 Friends And Related Function Documentation

15.105.3.1 Stream

```
friend class Stream [friend]
```

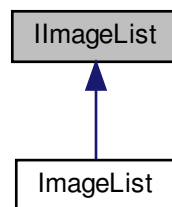
The documentation for this class was generated from the following file:

- include/Interface/[IImageEventHandler.h](#)

15.106 IImageList Class Reference

Used to hold a list of image objects.

Inheritance diagram for IImageList:



Public Member Functions

- virtual [~IImageList](#) (void)
- virtual [ImagePtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [ImagePtr GetByIndex](#) (unsigned int index) const =0
- virtual [ImagePtr GetByPixelFormat](#) ([PixelFormatEnums](#) pixelFormat) const =0
- virtual void [Clear](#) ()=0
- virtual void [RemoveByIndex](#) (unsigned int index)=0
- virtual void [RemoveByPixelFormat](#) ([PixelFormatEnums](#) pixelFormat)=0
- virtual void [Append](#) (const [ImageList](#) &list)=0
- virtual void [Add](#) ([ImagePtr](#) image)=0
- virtual void [Release](#) ()=0
- virtual void [Save](#) (const char *filename)=0

Protected Member Functions

- [IImageList](#) ()
- [IImageList](#) (const [IImageList](#) &)
- [IImageList](#) & [operator=](#) (const [IImageList](#) &)

Protected Attributes

- [ImageListData](#) * [m_pImageListData](#)

Friends

- class [ImageListImpl](#)

15.106.1 Detailed Description

Used to hold a list of image objects.

15.106.2 Constructor & Destructor Documentation

15.106.2.1 ~IImageList()

```
virtual ~IImageList (  
    void ) [inline], [virtual]
```

15.106.2.2 IImageList() [1/2]

```
IImageList ( ) [inline], [protected]
```

15.106.2.3 IImageList() [2/2]

```
IImageList (  
    const IImageList & ) [inline], [protected]
```

15.106.3 Member Function Documentation

15.106.3.1 Add()

```
virtual void Add (
    ImagePtr image ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.2 Append()

```
virtual void Append (
    const ImageList & list ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.3 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.4 GetByIndex()

```
virtual ImagePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.5 GetByPixelFormat()

```
virtual ImagePtr GetByPixelFormat (
    PixelFormatEnums pixelFormat ) const [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.6 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.7 operator=()

```
IImageList& operator= (
    const IImageList & ) [protected]
```

15.106.3.8 operator[]()

```
virtual ImagePtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.9 Release()

```
virtual void Release ( ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.10 RemoveByIndex()

```
virtual void RemoveByIndex (
    unsigned int index ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.11 RemoveByPixelFormat()

```
virtual void RemoveByPixelFormat (
    PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.3.12 Save()

```
virtual void Save (
    const char * filename ) [pure virtual]
```

Implemented in [ImageList](#).

15.106.4 Friends And Related Function Documentation

15.106.4.1 ImageListImpl

```
friend class ImageListImpl [friend]
```

15.106.5 Member Data Documentation

15.106.5.1 m_pImageListData

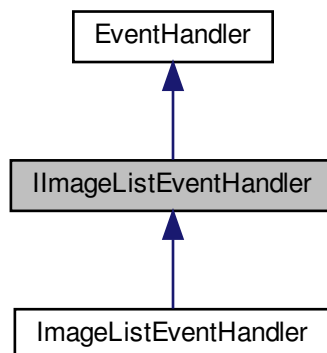
```
ImageListData* m_pImageListData [protected]
```

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

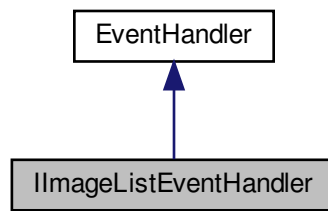
- include/Interface/[IImageList.h](#)

15.107 IImageListEventHandler Class Reference

Inheritance diagram for IImageListEventHandler:



Collaboration diagram for IImageListEventHandler:



Public Member Functions

- virtual [~IImageListEventHandler](#) ()

Protected Member Functions

- [IImageListEventHandler](#) ()
- [IImageListEventHandler](#) (const [IImageListEventHandler](#) &)
- [IImageListEventHandler](#) & [operator=](#) (const [IImageListEventHandler](#) &)
- virtual void [OnImageListEvent](#) ([ImageList](#) imageList)=0

Friends

- class [EventCallbackInfo](#)

Additional Inherited Members

15.107.1 Constructor & Destructor Documentation

15.107.1.1 ~IImageListEventHandler()

```
virtual ~IImageListEventHandler ( ) [inline], [virtual]
```

15.107.1.2 IImageListEventHandler() [1/2]

```
IImageListEventHandler ( ) [inline], [protected]
```

15.107.1.3 `IImageListEventHandler()` [2/2]

```
IImageListEventHandler (
    const IImageListEventHandler & ) [inline], [protected]
```

15.107.2 Member Function Documentation

15.107.2.1 `OnImageListEvent()`

```
virtual void OnImageListEvent (
    ImageList imageList ) [protected], [pure virtual]
```

Implemented in [ImageListEventHandler](#).

15.107.2.2 `operator=()`

```
IImageListEventHandler& operator= (
    const IImageListEventHandler & ) [protected]
```

15.107.3 Friends And Related Function Documentation

15.107.3.1 `EventCallbackInfo`

```
friend class EventCallbackInfo [friend]
```

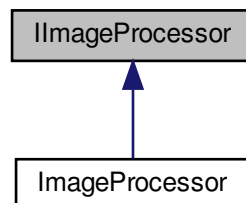
The documentation for this class was generated from the following file:

- [include/Interface/IImageListEventHandler.h](#)

15.108 `IImageProcessor` Class Reference

Used to hold a list of image objects.

Inheritance diagram for `IImageProcessor`:



Public Member Functions

- virtual [~IImageProcessor](#) (void)
- virtual void [SetColorProcessing](#) ([ColorProcessingAlgorithm](#) colorAlgorithm)=0
- virtual [ColorProcessingAlgorithm](#) [GetColorProcessing](#) ()=0
- virtual void [SetNumDecompressionThreads](#) (unsigned int numThreads)=0
- virtual unsigned int [GetNumDecompressionThreads](#) ()=0
- virtual [ImagePtr](#) [Convert](#) (const [ImagePtr](#) &srcImage, [PixelFormatEnums](#) destFormat) const =0
- virtual void [Convert](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [PixelFormatEnums](#) destFormat) const =0
- virtual [ImagePtr](#) [Convert](#) (const [ImageList](#) &srcImageList, [PixelFormatEnums](#) destFormat) const =0
- virtual void [Convert](#) (const [ImageList](#) &srcImageList, [ImagePtr](#) &destImage, [PixelFormatEnums](#) destFormat) const =0
- virtual [ImagePtr](#) [ApplyGamma](#) (const [ImagePtr](#) &srcImage, float gamma, bool isGammaInverse=false) const =0
- virtual void [ApplyGamma](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, float gamma, bool isGammaInverse=false) const =0

Protected Member Functions

- [IImageProcessor](#) ()
- [IImageProcessor](#) (const [IImageProcessor](#) &)
- [IImageProcessor](#) & [operator=](#) (const [IImageProcessor](#) &)

Protected Attributes

- ImageProcessorData * [m_pImageProcessorData](#)

15.108.1 Detailed Description

Used to hold a list of image objects.

15.108.2 Constructor & Destructor Documentation

15.108.2.1 ~IImageProcessor()

```
virtual ~IImageProcessor (
    void ) [inline], [virtual]
```

15.108.2.2 IImageProcessor() [1/2]

```
IImageProcessor ( ) [inline], [protected]
```

15.108.2.3 `IImageProcessor()` [2/2]

```
IImageProcessor (
    const IImageProcessor & ) [inline], [protected]
```

15.108.3 Member Function Documentation

15.108.3.1 `ApplyGamma()` [1/2]

```
virtual ImagePtr ApplyGamma (
    const ImagePtr & srcImage,
    float gamma,
    bool isGammaInverse = false ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.2 `ApplyGamma()` [2/2]

```
virtual void ApplyGamma (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    float gamma,
    bool isGammaInverse = false ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.3 `Convert()` [1/4]

```
virtual void Convert (
    const ImageList & srcImageList,
    ImagePtr & destImage,
    PixelFormatEnums destFormat ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.4 `Convert()` [2/4]

```
virtual ImagePtr Convert (
    const ImageList & srcImageList,
    PixelFormatEnums destFormat ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.5 Convert() [3/4]

```
virtual void Convert (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    PixelFormatEnums destFormat ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.6 Convert() [4/4]

```
virtual ImagePtr Convert (
    const ImagePtr & srcImage,
    PixelFormatEnums destFormat ) const [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.7 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.8 GetNumDecompressionThreads()

```
virtual unsigned int GetNumDecompressionThreads ( ) [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.9 operator=()

```
IImageProcessor& operator= (
    const IImageProcessor & ) [protected]
```

15.108.3.10 SetColorProcessing()

```
virtual void SetColorProcessing (
    ColorProcessingAlgorithm colorAlgorithm ) [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.3.11 SetNumDecompressionThreads()

```
virtual void SetNumDecompressionThreads (
    unsigned int numThreads ) [pure virtual]
```

Implemented in [ImageProcessor](#).

15.108.4 Member Data Documentation

15.108.4.1 m_pImageProcessorData

```
ImageProcessorData* m_pImageProcessorData [protected]
```

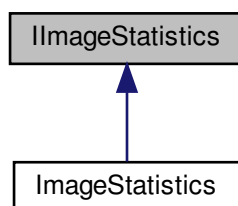
The documentation for this class was generated from the following file:

- include/Interface/[IImageProcessor.h](#)

15.109 IImageStatistics Class Reference

The interface file for image statistics.

Inheritance diagram for IImageStatistics:



Public Member Functions

- virtual [~IImageStatistics](#) ()
- virtual void [EnableAll](#) ()=0
- virtual void [DisableAll](#) ()=0
- virtual void [EnableGreyOnly](#) ()=0
- virtual void [EnableRGBOnly](#) ()=0
- virtual void [EnableHSLOnly](#) ()=0
- virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool *pEnabled) const =0
- virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)=0
- virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int *pMin, unsigned int *pMax) const =0
- virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const =0
- virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int *pNumPixelValues) const =0
- virtual void [GetMean](#) ([StatisticsChannel](#) channel, float *pPixelValueMean) const =0
- virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int **ppHistogram) const =0
- virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int *pRangeMin=NULL, unsigned int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *pPixelValueMean=NULL, int **ppHistogram=NULL) const =0

Protected Member Functions

- [IImageStatistics](#) ()
- [IImageStatistics](#) (const [IImageStatistics](#) &)

15.109.1 Detailed Description

The interface file for image statistics.

15.109.2 Constructor & Destructor Documentation

15.109.2.1 ~IImageStatistics()

```
virtual ~IImageStatistics ( ) [inline], [virtual]
```

15.109.2.2 IImageStatistics() [1/2]

```
IImageStatistics ( ) [inline], [protected]
```

15.109.2.3 IImageStatistics() [2/2]

```
IImageStatistics (
    const IImageStatistics & ) [inline], [protected]
```

15.109.3 Member Function Documentation

15.109.3.1 DisableAll()

```
virtual void DisableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.2 EnableAll()

```
virtual void EnableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.11 GetRange()

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.12 GetStatistics()

```
virtual void GetStatistics (
    StatisticsChannel channel,
    unsigned int * pRangeMin = NULL,
    unsigned int * pRangeMax = NULL,
    unsigned int * pPixelValueMin = NULL,
    unsigned int * pPixelValueMax = NULL,
    unsigned int * pNumPixelValues = NULL,
    float * pPixelValueMean = NULL,
    int ** ppHistogram = NULL ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

15.109.3.13 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [pure virtual]
```

Implemented in [ImageStatistics](#).

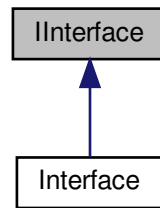
The documentation for this class was generated from the following file:

- [include/Interface/IImageStatistics.h](#)

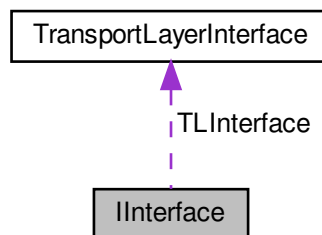
15.110 Interface Class Reference

The interface file for [Interface](#).

Inheritance diagram for IInterface:



Collaboration diagram for IInterface:



Public Member Functions

- virtual [~IInterface](#) ()
- virtual [CameraList GetCameras](#) (bool updateCameras=true) const =0
- virtual bool [UpdateCameras](#) ()=0
- virtual [GenApi::INodeMap & GetTLNodeMap](#) () const =0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual bool [IsInUse](#) () const =0
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL) const =0
- virtual bool [IsValid](#) ()=0

Public Attributes

- [TransportLayerInterface TLInterface](#)

Protected Member Functions

- [IInterface](#) ()
- [IInterface](#) (const [IInterface](#) &)
- [IInterface](#) & [operator=](#) (const [IInterface](#) &)

Protected Attributes

- InterfaceData * [m_pInterfaceData](#)

Friends

- class [InterfaceInternal](#)
- class [ProducerImpl](#)

15.110.1 Detailed Description

The interface file for [Interface](#).

15.110.2 Constructor & Destructor Documentation

15.110.2.1 ~IInterface()

```
virtual ~IInterface ( ) [inline], [virtual]
```

15.110.2.2 IInterface() [1/2]

```
IInterface ( ) [inline], [protected]
```

15.110.2.3 IInterface() [2/2]

```
IInterface (
    const IInterface & ) [inline], [protected]
```

15.110.3 Member Function Documentation

15.110.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateCameras = true ) const [pure virtual]
```

Implemented in [Interface](#).

15.110.3.2 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [Interface](#).

15.110.3.3 IsInUse()

```
virtual bool IsInUse ( ) const [pure virtual]
```

Implemented in [Interface](#).

15.110.3.4 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [Interface](#).

15.110.3.5 operator=()

```
IInterface& operator= (
    const IInterface & ) [protected]
```

15.110.3.6 RegisterEventHandler()

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [Interface](#).

15.110.3.7 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) const [pure virtual]
```

Implemented in [Interface](#).

15.110.3.8 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [Interface](#).

15.110.3.9 UpdateCameras()

```
virtual bool UpdateCameras ( ) [pure virtual]
```

Implemented in [Interface](#).

15.110.4 Friends And Related Function Documentation

15.110.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

15.110.4.2 ProducerImpl

```
friend class ProducerImpl [friend]
```

15.110.5 Member Data Documentation

15.110.5.1 m_pInterfaceData

```
InterfaceData* m_pInterfaceData [protected]
```

15.110.5.2 TLInterface

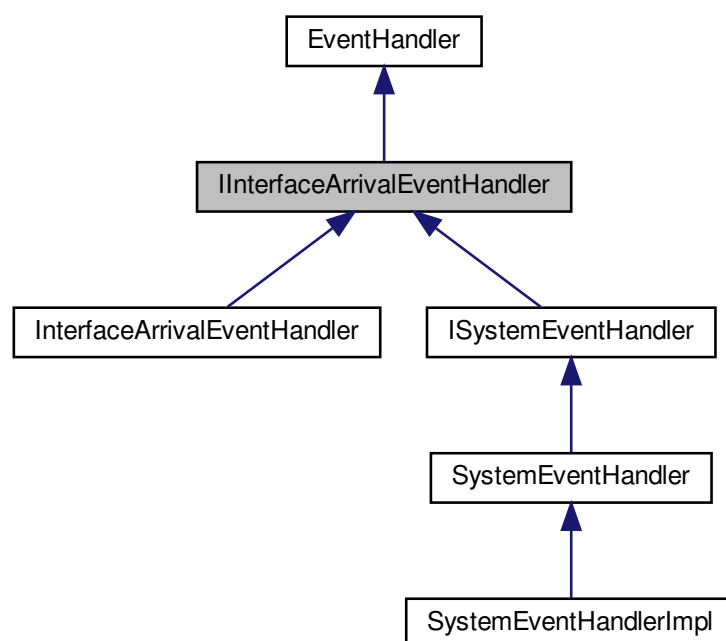
```
TransportLayerInterface TLInterface
```

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

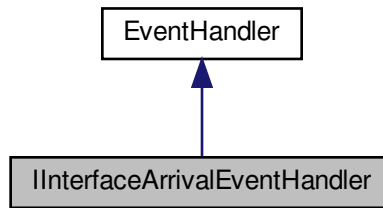
- include/Interface/IInterface.h

15.111 IInterfaceArrivalEventHandler Class Reference

Inheritance diagram for IInterfaceArrivalEventHandler:



Collaboration diagram for `IInterfaceArrivalEventHandler`:



Public Member Functions

- virtual `~IInterfaceArrivalEventHandler()`
- virtual void `OnInterfaceArrival (InterfacePtr pInterface)=0`

Protected Member Functions

- `IInterfaceArrivalEventHandler()`
- `IInterfaceArrivalEventHandler (const IInterfaceArrivalEventHandler &)`
- `IInterfaceArrivalEventHandler & operator= (const IInterfaceArrivalEventHandler &)`

Additional Inherited Members

15.111.1 Constructor & Destructor Documentation

15.111.1.1 `~IInterfaceArrivalEventHandler()`

```
virtual ~IInterfaceArrivalEventHandler ( ) [inline], [virtual]
```

15.111.1.2 `IInterfaceArrivalEventHandler()` [1/2]

```
IInterfaceArrivalEventHandler ( ) [inline], [protected]
```

15.111.1.3 IInterfaceArrivalEventHandler() [2/2]

```
IInterfaceArrivalEventHandler (
    const IInterfaceArrivalEventHandler & ) [inline], [protected]
```

15.111.2 Member Function Documentation

15.111.2.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    InterfacePtr pInterface ) [pure virtual]
```

Implemented in [InterfaceArrivalEventHandler](#), [SystemEventHandler](#), [ISystemEventHandler](#), and [SystemEventHandlerImpl](#).

15.111.2.2 operator=()

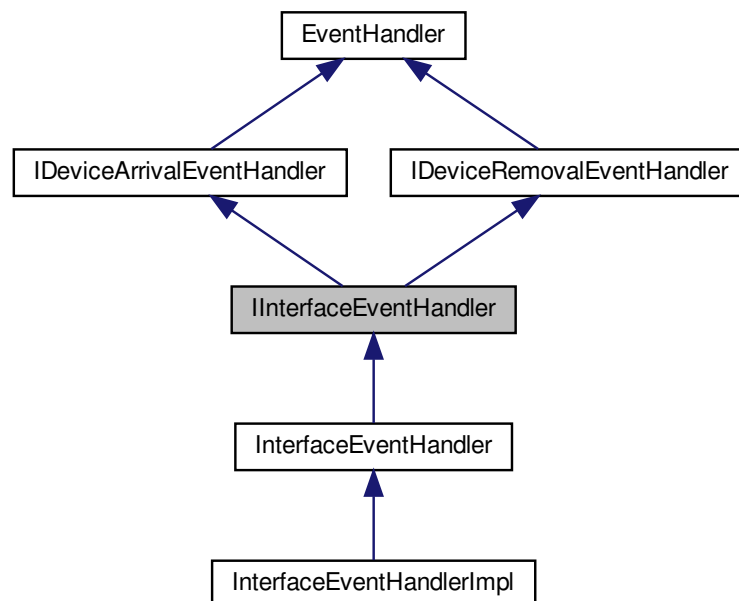
```
IInterfaceArrivalEventHandler& operator= (
    const IInterfaceArrivalEventHandler & ) [protected]
```

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

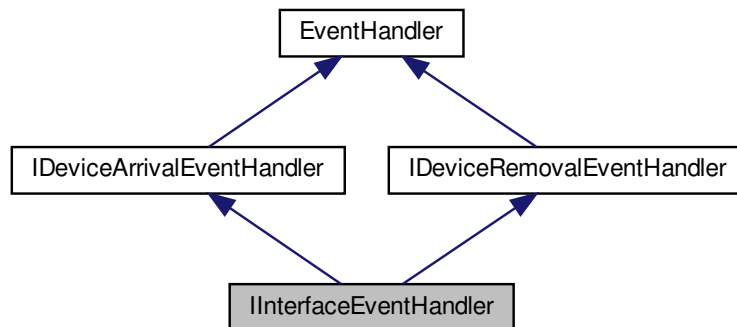
- include/Interface/[IInterfaceArrivalEventHandler.h](#)

15.112 IInterfaceEventHandler Class Reference

Inheritance diagram for IInterfaceEventHandler:



Collaboration diagram for `IInterfaceEventHandler`:



Public Member Functions

- virtual `~IInterfaceEventHandler` ()
- virtual void `OnDeviceArrival` (`CameraPtr` pCamera)=0
- virtual void `OnDeviceRemoval` (`CameraPtr` pCamera)=0

Protected Member Functions

- `IInterfaceEventHandler` ()
- `IInterfaceEventHandler` (const `IInterfaceEventHandler` &)
- `IInterfaceEventHandler` & `operator=` (const `IInterfaceEventHandler` &)

Additional Inherited Members

15.112.1 Constructor & Destructor Documentation

15.112.1.1 `~IInterfaceEventHandler()`

```
virtual ~IInterfaceEventHandler ( ) [inline], [virtual]
```

15.112.1.2 `IInterfaceEventHandler()` [1/2]

```
IInterfaceEventHandler ( ) [inline], [protected]
```

15.112.1.3 IInterfaceEventHandler() [2/2]

```
IInterfaceEventHandler (
    const IInterfaceEventHandler & ) [inline], [protected]
```

15.112.2 Member Function Documentation

15.112.2.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    CameraPtr pCamera ) [pure virtual]
```

Implements [IDeviceArrivalEventHandler](#).

Implemented in [InterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.112.2.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    CameraPtr pCamera ) [pure virtual]
```

Implements [IDeviceRemovalEventHandler](#).

Implemented in [InterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.112.2.3 operator=()

```
IInterfaceEventHandler& operator= (
    const IInterfaceEventHandler & ) [protected]
```

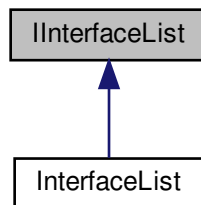
The documentation for this class was generated from the following file:

- [include/Interface/IInterfaceEventHandler.h](#)

15.113 IInterfaceList Class Reference

The interface file for [InterfaceList](#) class.

Inheritance diagram for IInterfaceList:



Public Member Functions

- virtual [~IInterfaceList](#) (void)
- virtual [InterfacePtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [InterfacePtr GetByIndex](#) (unsigned int index) const =0
- virtual void [Clear](#) ()=0
- virtual void [Add](#) ([InterfacePtr](#) iface)=0
- virtual void [Remove](#) ([InterfacePtr](#) iface)=0
- virtual void [Append](#) (const [InterfaceList](#) *list)=0

Protected Member Functions

- [IInterfaceList](#) (void)
- [IInterfaceList](#) (const [IInterfaceList](#) &)
- [IInterfaceList](#) & [operator=](#) (const [IInterfaceList](#) &)

Protected Attributes

- [InterfaceListData](#) * [m_pInterfaceListData](#)

Friends

- class [InterfaceListImpl](#)

15.113.1 Detailed Description

The interface file for [InterfaceList](#) class.

15.113.2 Constructor & Destructor Documentation

15.113.2.1 ~IInterfaceList()

```
virtual ~IInterfaceList (
    void ) [inline], [virtual]
```

15.113.2.2 IInterfaceList() [1/2]

```
IInterfaceList (
    void ) [inline], [protected]
```

15.113.2.3 IInterfaceList() [2/2]

```
IInterfaceList (
    const IInterfaceList & ) [inline], [protected]
```

15.113.3 Member Function Documentation

15.113.3.1 Add()

```
virtual void Add (
    InterfacePtr iface ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.2 Append()

```
virtual void Append (
    const InterfaceList * list ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.3 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.4 GetByIndex()

```
virtual InterfacePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.5 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.6 operator=()

```
IInterfaceList& operator= (
    const IInterfaceList & ) [protected]
```

15.113.3.7 operator[]()

```
virtual InterfacePtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.3.8 Remove()

```
virtual void Remove (
    InterfacePtr iface ) [pure virtual]
```

Implemented in [InterfaceList](#).

15.113.4 Friends And Related Function Documentation

15.113.4.1 InterfaceListImpl

```
friend class InterfaceListImpl [friend]
```

15.113.5 Member Data Documentation

15.113.5.1 m_pInterfaceListData

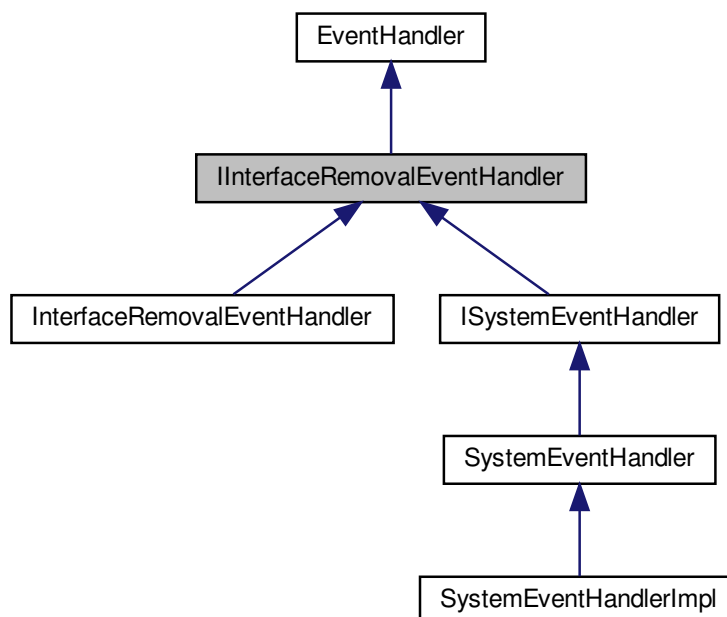
```
InterfaceListData* m_pInterfaceListData [protected]
```

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

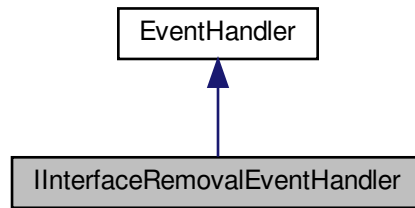
- [include/Interface/IInterfaceList.h](#)

15.114 IInterfaceRemovalEventHandler Class Reference

Inheritance diagram for IInterfaceRemovalEventHandler:



Collaboration diagram for `IInterfaceRemovalEventHandler`:



Public Member Functions

- virtual `~IInterfaceRemovalEventHandler()`
- virtual void `OnInterfaceRemoval (InterfacePtr pInterface)=0`

Protected Member Functions

- `IInterfaceRemovalEventHandler()`
- `IInterfaceRemovalEventHandler (const IInterfaceRemovalEventHandler &)`
- `IInterfaceRemovalEventHandler & operator= (const IInterfaceRemovalEventHandler &)`

Additional Inherited Members

15.114.1 Constructor & Destructor Documentation

15.114.1.1 `~IInterfaceRemovalEventHandler()`

```
virtual ~IInterfaceRemovalEventHandler ( ) [inline], [virtual]
```

15.114.1.2 `IInterfaceRemovalEventHandler()` [1/2]

```
IInterfaceRemovalEventHandler ( ) [inline], [protected]
```

15.114.1.3 InterfaceRemovalEventHandler() [2/2]

```
IInterfaceRemovalEventHandler (
    const IInterfaceRemovalEventHandler & ) [inline], [protected]
```

15.114.2 Member Function Documentation

15.114.2.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    InterfacePtr pInterface ) [pure virtual]
```

Implemented in [SystemEventHandler](#), [InterfaceRemovalEventHandler](#), [ISystemEventHandler](#), and [SystemEventHandlerImpl](#).

15.114.2.2 operator=()

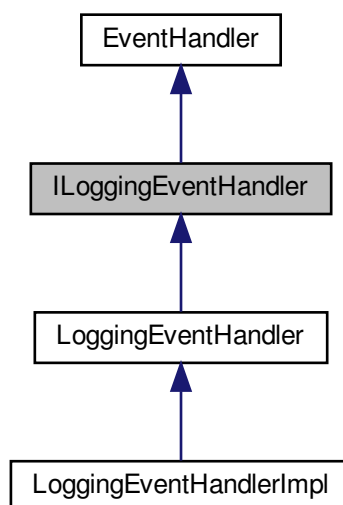
```
IInterfaceRemovalEventHandler& operator= (
    const IInterfaceRemovalEventHandler & ) [protected]
```

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

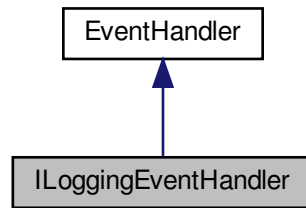
- [include/Interface/IInterfaceRemovalEventHandler.h](#)

15.115 ILoggingEventHandler Class Reference

Inheritance diagram for ILoggingEventHandler:



Collaboration diagram for ILoggingEventHandler:



Public Member Functions

- virtual [~ILoggingEventHandler](#) ()
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0

Protected Member Functions

- [ILoggingEventHandler](#) ()
- [ILoggingEventHandler](#) (const [ILoggingEventHandler](#) &)
- [ILoggingEventHandler](#) & [operator=](#) (const [ILoggingEventHandler](#) &)

Additional Inherited Members

15.115.1 Constructor & Destructor Documentation

15.115.1.1 ~ILoggingEventHandler()

```
virtual ~ILoggingEventHandler ( ) [inline], [virtual]
```

15.115.1.2 ILoggingEventHandler() [1/2]

```
ILoggingEventHandler ( ) [inline], [protected]
```

15.115.1.3 ILoggingEventHandler() [2/2]

```
ILoggingEventHandler (
    const ILoggingEventHandler & ) [inline], [protected]
```

15.115.2 Member Function Documentation

15.115.2.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

Implemented in [LoggingEventHandler](#).

15.115.2.2 operator=()

```
ILoggingEventHandler& operator= (
    const ILoggingEventHandler & ) [protected]
```

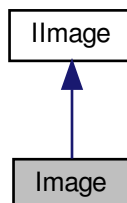
The documentation for this class was generated from the following file:

- [include/Interface/ILoggingEventHandler.h](#)

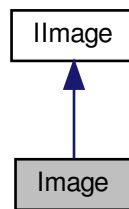
15.116 Image Class Reference

The image object class.

Inheritance diagram for Image:



Collaboration diagram for Image:



Public Member Functions

- virtual [~Image](#) ()
Virtual destructor.
- [ColorProcessingAlgorithm GetColorProcessing](#) () const
Gets the color algorithm used to produce the image.
- void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat)
Sets new dimensions of the image object and allocates memory.
- void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void *pData)
Sets new dimensions of the image object.
- void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData, [TLPayloadType](#) dataPayloadType, size_t dataSize)
Sets new dimensions of the image object.
- void [Release](#) ()
- uint64_t [GetID](#) () const
Gets a unique ID for this image that is associated with a stream.
- uint64_t [GetStreamIndex](#) () const
Gets the stream channel index of where the image is received.
- void * [GetData](#) () const
Gets a pointer to the data associated with the image.
- float [GetDataAbsoluteMax](#) () const
Get the value for which no image data will exceed.
- float [GetDataAbsoluteMin](#) () const
Get the value for which no image data will be less than.
- void * [GetPrivateData](#) () const
Gets a pointer to the user passed data associated with the image.
- size_t [GetBufferSize](#) () const
Gets the size of the buffer associated with the image in bytes.
- void [DeepCopy](#) (const [ImagePtr](#) pSrcImage)
Performs a deep copy of the [Image](#).
- size_t [GetWidth](#) () const
Gets the width of the image in pixels.
- size_t [GetHeight](#) () const

- Gets the height of the image in pixels.*

 - `size_t GetStride () const`
- Gets the stride of the image in bytes.*

 - `size_t GetBitsPerPixel () const`
- Gets the number of bits used per pixel in the image.*

 - `size_t GetNumChannels () const`
- Gets the number of channels (depth) used in the image.*

 - `size_t GetXOffset () const`
- Gets the ROI x offset in pixels for this image.*

 - `size_t GetYOffset () const`
- Gets the ROI y offset in pixels for this image.*

 - `size_t GetXPadding () const`
- Gets the x padding in bytes for this image.*

 - `size_t GetYPadding () const`
- Gets the y padding in bytes for this image.*

 - `uint64_t GetFrameID () const`
- Gets the frame ID for this image.*

 - `size_t GetPayloadType () const`
- Gets the payload type that was transmitted.*

 - `TLPayloadType GetTLPayloadType () const`
- Gets the GenTL specific payload type that was transmitted.*

 - `uint64_t GetTLPixelFormat () const`
- Gets the pixel format of the image.*

 - `TLPixelFormatNamespace GetTLPixelFormatNamespace () const`
- Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.*

 - `GenICam::gcstring GetPixelFormatName () const`
- Returns a string value that represents this image's pixel format.*

 - `Spinnaker::PixelFormatEnums GetPixelFormat () const`
- Returns an enum value that represents the pixel format of this image.*

 - `Spinnaker::PixelFormatIntType GetPixelFormatIntType () const`
- Returns an enum value that represents the integer type used in the pixel format of this image.*

 - `bool IsIncomplete () const`
- Returns a boolean value indicating if this image was incomplete.*

 - `size_t GetValidPayloadSize () const`
- Returns the size of valid data in the image payload.*

 - `uint64_t GetChunkLayoutId () const`
- Returns the id of the chunk data layout.*

 - `uint64_t GetTimeStamp () const`
- Gets the time stamp for the image in nanoseconds.*

 - `void Save (const char *pFilename, ImageFileFormat format=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) const`
- Saves the image to the specified file name with the file format specified.*

 - `void Save (const char *pFilename, PNGOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, PPMOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, PGMOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, TIFFOption &pOption) const`
- Saves the image to the specified file name with the options specified.*

 - `void Save (const char *pFilename, JPEGOption &pOption) const`

- Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char *pFilename, [JPG2Option](#) &pOption) const
Saves the image to the specified file name with the options specified.
- void [Save](#) (const char *pFilename, [BMPOption](#) &pOption) const
Saves the image to the specified file name with the options specified.
- void [Save](#) (const char *pFilename, [SIOption](#) &option) const
Saves the image to the specified file name with the options specified.
- bool [HasChunkData](#) () const
Checks if the image contains chunk data.
- const [ChunkData](#) & [GetChunkData](#) () const
Returns a pointer to a chunk data interface.
- void [CalculateStatistics](#) ([ImageStatistics](#) &pStatistics)
Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.
- bool [HasCRC](#) () const
Checks if the image contains ImageCRC checksum from chunk data.
- bool [CheckCRC](#) () const
Checks if the computed checksum matches with chunk data's ImageCRC.
- size_t [GetImageSize](#) () const
Returns the size of the image.
- bool [IsInUse](#) ()
Returns true if the image is still in use by the stream.
- [ImageStatus](#) [GetImageStatus](#) () const
Returns data integrity status of the image returned from GetNextImage()
- bool [IsCompressed](#) () const
Returns a boolean value indicating whether this image is compressed.

Static Public Member Functions

- static [ImagePtr](#) [Create](#) ()
Create an image object.
- static [ImagePtr](#) [Create](#) (const [ImagePtr](#) image)
Create an image object that is a deep copy of the input image.
- static [ImagePtr](#) [Create](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void *pData)
Create an image object with the specified parameters.
- static [ImagePtr](#) [Create](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData, [TLPayloadType](#) dataPayloadType, size_t dataSize)
Create an image object with the specified parameters.
- static [ImagePtr](#) [Load](#) (const char *pFilename, [ImageFileFormat](#) format=[SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_E](#))
Loads the image from the specified file name with the options specified.
- static const char * [GetImageStatusDescription](#) ([ImageStatus](#) status)
Returns a string describing the meaning of the status enum.

Protected Member Functions

- [ImageData](#) * [GetImageData](#) () const
- [Image](#) ()
- [Image](#) (const [ImagePtr](#) image)
- [Image](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData)
- [Image](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [PixelFormatEnums](#) pixelFormat, void *pData, [TLPayloadType](#) payloadType, size_t payloadSize)
- [ImagePtr](#) [CreateShared](#) () const
- void [DeepCopy](#) (const [Image](#) &pSrcImage)

Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageConverterIpp](#)
- class [ImageImpl](#)
- class [ImageListImpl](#)
- class [ImageFiler](#)
- class [ImageProcessorImpl](#)
- class [ImageStatsCalculator](#)
- class [ImageUtilityImpl](#)
- class [ImageUtilityPolarizationImpl](#)
- class [ImageUtilityPlanarImpl](#)

15.116.1 Detailed Description

The image object class.

15.116.2 Constructor & Destructor Documentation

15.116.2.1 `~Image()`

```
virtual ~Image ( ) [virtual]
```

Virtual destructor.

15.116.2.2 `Image()` [1/4]

```
Image ( ) [protected]
```

15.116.2.3 `Image()` [2/4]

```
Image (
    const ImagePtr image ) [protected]
```

15.116.2.4 Image() [3/4]

```
Image (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData ) [protected]
```

15.116.2.5 Image() [4/4]

```
Image (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    TLPayloadType payloadType,
    size_t payloadSize ) [protected]
```

15.116.3 Member Function Documentation**15.116.3.1 CalculateStatistics()**

```
void CalculateStatistics (
    ImageStatistics & pStatistics ) [virtual]
```

Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.

Parameters

<i>pStatistics</i>	The statistics of an image.

Implements [IImage](#).

15.116.3.2 CheckCRC()

```
bool CheckCRC ( ) const [virtual]
```

Checks if the computed checksum matches with chunk data's ImageCRC.

Returns

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

Implements [Image](#).

15.116.3.3 Create() [1/4]

```
static ImagePtr Create ( ) [static]
```

Create an image object.

15.116.3.4 Create() [2/4]

```
static ImagePtr Create (
    const ImagePtr image ) [static]
```

Create an image object that is a deep copy of the input image.

Parameters

<i>image</i>	The input image to copy

15.116.3.5 Create() [3/4]

```
static ImagePtr Create (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    TLPayloadType dataPayloadType,
    size_t dataSize ) [static]
```

Create an image object with the specified parameters.

This function is used to create an image from existing image data with a specific payload type, such as a compressed image.

Note that images with chunk payload types are saved with only the image data preserved. Remember to specify the non-chunk equivalent payload type when creating images with these chunk payload types. For example, images need to be created with PAYLOAD_TYPE_IMAGE payload type if the original image had PAYLOAD_TYPE_EXTENDED_CHUNK payload type.

Parameters

<i>width</i>	The image width in pixels
<i>height</i>	The image height in pixels
<i>offsetX</i>	The image X offset
<i>offsetY</i>	The image Y offset
<i>pixelFormat</i>	The image pixel format
<i>pData</i>	The compressed image data
<i>dataPayloadType</i>	The payload type of the data. This value can be retrieved from an existing image by using the GetTLPayloadType() function call.
<i>dataSize</i>	The size of the provided data in bytes

See also

[GetTLPayloadType\(\)](#)

15.116.3.6 Create() [4/4]

```
static ImagePtr Create (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData ) [static]
```

Create an image object with the specified parameters.

Parameters

<i>width</i>	The image width in pixels
<i>height</i>	The image height in pixels
<i>offsetX</i>	The image X offset
<i>offsetY</i>	The image Y offset
<i>pixelFormat</i>	The image pixel format
<i>pData</i>	The image data

15.116.3.7 CreateShared()

```
ImagePtr CreateShared ( ) const [protected]
```

15.116.3.8 DeepCopy() [1/2]

```
void DeepCopy (
    const Image & pSrcImage ) [protected]
```

15.116.3.9 DeepCopy() [2/2]

```
void DeepCopy (
    const ImagePtr pSrcImage ) [virtual]
```

Performs a deep copy of the [Image](#).

After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The [Image](#)'s current buffer will not be released.

Parameters

<i>pSrcImage</i>	The Image to copy the data from.

Implements [Image](#).

15.116.3.10 GetBitsPerPixel()

```
size_t GetBitsPerPixel ( ) const [virtual]
```

Gets the number of bits used per pixel in the image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The number of bits used per pixel.

Implements [Image](#).

15.116.3.11 GetBufferSize()

```
size_t GetBufferSize ( ) const [virtual]
```

Gets the size of the buffer associated with the image in bytes.

For user created images, this function returns the size of the user provided data if the data size was provided. If the data size was not provided, the buffer size is calculated based on the image dimensions and pixel format.

Returns

The size of the buffer, in bytes.

Implements [Image](#).

15.116.3.12 GetChunkData()

```
const ChunkData& GetChunkData ( ) const [virtual]
```

Returns a pointer to a chunk data interface.

No ownership is transferred, the chunk data interface reference is valid until [Image::Release\(\)](#) is called on this image.

Returns

[ChunkData](#) interface that provides access to image chunks.

Implements [IImage](#).

15.116.3.13 GetChunkLayoutId()

```
uint64_t GetChunkLayoutId ( ) const [virtual]
```

Returns the id of the chunk data layout.

Returns

uint64_t value representing the id of the chunk data layout.

Implements [IImage](#).

15.116.3.14 GetColorProcessing()

```
ColorProcessingAlgorithm GetColorProcessing ( ) const [virtual]
```

Gets the color algorithm used to produce the image.

See also

[ImageProcessor::Convert\(\)](#)

Returns

The color processing algorithm used to produce the image.

Implements [IImage](#).

15.116.3.15 GetData()

```
void* GetData ( ) const [virtual]
```

Gets a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

Returns

A pointer to the image data.

Implements [Image](#).

15.116.3.16 GetDataAbsoluteMax()

```
float GetDataAbsoluteMax ( ) const [virtual]
```

Get the value for which no image data will exceed.

Returns

the maximim theoretical image data value

Implements [Image](#).

15.116.3.17 GetDataAbsoluteMin()

```
float GetDataAbsoluteMin ( ) const [virtual]
```

Get the value for which no image data will be less than.

Returns

the minimum theoretical image data value

Implements [Image](#).

15.116.3.18 GetFrameID()

```
uint64_t GetFrameID ( ) const [virtual]
```

Gets the frame ID for this image.

Returns

The frame ID.

Implements [IImage](#).

15.116.3.19 GetHeight()

```
size_t GetHeight ( ) const [virtual]
```

Gets the height of the image in pixels.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The height in pixels.

Implements [IImage](#).

15.116.3.20 GetID()

```
uint64_t GetID ( ) const [virtual]
```

Gets a unique ID for this image that is associated with a stream.

Note that an ID of 0 indicates that the image is no longer associated with a stream or is a copy of the original image.

See also

[GetFrameID\(\)](#)

Returns

The 64 bit unique id for this image.

Implements [IImage](#).

15.116.3.21 GetImageData()

```
ImageData* GetImageData ( ) const [protected], [virtual]
```

Implements [IImage](#).

15.116.3.22 GetImageSize()

```
size_t GetImageSize ( ) const [virtual]
```

Returns the size of the image.

For chunk images, only the size of chunk image portion is reported here. The entire chunk data payload can be queried by [GetValidPayloadSize\(\)](#). For compressed images, this value may be different than the image size once decompressed.

See also

[GetBufferSize\(\)](#)

[GetValidPayloadSize\(\)](#)

Returns

The image size in bytes.

Implements [IImage](#).

15.116.3.23 GetImageStatus()

```
ImageStatus GetImageStatus ( ) const [virtual]
```

Returns data integrity status of the image returned from [GetNextImage\(\)](#)

Returns

Returns whether image has any data integrity issues.

Implements [IImage](#).

15.116.3.24 GetImageStatusDescription()

```
static const char* GetImageStatusDescription (
    ImageStatus status ) [static]
```

Returns a string describing the meaning of the status enum.

Returns

Returns the meaning of the status enum.

15.116.3.25 GetNumChannels()

```
size_t GetNumChannels ( ) const [virtual]
```

Gets the number of channels (depth) used in the image.

Returns 0 if the number of channels for the given pixel format is unknown.

Returns

The number of channels per pixel.

Implements [Image](#).

15.116.3.26 GetPayloadType()

```
size_t GetPayloadType ( ) const [virtual]
```

Gets the payload type that was transmitted.

This is a device types specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

Device types specific payload type.

Implements [Image](#).

15.116.3.27 GetPixelFormat()

```
Spinnaker::PixelFormatEnums GetPixelFormat ( ) const [virtual]
```

Returns an enum value that represents the pixel format of this image.

The enum can be used with the easy access [GenICam](#) features available through the [Camera.h](#) header file. This easy access enum can also be used in the [ImageProcessor::Convert\(\)](#) function.

See also

[ImageProcessor::Convert\(\)](#)

Returns

enum value representing the PixelFormat.

Implements [IImage](#).

15.116.3.28 GetPixelFormatIntType()

```
Spinnaker::PixelFormatIntType GetPixelFormatIntType ( ) const [virtual]
```

Returns an enum value that represents the integer type used in the pixel format of this image.

Returns

enum value representing the integer type used.

Implements [IImage](#).

15.116.3.29 GetPixelFormatName()

```
GenICam::gcstring GetPixelFormatName ( ) const [virtual]
```

Returns a string value that represents this image's pixel format.

The string is a valid SFNC name that maps to the underlying TL specific pixel format. This is the most generic way to identify the pixel format of the image.

Returns

string value representing the PixelFormat.

Implements [IImage](#).

15.116.3.30 GetPrivateData()

```
void* GetPrivateData ( ) const [virtual]
```

Gets a pointer to the user passed data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

TODO: no way to set private data for image yet.

Returns

A pointer to the user passed data pointer.

Implements [IImage](#).

15.116.3.31 GetStreamIndex()

```
uint64_t GetStreamIndex ( ) const [virtual]
```

Gets the stream channel index of where the image is received.

Returns

The stream channel index of where this image is received.

Implements [IImage](#).

15.116.3.32 GetStride()

```
size_t GetStride ( ) const [virtual]
```

Gets the stride of the image in bytes.

The stride of an image is how many bytes are in each row. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The stride in bytes.

Implements [IImage](#).

15.116.3.33 GetTimeStamp()

```
uint64_t GetTimeStamp ( ) const [virtual]
```

Gets the time stamp for the image in nanoseconds.

Returns

The time stamp of the image.

Implements [IImage](#).

15.116.3.34 GetTLPayloadType()

```
TLPayloadType GetTLPayloadType ( ) const [virtual]
```

Gets the GenTL specific payload type that was transmitted.

This is a Transport Layer specific value that identifies how the image was transmitted. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

Transport Layer specific payload type.

Implements [IImage](#).

15.116.3.35 GetTLPixelFormat()

```
uint64_t GetTLPixelFormat ( ) const [virtual]
```

Gets the pixel format of the image.

This is a Transport Layer specific pixel format that identifies how the pixels in the image should be interpreted. To understand how to interpret this value it is necessary to know what the transport layer namespace is. This can be retrieved through a call to [GetTLPixelFormatNamespace\(\)](#). This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

See also

[GetTLPixelFormatNamespace\(\)](#)

Returns

Transport Layer specific pixel format.

Implements [IImage](#).

15.116.3.36 GetTLPixelFormatNamespace()

```
TLPixelFormatNamespace GetTLPixelFormatNamespace ( ) const [virtual]
```

Returns an enum value that represents the namespace in which this image's TL specific pixel format resides.

This information is important to properly interpret the value returned by [GetTLPixelFormat\(\)](#)

See also

[GetTLPixelFormat\(\)](#)

Returns

enum value representing the PixelFormatNamespace.

Implements [IImage](#).

15.116.3.37 GetValidPayloadSize()

```
size_t GetValidPayloadSize ( ) const [virtual]
```

Returns the size of valid data in the image payload.

This is the actual amount of data read from the device. A user created image has a payload size of zero. The value returned here can be equal to the value returned by [GetImageSize\(\)](#) if image data is the only payload. Note that [GetBufferSize\(\)](#) returns the total size of bytes allocated for the image and could be equal to or greater than the size returned by this function.

See also

[GetBufferSize\(\)](#)

[GetImageSize\(\)](#)

Returns

size_t value representing valid payload.

Implements [IImage](#).

15.116.3.38 GetWidth()

```
size_t GetWidth ( ) const [virtual]
```

Gets the width of the image in pixels.

This information is retrieved from the Transport Layer image format headers. It is retrieved on a per image basis.

Returns

The width in pixels.

Implements [IImage](#).

15.116.3.39 GetXOffset()

```
size_t GetXOffset ( ) const [virtual]
```

Gets the ROI x offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x offset in pixels.

Implements [Image](#).

15.116.3.40 GetXPadding()

```
size_t GetXPadding ( ) const [virtual]
```

Gets the x padding in bytes for this image.

This is the number of bytes at the end of each line to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x padding in bytes.

Implements [Image](#).

15.116.3.41 GetYOffset()

```
size_t GetYOffset ( ) const [virtual]
```

Gets the ROI y offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y offset in pixels.

Implements [Image](#).

15.116.3.42 GetYPadding()

```
size_t GetYPadding ( ) const [virtual]
```

Gets the y padding in bytes for this image.

This is the number of bytes at the end of each image to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y padding in bytes.

Implements [Image](#).

15.116.3.43 HasChunkData()

```
bool HasChunkData ( ) const [virtual]
```

Checks if the image contains chunk data.

Returns

Returns true if image contains chunk data and false otherwise.

Implements [Image](#).

15.116.3.44 HasCRC()

```
bool HasCRC ( ) const [virtual]
```

Checks if the image contains ImageCRC checksum from chunk data.

Returns

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

Implements [Image](#).

15.116.3.45 IsCompressed()

```
bool IsCompressed ( ) const [virtual]
```

Returns a boolean value indicating whether this image is compressed.

Returns

Returns true if image is compressed, false otherwise.

Implements [IImage](#).

15.116.3.46 IsIncomplete()

```
bool IsIncomplete ( ) const [virtual]
```

Returns a boolean value indicating if this image was incomplete.

An image is marked as incomplete if the transport layer received less data then it requested.

Returns

Returns true if image is incomplete, false otherwise.

Implements [IImage](#).

15.116.3.47 IsInUse()

```
bool IsInUse ( ) [virtual]
```

Returns true if the image is still in use by the stream.

Returns

Returns true if the image is in use and false otherwise.

Implements [IImage](#).

15.116.3.48 Load()

```
static ImagePtr Load (
    const char * pFilename,
    ImageFileFormat format = SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT ) [static]
```

Loads the image from the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to load image with.
<i>pOption</i>	Options to use while loading image.

15.116.3.49 Release()

```
void Release ( ) [virtual]
```

Implements [IImage](#).

15.116.3.50 ResetImage() [1/3]

```
void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    PixelFormatEnums pixelFormat,
    void * pData,
    TLPayloadType dataPayloadType,
    size_t dataSize ) [virtual]
```

Sets new dimensions of the image object.

This function is used to create an image from existing image data with a specific payload type, such as a compressed image.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.
<i>pData</i>	Pointer to the image buffer.
<i>dataPayloadType</i>	The payload type of the data. This value can be retrieved from an existing image by using the GetTLPayloadType() function call.
<i>dataSize</i>	The size of the provided data in bytes

See also

[GetTLPayloadType\(\)](#)

Implements [IImage](#).

15.116.3.51 ResetImage() [2/3]

```
void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat ) [virtual]
```

Sets new dimensions of the image object and allocates memory.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.

Implements [Image](#).

15.116.3.52 ResetImage() [3/3]

```
void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData ) [virtual]
```

Sets new dimensions of the image object.

Parameters

<i>width</i>	The width of image in pixels to set.
<i>height</i>	The height of image in pixels to set.
<i>offsetX</i>	The x offset in pixels to set.
<i>offsetY</i>	The y offset in pixels to set.
<i>pixelFormat</i>	Pixel format to set.
<i>pData</i>	Pointer to the image buffer.

Implements [Image](#).

15.116.3.53 Save() [1/9]

```
void Save (
```

```
const char * pFilename,
    BMPOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.54 Save() [2/9]

```
void Save (
    const char * pFilename,
    ImageFileFormat format = SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT ) const [virtual]
```

Saves the image to the specified file name with the file format specified.

Note that only the image data is saved regardless of the payload type used to transmit the image from camera to host. Saving a RAW image that was transmitted using the chunk image payload type will only contain the image data portion and none of the other chunks are preserved.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>format</i>	File format to save in.

Implements [IImage](#).

15.116.3.55 Save() [3/9]

```
void Save (
    const char * pFilename,
    JPEGOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.56 Save() [4/9]

```
void Save (  
    const char * pFilename,  
    JPG2Option & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.57 Save() [5/9]

```
void Save (  
    const char * pFilename,  
    PGMOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.58 Save() [6/9]

```
void Save (  
    const char * pFilename,  
    PNGOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.59 Save() [7/9]

```
void Save (
    const char * pFilename,
    PPMOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.60 Save() [8/9]

```
void Save (
    const char * pFilename,
    SIOption & option ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.3.61 Save() [9/9]

```
void Save (
    const char * pFilename,
    TIFFOption & pOption ) const [virtual]
```

Saves the image to the specified file name with the options specified.

Parameters

<i>pFilename</i>	Filename to save image with.
<i>pOption</i>	Options to use while saving image.

Implements [IImage](#).

15.116.4 Friends And Related Function Documentation

15.116.4.1 IDataStream

```
friend class IDataStream [friend]
```

15.116.4.2 ImageConverter

```
friend class ImageConverter [friend]
```

15.116.4.3 ImageConverterIpp

```
friend class ImageConverterIpp [friend]
```

15.116.4.4 ImageFiler

```
friend class ImageFiler [friend]
```

15.116.4.5 ImageImpl

```
friend class ImageImpl [friend]
```

15.116.4.6 ImageListImpl

```
friend class ImageListImpl [friend]
```

15.116.4.7 ImageProcessorImpl

```
friend class ImageProcessorImpl [friend]
```

15.116.4.8 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

15.116.4.9 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

15.116.4.10 ImageUtilityPlanarImpl

```
friend class ImageUtilityPlanarImpl [friend]
```

15.116.4.11 ImageUtilityPolarizationImpl

```
friend class ImageUtilityPolarizationImpl [friend]
```

15.116.4.12 Stream

```
friend class Stream [friend]
```

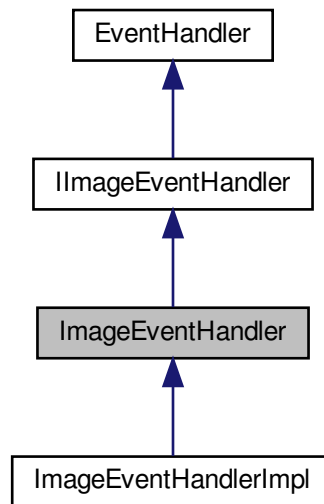
The documentation for this class was generated from the following file:

- [include/Image.h](#)

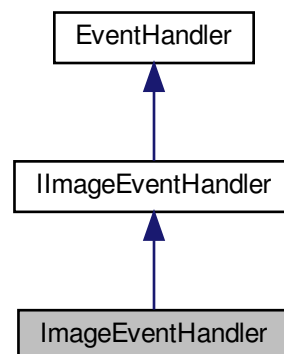
15.117 ImageEventHandler Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEventHandler:



Collaboration diagram for ImageEventHandler:



Public Member Functions

- [ImageEventHandler](#) ()
Default image handler constructor.
- virtual [~ImageEventHandler](#) ()
Virtual Destructor.

Protected Member Functions

- `ImageEventHandler & operator=` (const `ImageEventHandler` &)
Assignment operator.
- virtual void `OnImageEvent` (`ImagePtr` image)=0
Image event callback.

Additional Inherited Members

15.117.1 Detailed Description

A handler for capturing image arrival events.

15.117.2 Constructor & Destructor Documentation

15.117.2.1 ImageEventHandler()

```
ImageEventHandler ( )
```

Default image handler constructor.

15.117.2.2 ~ImageEventHandler()

```
virtual ~ImageEventHandler ( ) [virtual]
```

Virtual Destructor.

15.117.3 Member Function Documentation

15.117.3.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [protected], [pure virtual]
```

`Image` event callback.

Parameters

<code>image</code>	The <code>ImagePtr</code> object

Implements [ImageEventHandler](#).

Implemented in [ImageEventHandlerImpl](#), and [ImageEventHandlerImpl](#).

15.117.3.2 operator=()

```
ImageEventHandler& operator= (  
    const ImageEventHandler & ) [protected]
```

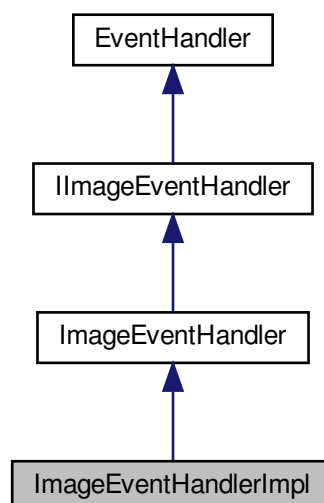
Assignment operator.

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

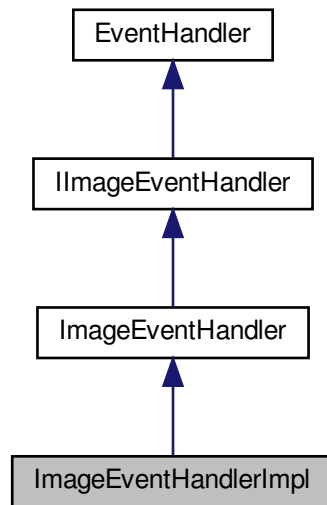
- [include/ImageEventHandler.h](#)

15.118 ImageEventHandlerImpl Class Reference

Inheritance diagram for ImageEventHandlerImpl:



Collaboration diagram for ImageEventHandlerImpl:



Public Member Functions

- [ImageEventHandlerImpl](#) (string deviceSerial)
- [~ImageEventHandlerImpl](#) ()
- void [OnImageEvent](#) ([ImagePtr](#) image)
Image event callback.
- [ImageEventHandlerImpl](#) ([CameraPtr](#) pCam)
- [~ImageEventHandlerImpl](#) ()
- void [OnImageEvent](#) ([ImagePtr](#) image)
Image event callback.
- int [getImageCount](#) ()
- int [getMaxImages](#) ()

Additional Inherited Members

15.118.1 Constructor & Destructor Documentation

15.118.1.1 ImageEventHandlerImpl() [1/2]

```
ImageEventHandlerImpl (
    string deviceSerial ) [inline]
```

15.118.1.2 ~ImageEventHandlerImpl() [1/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

15.118.1.3 ImageEventHandlerImpl() [2/2]

```
ImageEventHandlerImpl (
    CameraPtr pCam ) [inline]
```

15.118.1.4 ~ImageEventHandlerImpl() [2/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

15.118.2 Member Function Documentation

15.118.2.1 getImageCount()

```
int getImageCount ( ) [inline]
```

15.118.2.2 getMaxImages()

```
int getMaxImages ( ) [inline]
```

15.118.2.3 OnImageEvent() [1/2]

```
void OnImageEvent (
    ImagePtr image ) [virtual]
```

Image event callback.

Parameters

<i>image</i>	The ImagePtr object

Implements [ImageEventHandler](#).

15.118.2.4 OnImageEvent() [2/2]

```
void OnImageEvent (
    ImagePtr image ) [inline], [virtual]
```

Image event callback.

Parameters

<i>image</i>	The ImagePtr object

Implements [ImageEventHandler](#).

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

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)
- [src/ImageEvents/ImageEvents.cpp](#)

15.119 ImageInfo Struct Reference

Public Member Functions

- [ImageInfo](#) (string filename)

Public Attributes

- `size_t` [imageWidth](#)
- `size_t` [imageHeight](#)
- [PixelFormatEnums](#) [pixelFormat](#)
- `string` [imageFileName](#)
- `std::shared_ptr< fstream >` [imageFile](#)

15.119.1 Constructor & Destructor Documentation

15.119.1.1 ImageInfo()

```
ImageInfo (
    string filename ) [inline]
```

15.119.2 Member Data Documentation

15.119.2.1 imageFile

```
std::shared_ptr<fstream> imageFile
```

15.119.2.2 imageFileName

```
string imageFileName
```

15.119.2.3 imageHeight

```
size_t imageHeight
```

15.119.2.4 imageWidth

```
size_t imageWidth
```

15.119.2.5 pixelFormat

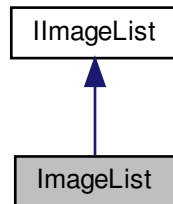
```
PixelFormatEnums pixelFormat
```

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

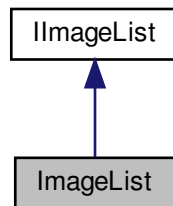
- [src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp](#)

15.120 ImageList Class Reference

Inheritance diagram for ImageList:



Collaboration diagram for ImageList:



Public Member Functions

- [ImageList](#) (void)
Default constructor.
- virtual [~ImageList](#) (void)
Virtual destructor.
- [ImageList](#) (const [ImageList](#) &iface)
Copy constructor.
- [ImageList](#) & [operator=](#) (const [ImageList](#) &iface)
Assignment operator.
- [ImagePtr](#) [operator\[\]](#) (unsigned int index)
Array subscription operators.
- const [ImagePtr](#) [operator\[\]](#) (unsigned int index) const
- void [Release](#) ()
- unsigned int [GetSize](#) () const
Returns the size of the image list.
- [ImagePtr](#) [GetByIndex](#) (unsigned int index) const

- Returns a pointer to an image object at the "index".*
- `ImagePtr GetByPixelFormat (PixelFormatEnums pixelFormat) const`
Returns a pointer to the first image object with the pixel format from the list.
- `void Clear ()`
Clears the list of images and destroys their corresponding reference counted objects.
- `void Add (ImagePtr image)`
Adds a copy of an image object.
- `void RemoveByIndex (unsigned int index)`
Removes an image at "index" and destroys its corresponding reference counted object.
- `void RemoveByPixelFormat (PixelFormatEnums pixelFormat)`
Removes an image object with the image pixel format from the list.
- `void Append (const ImageList &list)`
Appends a copy of the image list.
- `void Save (const char *filename)`
Saves an image list as an object to a file.

Static Public Member Functions

- `static ImageList Load (const char *filename)`
Creates an image list object from file.

Additional Inherited Members

15.120.1 Constructor & Destructor Documentation

15.120.1.1 ImageList() [1/2]

```
ImageList (
    void )
```

Default constructor.

15.120.1.2 ~ImageList()

```
virtual ~ImageList (
    void ) [virtual]
```

Virtual destructor.

15.120.1.3 ImageList() [2/2]

```
ImageList (
    const ImageList & iface )
```

Copy constructor.

15.120.2 Member Function Documentation

15.120.2.1 Add()

```
void Add (
    ImagePtr image ) [virtual]
```

Adds a copy of an image object.

Parameters

<i>image</i>	An image object to be added to this list.

Implements [ImageList](#).

15.120.2.2 Append()

```
void Append (
    const ImageList & list ) [virtual]
```

Appends a copy of the image list.

Parameters

<i>list</i>	Another ImageList object, whose elements are added to this list.

Implements [ImageList](#).

15.120.2.3 Clear()

```
void Clear ( ) [virtual]
```

Clears the list of images and destroys their corresponding reference counted objects.

This is necessary in order to clean up the parent interface. It is important that the camera list is destroyed or is cleared before calling `system->ReleaseInstance()` or else the call to `system->ReleaseInstance()` will result in an error message thrown that a reference to the camera is still held.

See also

[System::ReleaseInstance\(\)](#)

Implements [IImageList](#).

15.120.2.4 GetByIndex()

```
ImagePtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to an image object at the "index".

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to retrieve the Image object

Returns

A pointer to an image object.

Implements [IImageList](#).

15.120.2.5 GetByPixelFormat()

```
ImagePtr GetByPixelFormat (
    PixelFormatEnums pixelFormat ) const [virtual]
```

Returns a pointer to the first image object with the pixel format from the list.

This function will return a NULL [ImagePtr](#) if no matching pixel format is found.

Parameters

<i>pixelFormat</i>	The pixel format of the Image object to retrieve

Returns

A pointer to an image object.

Implements [IImageList](#).

15.120.2.6 GetSize()

```
unsigned int GetSize ( ) const [virtual]
```

Returns the size of the image list.

The size is the number of [Image](#) objects stored in the list.

Returns

An integer that represents the list size.

Implements [IImageList](#).

15.120.2.7 Load()

```
static ImageList Load (
    const char * filename ) [static]
```

Creates an image list object from file.

Parameters

<i>filename</i>	Name of the file to load an image object from.

See also

[Save\(\)](#)

15.120.2.8 operator=()

```
ImageList& operator= (
    const ImageList & iface )
```

Assignment operator.

15.120.2.9 operator[]() [1/2]

```
ImagePtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IImageList](#).

15.120.2.10 operator[]() [2/2]

```
const ImagePtr operator[] (
    unsigned int index ) const
```

15.120.2.11 Release()

```
void Release ( ) [virtual]
```

Implements [ImageList](#).

15.120.2.12 RemoveByIndex()

```
void RemoveByIndex (
    unsigned int index ) [virtual]
```

Removes an image at "index" and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

<i>index</i>	The index at which to remove the Image object

Implements [ImageList](#).

15.120.2.13 RemoveByPixelFormat()

```
void RemoveByPixelFormat (
    PixelFormatEnums pixelFormat ) [virtual]
```

Removes an image object with the image pixel format from the list.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the list does not have any image with the specified pixel format.

Parameters

<i>pixelFormat</i>	The pixel format of the image object to remove

Implements [ImageList](#).

15.120.2.14 Save()

```
void Save (
    const char * filename ) [virtual]
```

Saves an image list as an object to a file.

Parameters

<i>filename</i>	Name of the file to save the current image list object to. It is recommended to use the file extension 'sil'.

See also

[Load\(\)](#)

Implements [IImageList](#).

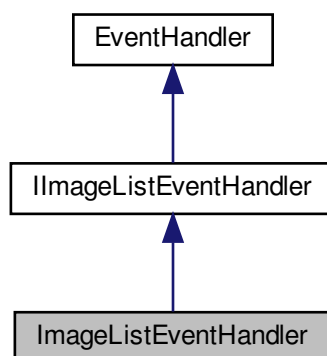
The documentation for this class was generated from the following file:

- include/[ImageList.h](#)

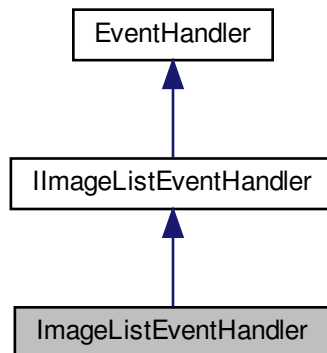
15.121 ImageListEventHandler Class Reference

A handler for capturing image list arrival events.

Inheritance diagram for ImageListEventHandler:



Collaboration diagram for ImageListEventHandler:



Public Member Functions

- [ImageListEventHandler](#) ()
Default Constructor.
- virtual [~ImageListEventHandler](#) ()
Virtual Destructor.

Protected Member Functions

- [ImageListEventHandler](#) & [operator=](#) (const [ImageListEventHandler](#) &)
Assignment operator.
- virtual void [OnImageListEvent](#) ([ImageList](#) imageList)=0
[Image](#) list event callback.

Additional Inherited Members

15.121.1 Detailed Description

A handler for capturing image list arrival events.

15.121.2 Constructor & Destructor Documentation

15.121.2.1 ImageListEventHandler()

```
ImageListEventHandler ( )
```

Default Constructor.

15.121.2.2 ~ImageListEventHandler()

```
virtual ~ImageListEventHandler ( ) [virtual]
```

Virtual Destructor.

15.121.3 Member Function Documentation

15.121.3.1 OnImageListEvent()

```
virtual void OnImageListEvent (
    ImageList imageList ) [protected], [pure virtual]
```

[Image](#) list event callback.

Parameters

<i>image</i>	The ImageList object

Implements [IImageListEventHandler](#).

15.121.3.2 operator=()

```
ImageListEventHandler& operator= (
    const ImageListEventHandler & ) [protected]
```

Assignment operator.

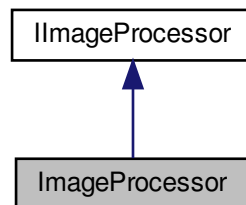
The documentation for this class was generated from the following file:

- [include/ImageListEventHandler.h](#)

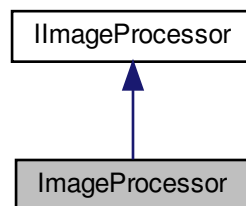
15.122 ImageProcessor Class Reference

[Image](#) post processing class for converting a source image to another pixel format.

Inheritance diagram for ImageProcessor:



Collaboration diagram for ImageProcessor:



Public Member Functions

- [ImageProcessor](#) ()
Default constructor.
- virtual [~ImageProcessor](#) ()
Default destructor.
- [ImageProcessor](#) (const [ImageProcessor](#) &iface)
Copy constructor.
- [ImageProcessor](#) & [operator=](#) (const [ImageProcessor](#) &iface)
Assignment operator.
- void [SetColorProcessing](#) ([ColorProcessingAlgorithm](#) colorAlgorithm)
Sets the color processing algorithm used at the time of the [Convert\(\)](#) call, therefore the most recent execution of this function will take precedence.
- [ColorProcessingAlgorithm](#) [GetColorProcessing](#) ()
Gets the default color processing algorithm.

- void [SetNumDecompressionThreads](#) (unsigned int numThreads)
Sets the default number of threads used for image decompression during [Convert\(\)](#).
- unsigned int [GetNumDecompressionThreads](#) ()
Gets the number of threads used for image decompression during [Convert\(\)](#).
- [ImagePtr](#) [Convert](#) (const [ImagePtr](#) &srcImage, [PixelFormatEnums](#) destFormat) const
Converts the source image buffer to the specified destination pixel format and returns the result in a new image.
- void [Convert](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [PixelFormatEnums](#) destFormat) const
Converts the source image buffer to the specified destination pixel format and stores the result in the destination image.
- [ImagePtr](#) [Convert](#) (const [ImageList](#) &srcImageList, [PixelFormatEnums](#) destFormat) const
Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.
- void [Convert](#) (const [ImageList](#) &srcImageList, [ImagePtr](#) &destImage, [PixelFormatEnums](#) destFormat) const
Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.
- [ImagePtr](#) [ApplyGamma](#) (const [ImagePtr](#) &srcImage, float gamma, bool applyGammaInverse=false) const
Applies gamma correction to the source image and returns a new image.
- void [ApplyGamma](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, float gamma, bool applyGammaInverse=false) const
Applies gamma correction to the source image and stores the result in the destination image.

Protected Member Functions

- void [Convert](#) (const [Image](#) *pSrcImage, [Image](#) *pDestImage, [PixelFormatEnums](#) destFormat) const

Friends

- class [ImageFiler](#)

Additional Inherited Members

15.122.1 Detailed Description

[Image](#) post processing class for converting a source image to another pixel format.

All supported input image pixel formats can be converted to supported output image pixel formats. If the input pixel format is a compressed format, the decompression will occur before converting to the output pixel format.

List of supported input image pixel formats:

- [PixelFormat_Mono8](#)
- [PixelFormat_Mono16](#)
- [PixelFormat_BayerGR8](#)
- [PixelFormat_BayerRG8](#)
- [PixelFormat_BayerGB8](#)
- [PixelFormat_BayerBG8](#)
- [PixelFormat_BayerGR16](#)
- [PixelFormat_BayerRG16](#)

- PixelFormat_BayerGB16
- PixelFormat_BayerBG16
- PixelFormat_Mono12Packed
- PixelFormat_BayerGR12Packed
- PixelFormat_BayerRG12Packed
- PixelFormat_BayerGB12Packed
- PixelFormat_BayerBG12Packed
- PixelFormat_YUV411Packed
- PixelFormat_YUV422Packed
- PixelFormat_YUV444Packed
- PixelFormat_Mono12p
- PixelFormat_BayerGR12p
- PixelFormat_BayerRG12p
- PixelFormat_BayerGB12p
- PixelFormat_BayerBG12p
- PixelFormat_YCbCr8
- PixelFormat_YCbCr422_8
- PixelFormat_YCbCr411_8
- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_Mono10Packed
- PixelFormat_BayerGR10Packed
- PixelFormat_BayerRG10Packed
- PixelFormat_BayerGB10Packed
- PixelFormat_BayerBG10Packed
- PixelFormat_Mono10p
- PixelFormat_BayerGR10p
- PixelFormat_BayerRG10p
- PixelFormat_BayerGB10p
- PixelFormat_BayerBG10p
- PixelFormat_Mono10
- PixelFormat_Mono12
- PixelFormat_Mono14
- PixelFormat_BayerBG10
- PixelFormat_BayerBG12
- PixelFormat_BayerGB10

- PixelFormat_BayerGB12
- PixelFormat_BayerGR10
- PixelFormat_BayerGR12
- PixelFormat_BayerRG10
- PixelFormat_BayerRG12
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_R12
- PixelFormat_G12
- PixelFormat_B12
- PixelFormat_YUV8_UYV
- PixelFormat_YUV411_8_UYYVYY
- PixelFormat_YUV422_8
- PixelFormat_Polarized8
- PixelFormat_Polarized10p
- PixelFormat_Polarized12p
- PixelFormat_Polarized16
- PixelFormat_BayerRGPolarized8
- PixelFormat_BayerRGPolarized10p
- PixelFormat_BayerRGPolarized12p
- PixelFormat_BayerRGPolarized16
- PixelFormat_LLCMono8
- PixelFormat_LLCBayerRG8
- PixelFormat_JPEGMono8
- PixelFormat_JPEGColor8
- PixelFormat_Raw16
- PixelFormat_Raw8
- PixelFormat_R12_Jpeg
- PixelFormat_GR12_Jpeg
- PixelFormat_GB12_Jpeg
- PixelFormat_B12_Jpeg

List of supported output image pixel formats

- PixelFormat_Mono8

- PixelFormat_Mono16
- PixelFormat_BayerBG8
- PixelFormat_BayerGB8
- PixelFormat_BayerRG8
- PixelFormat_BayerGR8
- PixelFormat_BayerBG16
- PixelFormat_BayerGB16
- PixelFormat_BayerRG16
- PixelFormat_BayerGR16
- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGB8
- PixelFormat_RGBa8
- PixelFormat_BGR16
- PixelFormat_RGB16
- PixelFormat_R12
- PixelFormat_G12
- PixelFormat_B12

15.122.2 Constructor & Destructor Documentation

15.122.2.1 ImageProcessor() [1/2]

`ImageProcessor ( )`

Default constructor.

15.122.2.2 ~ImageProcessor()

`virtual ~ImageProcessor ( ) [virtual]`

Default destructor.

15.122.2.3 ImageProcessor() [2/2]

```
ImageProcessor (
    const ImageProcessor & iface )
```

Copy constructor.

15.122.3 Member Function Documentation

15.122.3.1 ApplyGamma() [1/2]

```
ImagePtr ApplyGamma (
    const ImagePtr & srcImage,
    float gamma,
    bool applyGammaInverse = false ) const [virtual]
```

Applies gamma correction to the source image and returns a new image.

Parameters

<i>srcImage</i>	The source image from which to apply gamma on.
<i>gamma</i>	Gamma value to apply. A value between 0.5 and 4 is acceptable. (Default assuming image-to-screen)
<i>applyGammaInverse</i>	Converts a gamma corrected source image back to the original image using the inverse of the gamma value (used for applying screen-to-image gamma)

Implements [ImageProcessor](#).

15.122.3.2 ApplyGamma() [2/2]

```
void ApplyGamma (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    float gamma,
    bool applyGammaInverse = false ) const [virtual]
```

Applies gamma correction to the source image and stores the result in the destination image.

Parameters

<i>srcImage</i>	The source image from which to apply gamma on.
<i>destImage</i>	The destination image in which the gamma applied image data will be stored.
<i>gamma</i>	Gamma value to apply. A value between 0.5 and 4 is acceptable. (Default assuming image-to-screen)
<i>applyGammaInverse</i>	Converts a gamma corrected source image back to the original image using the inverse of the gamma value (used for applying screen-to-image gamma)

Implements [ImageProcessor](#).

15.122.3.3 Convert() [1/5]

```
void Convert (
    const Image * pSrcImage,
    Image * pDestImage,
    PixelFormatEnums destFormat ) const [protected]
```

15.122.3.4 Convert() [2/5]

```
void Convert (
    const ImageList & srcImageList,
    ImagePtr & destImage,
    PixelFormatEnums destFormat ) const [virtual]
```

Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.

The conversion could encompasses decompression, interleaving and conversion of image data depending on the source pixel format of images in the source image list. The destination image needs to be configured to have the correct buffer size before calling this function. See [ResetImage\(\)](#) to setup the correct buffer size according to specified pixel format.

Note that compressed images are decompressed before any further color processing, interleaving or conversion is performed. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

Note not all the supported image pixel formats described in the class description are supported in this function.

List of supported image pixel formats for the source image list:

- PixelFormat_R12
- PixelFormat_GR12
- PixelFormat_GB12
- PixelFormat_B12
- PixelFormat_R12_Jpeg
- PixelFormat_GR12_Jpeg
- PixelFormat_GB12_Jpeg
- PixelFormat_B12_Jpeg

See also

[PixelFormatEnums](#)

[ResetImage](#)

[SetNumDecompressionThreads](#)

Parameters

<i>srcImageList</i>	List of images from which to convert the images from.
<i>destImage</i>	The destination image in which the converted image data will be stored.
<i>destFormat</i>	Output format of the converted image.

Implements [IImageProcessor](#).

15.122.3.5 Convert() [3/5]

```
ImagePtr Convert (
    const ImageList & srcImageList,
    PixelFormatEnums destFormat ) const [virtual]
```

Converts the source list of image buffers to the specified output pixel format and returns the result in a new image.

The conversion could encompasses decompression, interleaving and conversion of image data depending on the source pixel format of images in the source image list. The destination image does not need to be configured in any way before the call is made.

Note that compressed images are decompressed before any further color processing, interleaving or conversion is performed. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

Note not all the supported image pixel formats described in the class description are supported in this function.

List of supported image pixel formats for the source image list:

- PixelFormat_R12
- PixelFormat_GR12
- PixelFormat_GB12
- PixelFormat_B12
- PixelFormat_R12_Jpeg
- PixelFormat_GR12_Jpeg
- PixelFormat_GB12_Jpeg
- PixelFormat_B12_Jpeg

See also

[PixelFormatEnums](#)

[SetNumDecompressionThreads](#)

Parameters

<i>srcImageList</i>	List of images from which to convert the images from.
<i>destFormat</i>	Output format of the converted image.

Returns

The converted image.

Implements [IImageProcessor](#).

15.122.3.6 Convert() [4/5]

```
void Convert (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    PixelFormatEnums destFormat ) const [virtual]
```

Converts the source image buffer to the specified destination pixel format and stores the result in the destination image.

The destination image needs to be configured to have the correct buffer size before calling this function. See [ResetImage\(\)](#) to setup the correct buffer size according to specified pixel format.

Note that compressed images are decompressed before any further color processing or conversion during this call. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

See also

[PixelFormatEnums](#)

[ResetImage](#)

[SetNumDecompressionThreads](#)

Parameters

<i>srcImage</i>	The source image from which to convert the image from.
<i>destImage</i>	The destination image in which the converted image data will be stored.
<i>destFormat</i>	Output format of the converted image.

Implements [IImageProcessor](#).

15.122.3.7 Convert() [5/5]

```
ImagePtr Convert (
    const ImagePtr & srcImage,
    PixelFormatEnums destFormat ) const [virtual]
```

Converts the source image buffer to the specified destination pixel format and returns the result in a new image.

The destination image does not need to be configured in any way before the call is made.

Note that compressed images are decompressed before any further color processing or conversion during this call. Decompression is multi-threaded and defaults to utilizing one less than the number of concurrent threads supported by the system. The default number of decompression threads can be set with [SetNumDecompressionThreads\(\)](#).

See also

[PixelFormatEnums](#)

[SetNumDecompressionThreads](#)

Parameters

<i>srcImage</i>	The source image from which to convert the image from.
<i>destFormat</i>	Output format of the converted image.

Returns

The converted image.

Implements [IImageProcessor](#).

15.122.3.8 GetColorProcessing()

```
ColorProcessingAlgorithm GetColorProcessing ( ) [virtual]
```

Gets the default color processing algorithm.

See also

[SetColorProcessing\(\)](#)

Returns

The default color processing algorithm.

Implements [IImageProcessor](#).

15.122.3.9 GetNumDecompressionThreads()

```
unsigned int GetNumDecompressionThreads ( ) [virtual]
```

Gets the number of threads used for image decompression during [Convert\(\)](#).

See also

[SetNumDecompressionThreads\(\)](#)

Returns

Number of parallel image decompression threads set to run.

Implements [IImageProcessor](#).

15.122.3.10 operator=()

```
ImageProcessor& operator= (
    const ImageProcessor & iface )
```

Assignment operator.

15.122.3.11 SetColorProcessing()

```
void SetColorProcessing (
    ColorProcessingAlgorithm colorAlgorithm ) [virtual]
```

Sets the color processing algorithm used at the time of the [Convert\(\)](#) call, therefore the most recent execution of this function will take precedence.

The DEFAULT algorithm is deprecated and should not be used in the [ImageProcessor](#) class.

Parameters

<i>colorAlgorithm</i>	The color processing algorithm to set.

See also

[GetColorProcessing\(\)](#)

Implements [IImageProcessor](#).

15.122.3.12 SetNumDecompressionThreads()

```
void SetNumDecompressionThreads (
    unsigned int numThreads ) [virtual]
```

Sets the default number of threads used for image decompression during [Convert\(\)](#).

The number of threads used is defaulted to be equal to one less than the number of concurrent threads supported by the system.

Parameters

<i>numThreads</i>	Number of parallel image decompression threads set to run

See also

[Convert\(\)](#)

Implements [IImageProcessor](#).

15.122.4 Friends And Related Function Documentation

15.122.4.1 ImageFiler

```
friend class ImageFiler [friend]
```

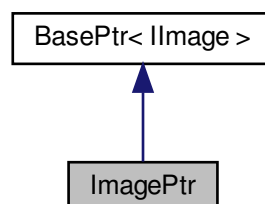
The documentation for this class was generated from the following file:

- [include/ImageProcessor.h](#)

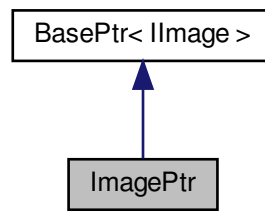
15.123 ImagePtr Class Reference

A reference tracked pointer to an image object.

Inheritance diagram for ImagePtr:



Collaboration diagram for ImagePtr:



Public Member Functions

- [ImagePtr](#) ()
Default constructor.
- [ImagePtr](#) (const int)
Default constructor with argument.
- [ImagePtr](#) (const long)
Default constructor with argument.
- [ImagePtr](#) (const std::nullptr_t)
Default constructor with argument.
- virtual [~ImagePtr](#) (void)
Virtual destructor.
- virtual [ImagePtr](#) & [operator=](#) (const [ImagePtr](#) &)
Assignment operator.

Additional Inherited Members

15.123.1 Detailed Description

A reference tracked pointer to an image object.

When the [ImagePtr](#) goes out of scope, it will trigger an auto release of the image from the stream.

15.123.2 Constructor & Destructor Documentation

15.123.2.1 ImagePtr() [1/4]

[ImagePtr](#) ()

Default constructor.

15.123.2.2 ImagePtr() [2/4]

```
ImagePtr (
    const int )
```

Default constructor with argument.

15.123.2.3 ImagePtr() [3/4]

```
ImagePtr (
    const long )
```

Default constructor with argument.

15.123.2.4 ImagePtr() [4/4]

```
ImagePtr (
    const std::nullptr_t )
```

Default constructor with argument.

15.123.2.5 ~ImagePtr()

```
virtual ~ImagePtr (
    void ) [virtual]
```

Virtual destructor.

15.123.3 Member Function Documentation

15.123.3.1 operator=()

```
virtual ImagePtr& operator= (
    const ImagePtr & ) [virtual]
```

Assignment operator.

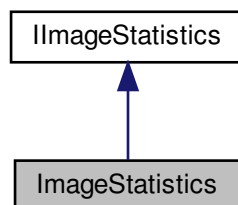
The documentation for this class was generated from the following file:

- [include/ImagePtr.h](#)

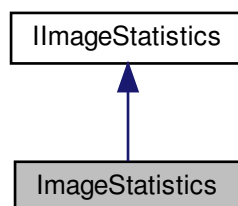
15.124 ImageStatistics Class Reference

Represents image statistics for an image.

Inheritance diagram for ImageStatistics:



Collaboration diagram for ImageStatistics:



Public Member Functions

- [ImageStatistics](#) ()
Default constructor.
- virtual [~ImageStatistics](#) ()
Default destructor.
- [ImageStatistics](#) (const [ImageStatistics](#) &other)
Copy constructor.
- [ImageStatistics](#) & [operator=](#) (const [ImageStatistics](#) &other)
Assignment operator.
- virtual void [EnableAll](#) ()
Enable all channels.
- virtual void [DisableAll](#) ()
Disable all channels.
- virtual void [EnableGreyOnly](#) ()

- Enable only the grey channel.*
 - virtual void [EnableRGBOOnly](#) ()
- Enable only the RGB channels.*
 - virtual void [EnableHSLOnly](#) ()
- Enable only the HSL channels.*
 - virtual void [GetChannelStatus](#) ([StatisticsChannel](#) channel, bool *pEnabled) const
- Gets the status of a statistics channel.*
 - virtual void [SetChannelStatus](#) ([StatisticsChannel](#) channel, bool enabled)
- Sets the status of a statistics channel.*
 - virtual void [GetRange](#) ([StatisticsChannel](#) channel, unsigned int *pMin, unsigned int *pMax) const
- Gets the range of a statistics channel.*
 - virtual void [GetPixelValueRange](#) ([StatisticsChannel](#) channel, unsigned int *pPixelValueMin, unsigned int *pPixelValueMax) const
- Gets the range of a statistics channel.*
 - virtual void [GetNumPixelValues](#) ([StatisticsChannel](#) channel, unsigned int *pNumPixelValues) const
- Gets the number of unique pixel values in the image.*
 - virtual void [GetMean](#) ([StatisticsChannel](#) channel, float *pPixelValueMean) const
- Gets the mean of the image.*
 - virtual void [GetHistogram](#) ([StatisticsChannel](#) channel, int **ppHistogram) const
- Gets the histogram for the image.*
 - virtual void [GetStatistics](#) ([StatisticsChannel](#) channel, unsigned int *pRangeMin=NULL, unsigned int *pRangeMax=NULL, unsigned int *pPixelValueMin=NULL, unsigned int *pPixelValueMax=NULL, unsigned int *pNumPixelValues=NULL, float *pPixelValueMean=NULL, int **ppHistogram=NULL) const
- Gets all statistics for the image.*

Friends

- class [ImageStatsCalculator](#)

Additional Inherited Members

15.124.1 Detailed Description

Represents image statistics for an image.

15.124.2 Constructor & Destructor Documentation

15.124.2.1 ImageStatistics() [1/2]

[ImageStatistics](#) ()

Default constructor.

15.124.2.2 ~ImageStatistics()

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

15.124.2.3 ImageStatistics() [2/2]

```
ImageStatistics (
    const ImageStatistics & other )
```

Copy constructor.

15.124.3 Member Function Documentation

15.124.3.1 DisableAll()

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements [IImageStatistics](#).

15.124.3.2 EnableAll()

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements [IImageStatistics](#).

15.124.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements [IImageStatistics](#).

15.124.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements [IImageStatistics](#).

15.124.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements [IImageStatistics](#).

15.124.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [virtual]
```

Gets the status of a statistics channel.

Parameters

<i>channel</i>	The statistics channel.
<i>pEnabled</i>	Whether the channel is enabled.

See also

[SetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

15.124.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [virtual]
```

Gets the histogram for the image.

Parameters

<i>channel</i>	The statistics channel.
<i>ppHistogram</i>	Pointer to an array containing the histogram.

Implements [IImageStatistics](#).

15.124.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [virtual]
```

Gets the mean of the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pPixelValueMean</i>	The mean of the image.

Implements [IImageStatistics](#).

15.124.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [virtual]
```

Gets the number of unique pixel values in the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pNumPixelValues</i>	The number of unique pixel values.

Implements [IImageStatistics](#).

15.124.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
```

```
unsigned int * pPixelValueMin,  
unsigned int * pPixelValueMax ) const [virtual]
```

Gets the range of a statistics channel.

The values returned are the maximum values recorded for all pixels in the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pPixelValueMin</i>	The minimum pixel value.
<i>pPixelValueMax</i>	The maximum pixel value.

Implements [IImageStatistics](#).

15.124.3.11 GetRange()

```
virtual void GetRange (  
    StatisticsChannel channel,  
    unsigned int * pMin,  
    unsigned int * pMax ) const [virtual]
```

Gets the range of a statistics channel.

The values returned are the maximum possible values for any given pixel in the image. This is generally 0-255 for 8 bit images, and 0-65535 for 16 bit images.

Parameters

<i>channel</i>	The statistics channel.
<i>pMin</i>	The minimum possible value.
<i>pMax</i>	The maximum possible value.

Implements [IImageStatistics](#).

15.124.3.12 GetStatistics()

```
virtual void GetStatistics (  
    StatisticsChannel channel,  
    unsigned int * pRangeMin = NULL,  
    unsigned int * pRangeMax = NULL,  
    unsigned int * pPixelValueMin = NULL,  
    unsigned int * pPixelValueMax = NULL,  
    unsigned int * pNumPixelValues = NULL,  
    float * pPixelValueMean = NULL,  
    int ** ppHistogram = NULL ) const [virtual]
```

Gets all statistics for the image.

Parameters

<i>channel</i>	The statistics channel.
<i>pRangeMin</i>	The minimum possible value.
<i>pRangeMax</i>	The maximum possible value.
<i>pPixelValueMin</i>	The minimum pixel value.
<i>pPixelValueMax</i>	The maximum pixel value.
<i>pNumPixelValues</i>	The number of unique pixel values.
<i>pPixelValueMean</i>	The mean of the image.
<i>ppHistogram</i>	Pointer to an array containing the histogram.

Implements [IImageStatistics](#).

15.124.3.13 operator=()

```
ImageStatistics& operator= (
    const ImageStatistics & other )
```

Assignment operator.

Parameters

<i>other</i>	The ImageStatistics object to copy from.

15.124.3.14 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [virtual]
```

Sets the status of a statistics channel.

Parameters

<i>channel</i>	The statistics channel.
<i>enabled</i>	Whether the channel should be enabled.

See also

[GetChannelStatus\(\)](#)

Implements [IImageStatistics](#).

15.124.4 Friends And Related Function Documentation

15.124.4.1 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

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

- include/[ImageStatistics.h](#)

15.125 ImageUtility Class Reference

Static helper functions for the image object class.

Static Public Member Functions

- static [ImagePtr](#) [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static void [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=[SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE](#))
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max, [SourceDataRange](#) srcDataRange=[SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE](#))
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=[SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE](#))
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [SourceDataRange](#) srcDataRange=[SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE](#))
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, const double min, const double max, [SourceDataRange](#) srcDataRange=[SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE](#))
Computes a normalized image.

15.125.1 Detailed Description

Static helper functions for the image object class.

15.125.2 Member Function Documentation

15.125.2.1 CreateNormalized() [1/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
[static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The destination pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>destPixelFormat</i>	The desired pixel format for the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.125.2.2 CreateNormalized() [2/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
[static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The normalized image pixel format will be the same as the source image.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.125.2.3 CreateNormalized() [3/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
[static]
```

Computes a normalized image.

The full range of the destination pixel format data type will be used as the min and max range for normalization. The destination pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destPixelFormat</i>	The desired pixel format for the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

Returns

The normalized image

15.125.2.4 CreateNormalized() [4/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
[static]
```

Computes a normalized image.

The min and max must be within range of the destination pixel format data type. The destination image must be initialized and have the same width and height as the source image. The destination image pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destImage</i>	The destination image in which to store the normalized image
<i>min</i>	The lower bound of the normalization range
<i>max</i>	The upper bound of the normalization range
<i>srcDataRange</i>	The desired option for the source data range to normalize from

15.125.2.5 CreateNormalized() [5/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
[static]
```

Computes a normalized image.

The full range of the destination pixel format data type will be used as the min and max range for normalization. The destination image must be initialized and have the same width and height as the source image. The destination image pixel format must be of the same data type as the source image pixel format.

Parameters

<i>srcImage</i>	The source image from which to create normalized image
<i>destImage</i>	The destination image in which to store the normalized image
<i>srcDataRange</i>	The desired option for the source data range to normalize from

15.125.2.6 CreateScaled() [1/2]

```
static void CreateScaled (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    ImageScalingAlgorithm scalingAlg,
    double scalingFactor ) [static]
```

Computes a scaled image using the specified parameters.

Does not support scaling of raw bayer images. The destination image height and width must be sufficient to store the calculated data. The destination image pixel format must be the same as the source image.

Parameters

<i>srcImage</i>	The source image from which to create scaled image
<i>destImage</i>	An image object in which to store the scaled data
<i>scalingAlg</i>	The desired image scaling algorithm to use
<i>scalingFactor</i>	The desired image scaling factor to use

15.125.2.7 CreateScaled() [2/2]

```
static ImagePtr CreateScaled (
    const ImagePtr & srcImage,
```

```
ImageScalingAlgorithm scalingAlg,
double scalingFactor ) [static]
```

Computes a scaled image using the specified parameters.

Does not support scaling of raw bayer images.

Parameters

<i>srcImage</i>	The source image from which to create scaled image
<i>scalingAlg</i>	The desired image scaling algorithm to use
<i>scalingFactor</i>	The desired image scaling factor to use

Returns

The scaled image

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

- [include/ImageUtility.h](#)

15.126 ImageUtilityCCM Class Reference

Static function to create color corrected images from an image object.

Static Public Member Functions

- static [ImagePtr CreateColorCorrected](#) (const [ImagePtr](#) &srcImage, const [CCMSettings](#) &settings)
Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.
- static void [CreateColorCorrected](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, const [CCMSettings](#) &settings)
Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

15.126.1 Detailed Description

Static function to create color corrected images from an image object.

15.126.2 Member Function Documentation

15.126.2.1 CreateColorCorrected() [1/2]

```
static ImagePtr CreateColorCorrected (
    const ImagePtr & srcImage,
    const CCMSettings & settings ) [static]
```

Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

When using [ImageUtilityCCM](#), users are advised to disable CCM on the camera before capturing source images. This can be done through the camera node "ColorTransformationEnable".

Color correction is currently supported for the following pixel formats:

- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_BGRa16
- PixelFormat_RGBa16
- PixelFormat_RGB16 The output image will have the same pixel format as the source image.

Parameters

<i>srcImage</i>	The source image to which the CCM is applied
<i>settings</i>	Selected CCM settings including CCMColorTemperature, CCMTType, CCMSensor, etc

Returns

The color corrected image

See also

[CCMSettings](#)

15.126.2.2 CreateColorCorrected() [2/2]

```
static void CreateColorCorrected (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const CCMSettings & settings ) [static]
```

Create a color corrected image from the source image by applying a color correction matrix calibrated according to the settings specified.

When using [ImageUtilityCCM](#), users are advised to disable CCM on the camera before capturing source images. This can be done through the camera node "ColorTransformationEnable".

Color correction is currently supported for the following pixel formats:

- PixelFormat_BGR8
- PixelFormat_BGRa8
- PixelFormat_RGBa8
- PixelFormat_RGB8
- PixelFormat_BGR16
- PixelFormat_BGRa16
- PixelFormat_RGBa16
- PixelFormat_RGB16

The destination image height and width must be the same as the source image.

Parameters

<i>srcImage</i>	The source image to which the CCM is applied
<i>destImage</i>	The destination image in which to store the color corrected image
<i>settings</i>	Selected CCM settings including CCMColorTemperature, CCMTType, CCMSensor, etc

See also

[CCMSettings](#)

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

- [include/ImageUtilityCCM.h](#)

15.127 ImageUtilityHeatmap Class Reference

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

Static Public Member Functions

- static [ImagePtr](#) [CreateHeatmap](#) (const [ImagePtr](#) &srcImage)
Computes a heatmap image.
- static void [CreateHeatmap](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage)
Computes a heatmap image.
- static void [SetHeatmapColorGradient](#) (const [HeatmapColor](#) newLowColor, const [HeatmapColor](#) newHighColor)
Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_WHITE.
- static void [GetHeatmapColorGradient](#) ([HeatmapColor](#) &currentLowColor, [HeatmapColor](#) &currentHighColor)
Returns the current heatmap gradient color range.
- static void [SetHeatmapRange](#) (const unsigned int newLowValue, const unsigned int newHighValue)
Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.
- static void [GetHeatmapRange](#) (unsigned int &currentLowValue, unsigned int &currentHighValue)
Returns the current high and low values used in heatmap representations.

15.127.1 Detailed Description

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

15.127.2 Member Function Documentation

15.127.2.1 CreateHeatmap() [1/2]

```
static ImagePtr CreateHeatmap (  
    const ImagePtr & srcImage ) [static]
```

Computes a heatmap image.

A heatmap image reinterprets monochrome data by mapping the luminosity of each pixel to a color value defined in the heatmap color gradient. The created image can be modified by changing the color gradient and heatmap range from the accompanying functions. The source image is required to be Mono8 or Mono16 pixel format.

Parameters

<i>srcImage</i>	The source image from which to create the heatmap

See also

[SetHeatmapRange\(\)](#)

[SetHeatmapColorGradient\(\)](#)

Returns

The heatmap image

15.127.2.2 CreateHeatmap() [2/2]

```
static void CreateHeatmap (  
    const ImagePtr & srcImage,  
    ImagePtr & destImage ) [static]
```

Computes a heatmap image.

A heatmap image reinterprets monochrome data by mapping the luminosity of each pixel to a color value defined in the heatmap color gradient. The created image can be modified by changing the color gradient and heatmap range from the accompanying functions. The source image is required to be Mono8 or Mono16 pixel format. The destination is required to be initialized, RGB8 or RGB16 pixel format, and have the same width, height, x offset, and y offset as the source image.

Parameters

<i>srcImage</i>	The source image from which to create the heatmap
<i>destImage</i>	The destination image in which to store the created heatmap

See also

[SetHeatmapRange\(\)](#)[SetHeatmapColorGradient\(\)](#)**15.127.2.3 GetHeatmapColorGradient()**

```
static void GetHeatmapColorGradient (
    HeatmapColor & currentLowColor,
    HeatmapColor & currentHighColor ) [static]
```

Returns the current heatmap gradient color range.

Parameters

<i>currentLowColor</i>	Current color at which the gradient begins.
<i>currentHighColor</i>	Current color at which the gradient ends.

See also

[SetHeatmapColorGradient\(\)](#)**15.127.2.4 GetHeatmapRange()**

```
static void GetHeatmapRange (
    unsigned int & currentLowValue,
    unsigned int & currentHighValue ) [static]
```

Returns the current high and low values used in heatmap representations.

Parameters

<i>currentLowValue</i>	Current value at which color representation begins.
<i>currentHighValue</i>	Current value at which color representation ends.

See also

[SetHeatmapRange\(\)](#)

15.127.2.5 SetHeatmapColorGradient()

```
static void SetHeatmapColorGradient (
    const HeatmapColor newLowColor,
    const HeatmapColor newHighColor ) [static]
```

Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_WHITE.

Parameters

<i>newLowColor</i>	New color at which to begin the gradient.
<i>newHighColor</i>	New color at which to end the gradient.

15.127.2.6 SetHeatmapRange()

```
static void SetHeatmapRange (
    const unsigned int newLowValue,
    const unsigned int newHighValue ) [static]
```

Sets the high and low values used to determine which grayscale values are converted to a color 'heatmap' representation.

Acceptable values range from 0 to 100.

Parameters

<i>newLowValue</i>	New value at which to begin color representation.
<i>newHighValue</i>	New value at which to end color representation.

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

- [include/ImageUtilityHeatmap.h](#)

15.128 ImageUtilityPolarization Class Reference

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

Static Public Member Functions

- static [ImagePtr](#) [ExtractPolarQuadrant](#) (const [ImagePtr](#) &srcImage, const [PolarizationQuadrant](#) desiredQuadrant)
Extracts all pixels of a specified degree of linear polarization into a new image object.
- static void [ExtractPolarQuadrant](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destQuadImage, const [PolarizationQuadrant](#) desiredQuadrant)

Extracts all pixels of a specified degree of linear polarization into the provided image object.

- static [ImagePtr CreateGlareReduced](#) (const [ImagePtr](#) &srcImage)

*Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant
The source image pixel format must be Polarized8 or BayerRGPolarized8.*

- static void [CreateGlareReduced](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destGlareReducedImage)

*Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant
The source image pixel format must be Polarized8 or BayerRGPolarized8.*

- static [ImagePtr CreateStokesS0](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the overall intensity of light from a polarized image.

- static void [CreateStokesS0](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS0Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the overall intensity of light from a polarized image.

- static [ImagePtr CreateStokesS1](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

- static void [CreateStokesS1](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS1Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

- static [ImagePtr CreateStokesS2](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.

- static void [CreateStokesS2](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS2Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the difference in intensity accepted through the polarizers.

- static [ImagePtr CreateDolp](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the fraction of incident light intensity in the linear polarization states.

- static void [CreateDolp](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destDolpImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the fraction of incident light intensity in the linear polarization states.

- static [ImagePtr CreateAolp](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

- static void [CreateAolp](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destAolpImg, const [ColorProcessingAlgorithm](#) colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

15.128.1 Detailed Description

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

15.128.2 Member Function Documentation

15.128.2.1 CreateAolp() [1/2]

```
static ImagePtr CreateAolp (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST
) [static]
```

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono32f or RGB32f respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The angle of linear polarization (aolp) image

15.128.2.2 CreateAolp() [2/2]

```
static void CreateAolp (
    const ImagePtr & srcImage,
    ImagePtr & destAolpImg,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST
) [static]
```

Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono32f or RGB32f respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destAolpImg</i>	The destination image in which to store the angle of linear polarization (aolp) image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.128.2.3 CreateDolp() [1/2]

```
static ImagePtr CreateDolp (
    const ImagePtr & srcImage,
```

```

        const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST
    ) [static]

```

Computes an image representing the fraction of incident light intensity in the linear polarization states.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono32f or RGB32f respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The degree of linear polarization (dolp) image

15.128.2.4 CreateDolp() [2/2]

```

static void CreateDolp (
    const ImagePtr & srcImage,
    ImagePtr & destDolpImage,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST
) [static]

```

Computes an image representing the fraction of incident light intensity in the linear polarization states.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono32f or RGB32f respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destDolpImage</i>	The destination image in which to store the degree of linear polarization (dolp) image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.128.2.5 CreateGlareReduced() [1/2]

```

static ImagePtr CreateGlareReduced (
    const ImagePtr & srcImage ) [static]

```

Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant. The source image pixel format must be Polarized8 or BayerRGPolarized8.

The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data

Returns

The reduced glare image

15.128.2.6 CreateGlareReduced() [2/2]

```
static void CreateGlareReduced (
    const ImagePtr & srcImage,
    ImagePtr & destGlareReducedImage ) [static]
```

Create a glare reduced image from the source image by choosing the darkest pixel from each polarization quadrant. The source image pixel format must be Polarized8 or BayerRGPolarized8.

The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to apply glare reduction
<i>destGlareReducedImage</i>	The destination image in which to store the image with reduced glare

15.128.2.7 CreateStokesS0() [1/2]

```
static ImagePtr CreateStokesS0 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST
) [static]
```

Computes an image representing the overall intensity of light from a polarized image.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S0 image

15.128.2.8 CreateStokesS0() [2/2]

```
static void CreateStokesS0 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS0Image,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER\_COLOR\_PROCESSING\_ALGORITHM\_NEAREST
) [static]
```

Computes an image representing the overall intensity of light from a polarized image.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS0Image</i>	The destination image in which to store the Stokes' S0 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.128.2.9 CreateStokesS1() [1/2]

```
static ImagePtr CreateStokesS1 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER\_COLOR\_PROCESSING\_ALGORITHM\_NEAREST
) [static]
```

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S1 image

15.128.2.10 CreateStokesS1() [2/2]

```
static void CreateStokesS1 (
    const ImagePtr & srcImage,
```

```

    ImagePtr & destStokesS1Image,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_
) [static]

```

Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS1Image</i>	The destination image in which to store the Stokes' S1 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.128.2.11 CreateStokesS2() [1/2]

```

static ImagePtr CreateStokesS2 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_
) [static]

```

Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono16s or RGB16s respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

Returns

The Stokes' S2 image

15.128.2.12 CreateStokesS2() [2/2]

```

static void CreateStokesS2 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS2Image,
    const ColorProcessingAlgorithm colorProcessingAlg = SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_
) [static]

```

Computes an image representing the difference in intensity accepted through the polarizers.

at 45 and -45 to the horizontal. The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono16s or RGB16s respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destStokesS2Image</i>	The destination image in which to store the Stokes' S2 image
<i>colorProcessingAlg</i>	The color processing algorithm to use for color images

15.128.2.13 ExtractPolarQuadrant() [1/2]

```
static ImagePtr ExtractPolarQuadrant (
    const ImagePtr & srcImage,
    const PolarizationQuadrant desiredQuadrant ) [static]
```

Extracts all pixels of a specified degree of linear polarization into a new image object.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format will be Mono8 or BayerRG8 respectively. The destination image height and width will be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>desiredQuadrant</i>	The polarization quadrant to extract

Returns

The specified polarization quadrant image

15.128.2.14 ExtractPolarQuadrant() [2/2]

```
static void ExtractPolarQuadrant (
    const ImagePtr & srcImage,
    ImagePtr & destQuadImage,
    const PolarizationQuadrant desiredQuadrant ) [static]
```

Extracts all pixels of a specified degree of linear polarization into the provided image object.

The source image pixel format must be Polarized8 or BayerRGPolarized8. The destination image pixel format must be Mono8 or BayerRG8 respectively. The destination image height and width must be half of the source image.

Parameters

<i>srcImage</i>	The source image from which to extract polarization data
<i>destQuadImage</i>	The destination image in which to store the extracted polarization quadrant
<i>desiredQuadrant</i>	The polarization quadrant to extract

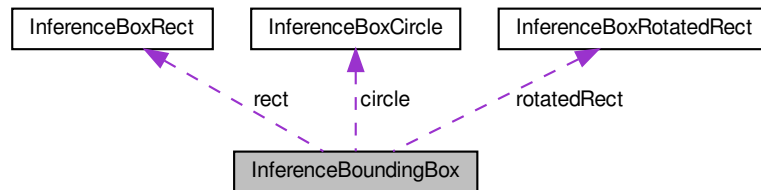
The documentation for this class was generated from the following file:

- [include/ImageUtilityPolarization.h](#)

15.129 InferenceBoundingBox Struct Reference

Inference Bounding Boxes data structure.

Collaboration diagram for InferenceBoundingBox:



Public Attributes

- [InferenceBoxType](#) `boxType`
- `int16_t` `classId`
- `float32_t` `confidence`
- [InferenceBoxRect](#) `rect`
- [InferenceBoxCircle](#) `circle`
- [InferenceBoxRotatedRect](#) `rotatedRect`

15.129.1 Detailed Description

Inference Bounding Boxes data structure.

15.129.2 Member Data Documentation

15.129.2.1 `boxType`

[InferenceBoxType](#) `boxType`

15.129.2.2 `circle`

[InferenceBoxCircle](#) `circle`

15.129.2.3 classId

```
int16_t classId
```

15.129.2.4 confidence

```
float32_t confidence
```

15.129.2.5 rect

```
InferenceBoxRect rect
```

15.129.2.6 rotatedRect

```
InferenceBoxRotatedRect rotatedRect
```

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

- include/ChunkDataInference.h

15.130 InferenceBoundingBoxResult Class Reference

An inference bounding boxes object which holds information about the detected bounding boxes.

Public Member Functions

- [InferenceBoundingBoxResult](#) ()
Default Constructor.
- [~InferenceBoundingBoxResult](#) ()
Destructor.
- [InferenceBoundingBoxResult](#) (const uint8_t *data, const int64_t lengthInBytes)
Default Constructor with arguments.
- [InferenceBoundingBoxResult](#) (const [InferenceBoundingBoxResult](#) &other)
Copy Constructor.
- [InferenceBoundingBoxResult](#) & [operator=](#) (const [InferenceBoundingBoxResult](#) &rhs)
Assignment Operator.
- int8_t [GetVersion](#) () const
Returns the bounding box format version number.
- int16_t [GetBoxCount](#) () const
Returns the number of bounding boxes.
- int8_t [GetBoxSize](#) () const
Returns the number of bytes allocated for one bounding box.
- [InferenceBoundingBox](#) [GetBoxAt](#) (const uint16_t index) const
Returns the bounding box at specified index.

Friends

- class [ChunkDataImpl](#)

15.130.1 Detailed Description

An inference bounding boxes object which holds information about the detected bounding boxes.

15.130.2 Constructor & Destructor Documentation

15.130.2.1 InferenceBoundingBoxResult() [1/3]

```
InferenceBoundingBoxResult ( )
```

Default Constructor.

15.130.2.2 ~InferenceBoundingBoxResult()

```
~InferenceBoundingBoxResult ( )
```

Destructor.

15.130.2.3 InferenceBoundingBoxResult() [2/3]

```
InferenceBoundingBoxResult (
    const uint8_t * data,
    const int64_t lengthInBytes )
```

Default Constructor with arguments.

Parameters

<i>data</i>	The bounding box binary data from chunk data.
<i>lengthInBytes</i>	The length of bounding box binary data in bytes.

15.130.2.4 InferenceBoundingBoxResult() [3/3]

```
InferenceBoundingBoxResult (
```

```
const InferenceBoundingBoxResult & other )
```

Copy Constructor.

15.130.3 Member Function Documentation

15.130.3.1 GetBoxAt()

```
InferenceBoundingBox GetBoxAt (
    const uint16_t index ) const
```

Returns the bounding box at specified index.

Parameters

<i>index</i>	Index of the bounding box to return.

15.130.3.2 GetBoxCount()

```
int16_t GetBoxCount ( ) const
```

Returns the number of bounding boxes.

15.130.3.3 GetBoxSize()

```
int8_t GetBoxSize ( ) const
```

Returns the number of bytes allocated for one bounding box.

15.130.3.4 GetVersion()

```
int8_t GetVersion ( ) const
```

Returns the bounding box format version number.

15.130.3.5 operator=()

```
InferenceBoundingBoxResult& operator= (
    const InferenceBoundingBoxResult & rhs )
```

Assignment Operator.

15.130.4 Friends And Related Function Documentation

15.130.4.1 ChunkDataImpl

```
friend class ChunkDataImpl [friend]
```

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

- [include/ChunkDataInference.h](#)

15.131 InferenceBoxCircle Struct Reference

Public Attributes

- `int16_t` [centerXCoord](#)
- `int16_t` [centerYCoord](#)
- `int16_t` [radius](#)

15.131.1 Member Data Documentation

15.131.1.1 centerXCoord

```
int16_t centerXCoord
```

15.131.1.2 centerYCoord

```
int16_t centerYCoord
```

15.131.1.3 radius

```
int16_t radius
```

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

- include/[ChunkDataInference.h](#)

15.132 InferenceBoxRect Struct Reference

Inference Bounding Box Type Data Structures.

Public Attributes

- int16_t [topLeftXCoord](#)
- int16_t [topLeftYCoord](#)
- int16_t [bottomRightXCoord](#)
- int16_t [bottomRightYCoord](#)

15.132.1 Detailed Description

Inference Bounding Box Type Data Structures.

15.132.2 Member Data Documentation

15.132.2.1 bottomRightXCoord

```
int16_t bottomRightXCoord
```

15.132.2.2 bottomRightYCoord

```
int16_t bottomRightYCoord
```

15.132.2.3 topLeftXCoord

```
int16_t topLeftXCoord
```

15.132.2.4 topLeftYCoord

```
int16_t topLeftYCoord
```

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

- [include/ChunkDataInference.h](#)

15.133 InferenceBoxRotatedRect Struct Reference

Public Attributes

- int16_t [topLeftXCoord](#)
- int16_t [topLeftYCoord](#)
- int16_t [bottomRightXCoord](#)
- int16_t [bottomRightYCoord](#)
- short [rotationAngle](#)

15.133.1 Member Data Documentation

15.133.1.1 bottomRightXCoord

```
int16_t bottomRightXCoord
```

15.133.1.2 bottomRightYCoord

```
int16_t bottomRightYCoord
```

15.133.1.3 rotationAngle

```
short rotationAngle
```

15.133.1.4 topLeftXCoord

```
int16_t topLeftXCoord
```

15.133.1.5 topLeftYCoord

```
int16_t topLeftYCoord
```

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

- include/[ChunkDataInference.h](#)

15.134 int64_autovector_t Class Reference

Vector of integers with reference counting.

Public Member Functions

- [int64_autovector_t](#) ()
- [int64_autovector_t](#) (const [int64_autovector_t](#) &obj)
- [int64_autovector_t](#) (size_t n)
- virtual [~int64_autovector_t](#) (void)
- [int64_autovector_t](#) & [operator=](#) (const [int64_autovector_t](#) &obj)
- void [operator delete](#) (void *pWhere)
- void * [operator new](#) (size_t uiSize)
- [int64_t](#) & [operator\[\]](#) (size_t uiIndex)
- const [int64_t](#) & [operator\[\]](#) (size_t uiIndex) const
- size_t [size](#) () const

Protected Attributes

- std::vector< [int64_t](#) > * [_pv](#)
- ATOMIC_VARIABLE * [_pCount](#)

15.134.1 Detailed Description

Vector of integers with reference counting.

15.134.2 Constructor & Destructor Documentation

15.134.2.1 int64_autovector_t() [1/3]

```
int64\_autovector\_t ( )
```

15.134.2.2 int64_autovector_t() [2/3]

```
int64_autovector_t (
    const int64_autovector_t & obj )
```

15.134.2.3 int64_autovector_t() [3/3]

```
int64_autovector_t (
    size_t n ) [explicit]
```

15.134.2.4 ~int64_autovector_t()

```
virtual ~int64_autovector_t (
    void ) [virtual]
```

15.134.3 Member Function Documentation

15.134.3.1 operator delete()

```
void operator delete (
    void * pWhere )
```

15.134.3.2 operator new()

```
void* operator new (
    size_t uiSize )
```

15.134.3.3 operator=()

```
int64_autovector_t& operator= (
    const int64_autovector_t & obj )
```


15.134.3.4 operator[]() [1/2]

```
int64_t& operator[] (
    size_t uiIndex )
```

15.134.3.5 operator[]() [2/2]

```
const int64_t& operator[] (
    size_t uiIndex ) const
```

15.134.3.6 size()

```
size_t size ( ) const
```

15.134.4 Member Data Documentation

15.134.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

15.134.4.2 _pv

```
std::vector<int64_t>* _pv [protected]
```

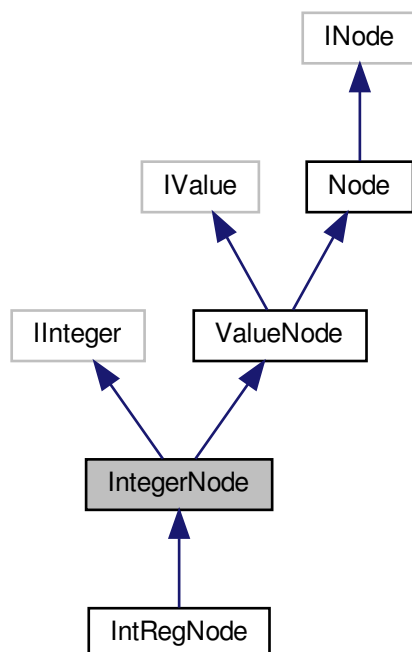
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Autovector.h](#)

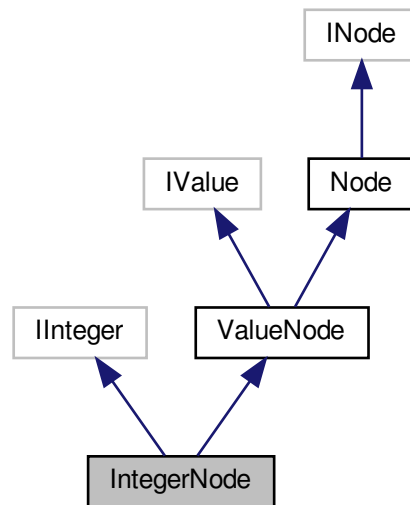
15.135 IntegerNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntegerNode:



Collaboration diagram for IntegerNode:



Public Member Functions

- [IntegerNode](#) ()
- [IntegerNode](#) (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual [~IntegerNode](#) ()
- virtual void [SetValue](#) (int64_t Value, bool [Verify](#)=true)
 - Set node value.*
- virtual [IInteger](#) & [operator=](#) (int64_t Value)
 - Set node value.*
- virtual int64_t [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
 - Get node value.*
- virtual int64_t [operator\(\)](#) ()
 - Get node value.*
- virtual int64_t [operator*](#) ()
 - Get node value.*
- virtual int64_t [GetMin](#) ()
 - Get minimum value allowed.*
- virtual int64_t [GetMax](#) ()
 - Get maximum value allowed.*
- virtual EIncMode [GetIncMode](#) ()
 - Get increment mode.*
- virtual int64_t [GetInc](#) ()
 - Get increment.*
- virtual [int64_t](#) [autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
 - Get list of valid value.*
- virtual ERepresentation [GetRepresentation](#) ()
 - Get recommended representation.*

- virtual [GenlCam::gcstring GetUnit](#) ()
Get the physical unit name.
- virtual [IFloat * GetFloatAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (int64_t Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)
Restrict maximum value.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Integer

Additional Inherited Members

15.135.1 Detailed Description

[Interface](#) for string properties.

15.135.2 Constructor & Destructor Documentation

15.135.2.1 IntegerNode() [1/2]

[IntegerNode](#) ()

15.135.2.2 IntegerNode() [2/2]

[IntegerNode](#) (
 std::shared_ptr< Node::NodeImpl > pInteger)

15.135.2.3 ~IntegerNode()

virtual ~[IntegerNode](#) () [virtual]

15.135.3 Member Function Documentation

15.135.3.1 GetFloatAlias()

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

15.135.3.2 GetInc()

```
virtual int64_t GetInc ( ) [virtual]
```

Get increment.

15.135.3.3 GetIncMode()

```
virtual EIncMode GetIncMode ( ) [virtual]
```

Get increment mode.

15.135.3.4 GetListOfValidValues()

```
virtual int64_autovector_t GetListOfValidValues (
    bool bounded = true ) [virtual]
```

Get list of valid value.

15.135.3.5 GetMax()

```
virtual int64_t GetMax ( ) [virtual]
```

Get maximum value allowed.

15.135.3.6 GetMin()

```
virtual int64_t GetMin ( ) [virtual]
```

Get minimum value allowed.

15.135.3.7 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

15.135.3.8 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) [virtual]
```

Get the physical unit name.

15.135.3.9 GetValue()

```
virtual int64_t GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.135.3.10 ImposeMax()

```
virtual void ImposeMax (
    int64_t Value ) [virtual]
```

Restrict maximum value.

15.135.3.11 ImposeMin()

```
virtual void ImposeMin (
    int64_t Value ) [virtual]
```

Restrict minimum value.

15.135.3.12 operator>()()

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

15.135.3.13 operator*()

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

15.135.3.14 operator=()

```
virtual Integer& operator= (
    int64_t Value ) [virtual]
```

Set node value.

15.135.3.15 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

15.135.3.16 SetValue()

```
virtual void SetValue (
    int64_t Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

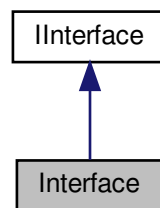
The documentation for this class was generated from the following file:

- include/SpinGenApi/[IntegerNode.h](#)

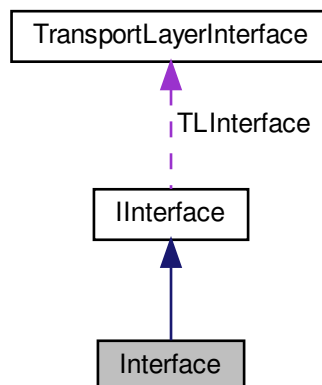
15.136 Interface Class Reference

An interface object which holds a list of cameras.

Inheritance diagram for Interface:



Collaboration diagram for Interface:



Public Member Functions

- virtual [~Interface](#) (void)
Virtual Destructor.
- [CameraList GetCameras](#) (bool updateCameras=true) const

- Returns a list of cameras available on this interface.*

 - bool [UpdateCameras](#) ()

Updates the list of cameras on this interface.
- [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const

Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.
- void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister)

Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.
- void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)

Unregisters an event handler for the interface.
- bool [IsInUse](#) () const

Checks if the interface is in use by any camera objects.
- void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL) const

Broadcast an Action Command to all devices on interface.
- bool [IsValid](#) ()

IsValid Checks a flag to determine if interface is still valid for use.

Friends

- class [InterfaceInternal](#)

Additional Inherited Members

15.136.1 Detailed Description

An interface object which holds a list of cameras.

15.136.2 Constructor & Destructor Documentation

15.136.2.1 ~Interface()

```
virtual ~Interface (
    void ) [virtual]
```

Virtual Destructor.

15.136.3 Member Function Documentation

15.136.3.1 GetCameras()

```
CameraList GetCameras (
    bool updateCameras = true ) const [virtual]
```

Returns a list of cameras available on this interface.

This call returns either usb3 vision or gige vision cameras depending on the underlying transport layer of this interface. The camera list object will reference count the cameras that it holds. It is important that the [CameraList](#) is destroyed or is cleared before [System::ReleaseInstance\(\)](#) can be called or an [InterfaceList](#) that holds this interface can be cleared.

See also

[System::ReleaseInstance\(\)](#)
[InterfaceList::Clear\(\)](#)
[CameraList::Clear\(\)](#)

Parameters

<i>updateCameras</i>	A flag used to issue an updateCameras() call internally before getting the camera list

Returns

An [CameraList](#) object that contains a list of cameras on this interface.

Implements [IInterface](#).

15.136.3.2 GetTLNodeMap()

```
GenApi::INodeMap& GetTLNodeMap ( ) const [virtual]
```

Gets a nodeMap that is generated from a [GenICam](#) XML file for the GenTL interface Module.

Returns

A reference to a INodeMap object.

Implements [IInterface](#).

15.136.3.3 IsInUse()

```
bool IsInUse ( ) const [virtual]
```

Checks if the interface is in use by any camera objects.

Returns

Returns true if the interface is in use and false otherwise.

Implements [IInterface](#).

15.136.3.4 IsValid()

```
bool IsValid ( ) [virtual]
```

IsValid Checks a flag to determine if interface is still valid for use.

Returns

If interface is valid or not

Implements [IInterface](#).

15.136.3.5 RegisterEventHandler()

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

Registers an event handler for the interface Event handlers are automatically cleaned up when an interface is removed, and must be registered to interfaces as they arrive.

Note that GEV interfaces experience arrival/removal events when the IP information changes, similar to GEV cameras. Please refer to the EnumerationEvents example for recommended use.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the interface

Implements [IInterface](#).

15.136.3.6 SendActionCommand()

```
void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) const [virtual]
```

Broadcast an Action Command to all devices on interface.

Parameters

<i>deviceKey</i>	The Action Command's device key
<i>groupKey</i>	The Action Command's group key
<i>groupMask</i>	The Action Command's group mask
<i>actionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.
<i>pResultSize</i>	(Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted.
<i>results</i>	(Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Implements [IInterface](#).

15.136.3.7 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the interface.

See also

[InterfaceEventHandler](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the interface

Implements [IInterface](#).

15.136.3.8 UpdateCameras()

```
bool UpdateCameras ( ) [virtual]
```

Updates the list of cameras on this interface.

This function needs to be called before any cameras can be discovered using [GetCameras\(\)](#). [System::GetCameras\(\)](#) will automatically call this function for each interface it enumerates. If the list changed after the last time [System::GetCameras\(\)](#) or [UpdateCameras\(\)](#) was called then the return value will be true, otherwise it is false.

See also

[System::GetCameras\(\)](#)
[GetCameras\(\)](#)

Returns

true if cameras changed on interface and false otherwise.

Implements [IInterface](#).

15.136.4 Friends And Related Function Documentation

15.136.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

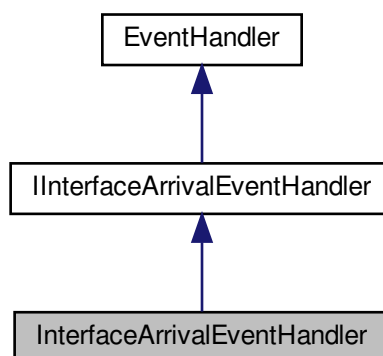
The documentation for this class was generated from the following file:

- [include/Interface.h](#)

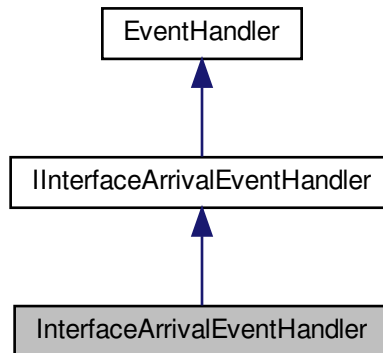
15.137 InterfaceArrivalEventHandler Class Reference

An event handler for capturing the interface arrival event.

Inheritance diagram for InterfaceArrivalEventHandler:



Collaboration diagram for InterfaceArrivalEventHandler:



Public Member Functions

- [InterfaceArrivalEventHandler](#) ()
Default constructor.
- virtual [~InterfaceArrivalEventHandler](#) ()
Virtual destructor.
- virtual void [OnInterfaceArrival](#) ([InterfacePtr](#) pInterface)=0
[Interface](#) arrival event callback.

Protected Member Functions

- [InterfaceArrivalEventHandler](#) & [operator=](#) (const [InterfaceArrivalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.137.1 Detailed Description

An event handler for capturing the interface arrival event.

Note that only GEV interface arrivals are currently handled.

15.137.2 Constructor & Destructor Documentation

15.137.2.1 InterfaceArrivalEventHandler()

```
InterfaceArrivalEventHandler ( )
```

Default constructor.

15.137.2.2 ~InterfaceArrivalEventHandler()

```
virtual ~InterfaceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

15.137.3 Member Function Documentation

15.137.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    InterfacePtr pInterface ) [pure virtual]
```

[Interface](#) arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter attached to the system

Implements [IInterfaceArrivalEventHandler](#).

15.137.3.2 operator=()

```
InterfaceArrivalEventHandler& operator= (
    const InterfaceArrivalEventHandler & ) [protected]
```

Assignment operator.

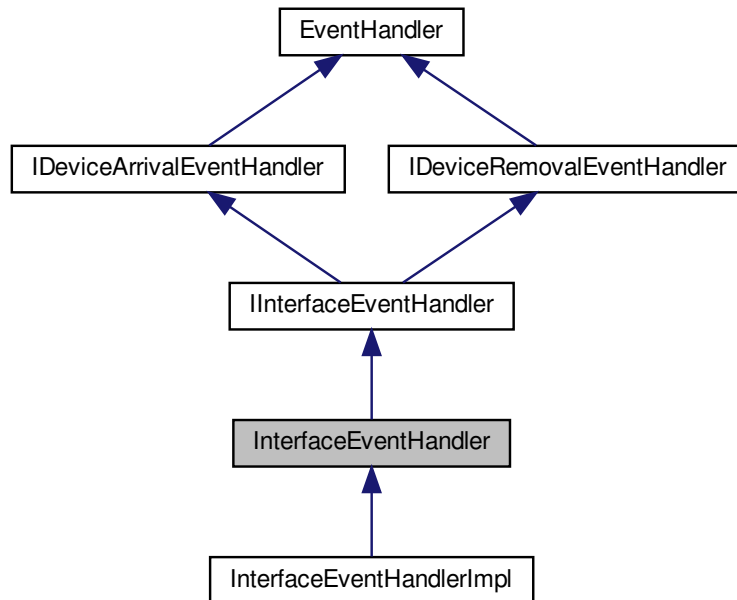
The documentation for this class was generated from the following file:

- include/[InterfaceArrivalEventHandler.h](#)

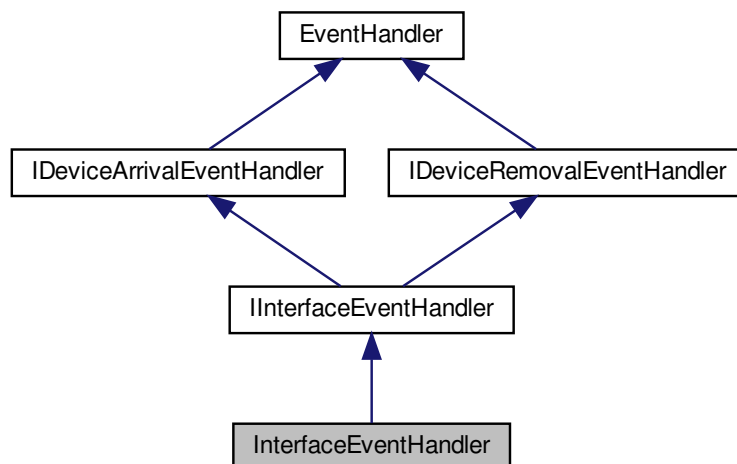
15.138 InterfaceEventHandler Class Reference

A handler to device arrival and removal events on all interfaces.

Inheritance diagram for InterfaceEventHandler:



Collaboration diagram for InterfaceEventHandler:



Public Member Functions

- [InterfaceEventHandler](#) ()
Default constructor.
- virtual [~InterfaceEventHandler](#) ()
Virtual destructor.
- virtual void [OnDeviceArrival](#) ([CameraPtr](#) pCamera)=0
Device arrival event callback.
- virtual void [OnDeviceRemoval](#) ([CameraPtr](#) pCamera)=0
Callback to the device removal event.

Protected Member Functions

- [InterfaceEventHandler](#) & [operator=](#) (const [InterfaceEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.138.1 Detailed Description

A handler to device arrival and removal events on all interfaces.

15.138.2 Constructor & Destructor Documentation

15.138.2.1 InterfaceEventHandler()

```
InterfaceEventHandler ( )
```

Default constructor.

15.138.2.2 ~InterfaceEventHandler()

```
virtual ~InterfaceEventHandler ( ) [virtual]
```

Virtual destructor.

15.138.3 Member Function Documentation

15.138.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (  
    CameraPtr pCamera ) [pure virtual]
```

Device arrival event callback.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device attached to the system

Implements [InterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.138.3.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    CameraPtr pCamera ) [pure virtual]
```

Callback to the device removal event.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device removed from the system

Implements [InterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

15.138.3.3 operator=()

```
InterfaceEventHandler& operator= (
    const InterfaceEventHandler & ) [protected]
```

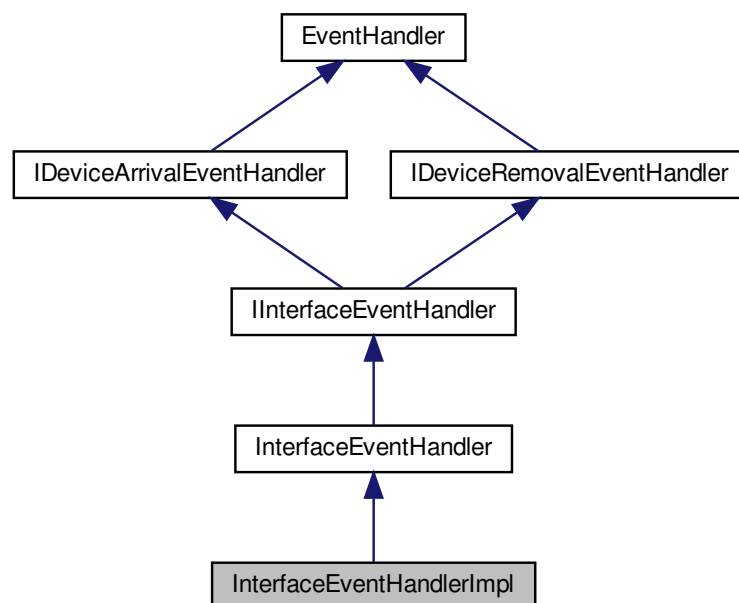
Assignment operator.

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

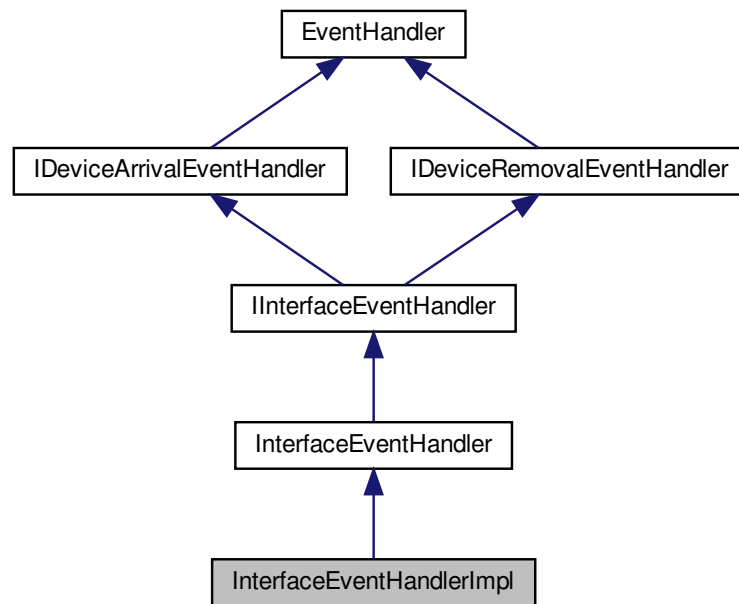
- [include/InterfaceEventHandler.h](#)

15.139 InterfaceEventHandlerImpl Class Reference

Inheritance diagram for InterfaceEventHandlerImpl:



Collaboration diagram for InterfaceEventHandlerImpl:



Public Member Functions

- [InterfaceEventHandlerImpl](#) ([SystemPtr](#) system)
- [~InterfaceEventHandlerImpl](#) ()
- void [OnDeviceArrival](#) ([CameraPtr](#) pCamera)
Device arrival event callback.
- void [OnDeviceRemoval](#) ([CameraPtr](#) pCamera)
Callback to the device removal event.
- [InterfaceEventHandlerImpl](#) ([SystemPtr](#) system)
- [InterfaceEventHandlerImpl](#) ([InterfacePtr](#) iface, std::string interfaceID)
- [~InterfaceEventHandlerImpl](#) ()
- void [PrintGenericHandlerMessage](#) (const unsigned long long deviceCount)
- void [OnDeviceArrival](#) ([CameraPtr](#) pCamera)
Device arrival event callback.
- void [OnDeviceRemoval](#) ([CameraPtr](#) pCamera)
Callback to the device removal event.
- std::string [GetInterfaceId](#) ()

Additional Inherited Members

15.139.1 Constructor & Destructor Documentation

15.139.1.1 InterfaceEventHandlerImpl() [1/3]

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

15.139.1.2 ~InterfaceEventHandlerImpl() [1/2]

```
~InterfaceEventHandlerImpl ( ) [inline]
```

15.139.1.3 InterfaceEventHandlerImpl() [2/3]

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

15.139.1.4 InterfaceEventHandlerImpl() [3/3]

```
InterfaceEventHandlerImpl (
    InterfacePtr iface,
    std::string interfaceID ) [inline]
```

15.139.1.5 ~InterfaceEventHandlerImpl() [2/2]

```
~InterfaceEventHandlerImpl ( ) [inline]
```

15.139.2 Member Function Documentation

15.139.2.1 GetInterfaceId()

```
std::string GetInterfaceId ( ) [inline]
```

15.139.2.2 OnDeviceArrival() [1/2]

```
void OnDeviceArrival (
    CameraPtr pCamera ) [inline], [virtual]
```

Device arrival event callback.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device attached to the system

Implements [InterfaceEventHandler](#).

15.139.2.3 OnDeviceArrival() [2/2]

```
void OnDeviceArrival (
    CameraPtr pCamera ) [inline], [virtual]
```

Device arrival event callback.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device attached to the system

Implements [InterfaceEventHandler](#).

15.139.2.4 OnDeviceRemoval() [1/2]

```
void OnDeviceRemoval (
    CameraPtr pCamera ) [inline], [virtual]
```

Callback to the device removal event.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device removed from the system

Implements [InterfaceEventHandler](#).

15.139.2.5 OnDeviceRemoval() [2/2]

```
void OnDeviceRemoval (
    CameraPtr pCamera ) [inline], [virtual]
```

Callback to the device removal event.

Parameters

<i>pCamera</i>	Reference tracked pointer to Camera object of the device removed from the system

Implements [InterfaceEventHandler](#).

15.139.2.6 PrintGenericHandlerMessage()

```
void PrintGenericHandlerMessage (
    const unsigned long long deviceCount ) [inline]
```

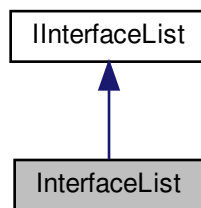
The documentation for this class was generated from the following files:

- [src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp](#)
- [src/EnumerationEvents/EnumerationEvents.cpp](#)

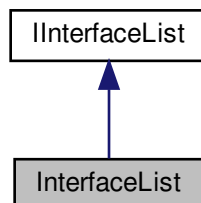
15.140 InterfaceList Class Reference

A list of the available interfaces on the system.

Inheritance diagram for InterfaceList:



Collaboration diagram for InterfaceList:



Public Member Functions

- [InterfaceList](#) (void)
- virtual [~InterfaceList](#) (void)
- [InterfaceList](#) (const [InterfaceList](#) &iface)
- [InterfaceList](#) & [operator=](#) (const [InterfaceList](#) &iface)
Assignment operator.
- [InterfacePtr](#) [operator\[\]](#) (unsigned int index)
Array subscription operators.
- unsigned int [GetSize](#) () const
Returns the size of the interface list.
- [InterfacePtr](#) [GetByIndex](#) (unsigned int index) const
Returns a pointer to an [Interface](#) object at the "index".
- [InterfacePtr](#) [GetByInterfaceID](#) (std::string interfaceID) const
Returns a pointer to a interface object with the specified interface identifier.
- void [Clear](#) ()
Clears the list of interfaces and destroys their corresponding objects.
- void [Add](#) ([InterfacePtr](#) iface)
Adds a copy of an interface object.
- void [Remove](#) ([InterfacePtr](#) iface)
Removes all occurrences of an interface that is pointed to by the input interface pointer and destroys its corresponding reference counted object.
- void [Append](#) (const [InterfaceList](#) *list)
Appends a copy of the interfact list.

Friends

- class [SystemImpl](#)
- class [ProducerImpl](#)

Additional Inherited Members

15.140.1 Detailed Description

A list of the available interfaces on the system.

15.140.2 Constructor & Destructor Documentation

15.140.2.1 [InterfaceList](#)() [1/2]

```
InterfaceList (
    void )
```


15.140.2.2 ~InterfaceList()

```
virtual ~InterfaceList (
    void ) [virtual]
```

15.140.2.3 InterfaceList() [2/2]

```
InterfaceList (
    const InterfaceList & iface )
```

15.140.3 Member Function Documentation

15.140.3.1 Add()

```
void Add (
    InterfacePtr iface ) [virtual]
```

Adds a copy of an interface object.

Parameters

<i>iface</i>	An interface object to be added to this list.

Implements [InterfaceList](#).

15.140.3.2 Append()

```
void Append (
    const InterfaceList * list ) [virtual]
```

Appends a copy of the interface list.

Parameters

<i>list</i>	Another InterfaceList object, whose elements are added to this list.

Implements [InterfaceList](#).

15.140.3.3 Clear()

```
void Clear ( ) [virtual]
```

Clears the list of interfaces and destroys their corresponding objects.

It is important to first make sure there are no referenced cameras still in use before calling [Clear\(\)](#). If a camera on any of the interfaces is still in use this function will throw an exception.

Implements [IInterfaceList](#).

15.140.3.4 GetByIndex()

```
InterfacePtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to an [Interface](#) object at the "index".

Parameters

<i>index</i>	The index at which to retrieve the Interface object

Returns

A pointer to an [Interface](#) object.

Implements [IInterfaceList](#).

15.140.3.5 GetByInterfaceID()

```
InterfacePtr GetByInterfaceID (
    std::string interfaceID ) const
```

Returns a pointer to a interface object with the specified interface identifier.

This function will return a NULL [InterfacePtr](#) if no matching interface identifier is found.

Parameters

<i>interfaceID</i>	The unique interface identifier of the interface object to retrieve

Returns

A pointer to an [Interface](#) object.

15.140.3.6 GetSize()

```
unsigned int GetSize ( ) const [virtual]
```

Returns the size of the interface list.

The size is the number of [Interface](#) objects stored in the list.

Returns

An integer that represents the list size.

Implements [IInterfaceList](#).

15.140.3.7 operator=()

```
IInterfaceList& operator= (
    const IInterfaceList & iface )
```

Assignment operator.

15.140.3.8 operator[]()

```
IInterfacePtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IInterfaceList](#).

15.140.3.9 Remove()

```
void Remove (
    IInterfacePtr iface ) [virtual]
```

Removes all occurrences of an interface that is pointed to by the input interface pointer and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with `SPINNAKER_ERR_NOT_AVAILABLE` error if no matching interface is found.

Parameters

<i>iface</i>	Pointer to the interface that is to be removed

Implements [InterfaceList](#).

15.140.4 Friends And Related Function Documentation

15.140.4.1 ProducerImpl

```
friend class ProducerImpl [friend]
```

15.140.4.2 SystemImpl

```
friend class SystemImpl [friend]
```

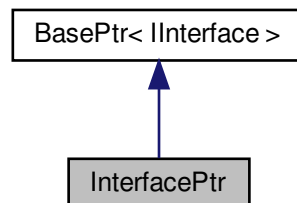
The documentation for this class was generated from the following file:

- include/[InterfaceList.h](#)

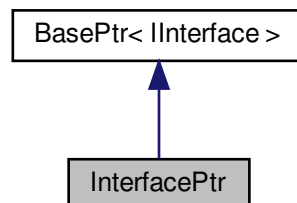
15.141 InterfacePtr Class Reference

A reference tracked pointer to the interface object.

Inheritance diagram for InterfacePtr:



Collaboration diagram for InterfacePtr:



Public Member Functions

- [InterfacePtr](#) () throw ()
Default Constructor.
- [InterfacePtr](#) (const int) throw ()
Default Constructor with argument.
- [InterfacePtr](#) (const long) throw ()
- [InterfacePtr](#) (const std::nullptr_t) throw ()

Additional Inherited Members

15.141.1 Detailed Description

A reference tracked pointer to the interface object.

15.141.2 Constructor & Destructor Documentation

15.141.2.1 [InterfacePtr\(\)](#) [1/4]

```
InterfacePtr ( ) throw ( )    [inline]
```

Default Constructor.

15.141.2.2 [InterfacePtr\(\)](#) [2/4]

```
InterfacePtr (  
    const int  ) throw ( )    [inline]
```

Default Constructor with argument.

15.141.2.3 [InterfacePtr\(\)](#) [3/4]

```
InterfacePtr (  
    const long  ) throw ( )    [inline]
```

15.141.2.4 InterfacePtr() [4/4]

```
InterfacePtr (
    const std::nullptr_t ) throw ( )    [inline]
```

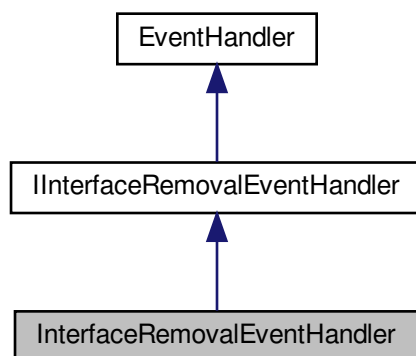
The documentation for this class was generated from the following file:

- include/[InterfacePtr.h](#)

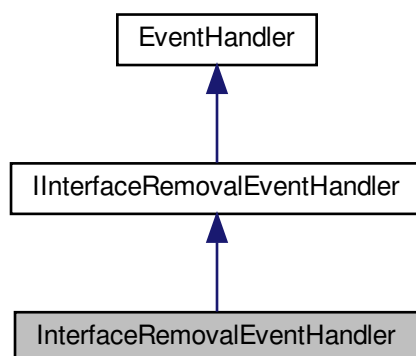
15.142 InterfaceRemovalEventHandler Class Reference

An event handler for capturing the interface removal event.

Inheritance diagram for InterfaceRemovalEventHandler:



Collaboration diagram for InterfaceRemovalEventHandler:



Public Member Functions

- [InterfaceRemovalEventHandler](#) ()
Default Constructor.
- virtual [~InterfaceRemovalEventHandler](#) ()
Virtual Destructor.
- virtual void [OnInterfaceRemoval](#) ([InterfacePtr](#) pInterface)=0
Interface removal event callback.

Protected Member Functions

- [InterfaceRemovalEventHandler](#) & [operator=](#) (const [InterfaceRemovalEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.142.1 Detailed Description

An event handler for capturing the interface removal event.

Note that only GEV interface removals are currently handled.

15.142.2 Constructor & Destructor Documentation

15.142.2.1 InterfaceRemovalEventHandler()

```
InterfaceRemovalEventHandler ( )
```

Default Constructor.

15.142.2.2 ~InterfaceRemovalEventHandler()

```
virtual ~InterfaceRemovalEventHandler ( ) [virtual]
```

Virtual Destructor.

15.142.3 Member Function Documentation

15.142.3.1 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (  
    InterfacePtr pInterface ) [pure virtual]
```

[Interface](#) removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter removed from the system

Implements [InterfaceRemovalEventHandler](#).

15.142.3.2 operator=()

```
InterfaceRemovalEventHandler& operator= (
    const InterfaceRemovalEventHandler & ) [protected]
```

Assignment operator.

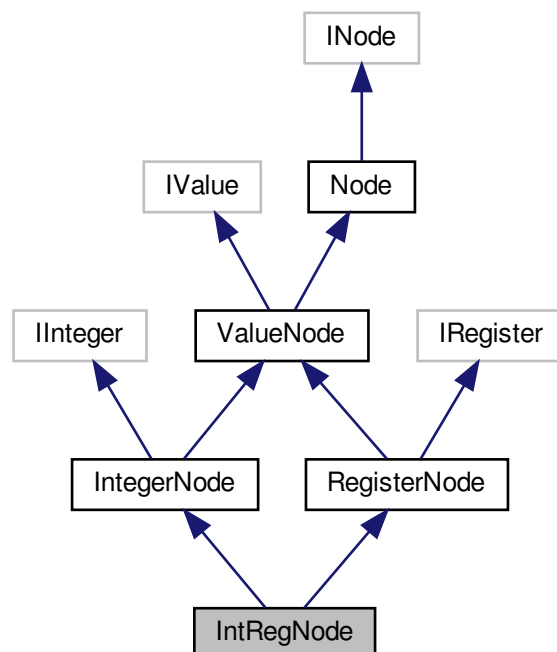
The documentation for this class was generated from the following file:

- include/[InterfaceRemovalEventHandler.h](#)

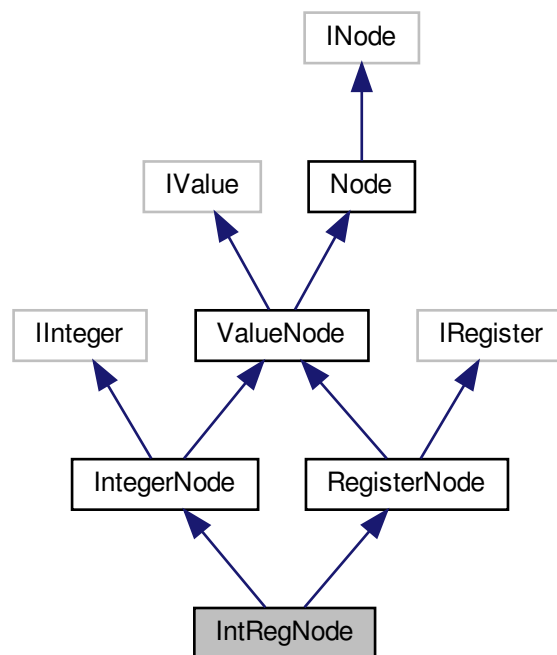
15.143 IntRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntRegNode:



Collaboration diagram for IntRegNode:



Public Member Functions

- [IntRegNode](#) ()
- [IntRegNode](#) (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual [~IntRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.143.1 Detailed Description

[Interface](#) for string properties.

15.143.2 Constructor & Destructor Documentation

15.143.2.1 IntRegNode() [1/2]

```
IntRegNode ( )
```

15.143.2.2 IntRegNode() [2/2]

```
IntRegNode (
    std::shared_ptr< Node::NodeImpl > pInteger )
```

15.143.2.3 ~IntRegNode()

```
virtual ~IntRegNode ( ) [virtual]
```

15.143.3 Member Function Documentation**15.143.3.1 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [IntegerNode](#).

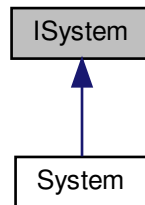
The documentation for this class was generated from the following file:

- include/SpinGenApi/[IntRegNode.h](#)

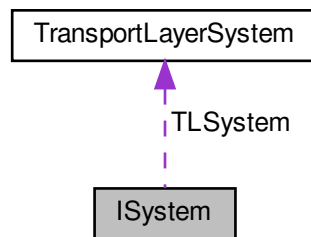
15.144 ISystem Class Reference

The interface file for [System](#).

Inheritance diagram for ISystem:



Collaboration diagram for ISystem:



Public Member Functions

- virtual [~ISystem](#) ()
- virtual void [ReleaseInstance](#) ()=0
- virtual [InterfaceList](#) [GetInterfaces](#) (bool updateInterface=true)=0
- virtual void [UpdateInterfaceList](#) ()=0
- virtual [CameraList](#) [GetCameras](#) (bool updateInterfaces=true, bool updateCameras=true)=0
- virtual bool [UpdateCameras](#) (bool updateInterfaces=true)=0
- virtual void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister, bool updateInterface=false)=0
- virtual void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)=0
- virtual void [RegisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)=0
- virtual void [UnregisterAllLoggingEventHandlers](#) ()=0
- virtual void [UnregisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)=0
- virtual void [SetLoggingEventPriorityLevel](#) ([SpinnakerLogLevel](#) level)=0
- virtual [SpinnakerLogLevel](#) [GetLoggingEventPriorityLevel](#) ()=0
- virtual bool [IsInUse](#) ()=0
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL)=0
- virtual const [LibraryVersion](#) [GetLibraryVersion](#) ()=0
- virtual [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const =0

Public Attributes

- [TransportLayerSystem](#) [TLSystem](#)

Protected Member Functions

- [ISystem](#) ()
- [ISystem](#) (const [ISystem](#) &)
- [ISystem](#) & [operator=](#) (const [ISystem](#) &)

Friends

- class [SystemPtrInternal](#)

15.144.1 Detailed Description

The interface file for [System](#).

15.144.2 Constructor & Destructor Documentation

15.144.2.1 ~ISystem()

```
virtual ~ISystem ( ) [inline], [virtual]
```

15.144.2.2 ISystem() [1/2]

```
ISystem ( ) [inline], [protected]
```

15.144.2.3 ISystem() [2/2]

```
ISystem (  
    const ISystem & ) [inline], [protected]
```

15.144.3 Member Function Documentation

15.144.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [pure virtual]
```

Implemented in [System](#).

15.144.3.2 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

15.144.3.3 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( ) [pure virtual]
```

Implemented in [System](#).

15.144.3.4 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [pure virtual]
```

Implemented in [System](#).

15.144.3.5 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [System](#).

15.144.3.6 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [System](#).

15.144.3.7 operator=()

```
ISystem& operator= (
    const ISystem & ) [protected]
```

15.144.3.8 RegisterEventHandler()

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = false ) [pure virtual]
```

Implemented in [System](#).

15.144.3.9 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

15.144.3.10 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in [System](#).

15.144.3.11 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [pure virtual]
```

Implemented in [System](#).

15.144.3.12 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [pure virtual]
```

Implemented in [System](#).

15.144.3.13 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [pure virtual]
```

Implemented in [System](#).

15.144.3.14 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [System](#).

15.144.3.15 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

15.144.3.16 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [pure virtual]
```

Implemented in [System](#).

15.144.3.17 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [pure virtual]
```

Implemented in [System](#).

15.144.4 Friends And Related Function Documentation

15.144.4.1 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

15.144.5 Member Data Documentation

15.144.5.1 TLSystem

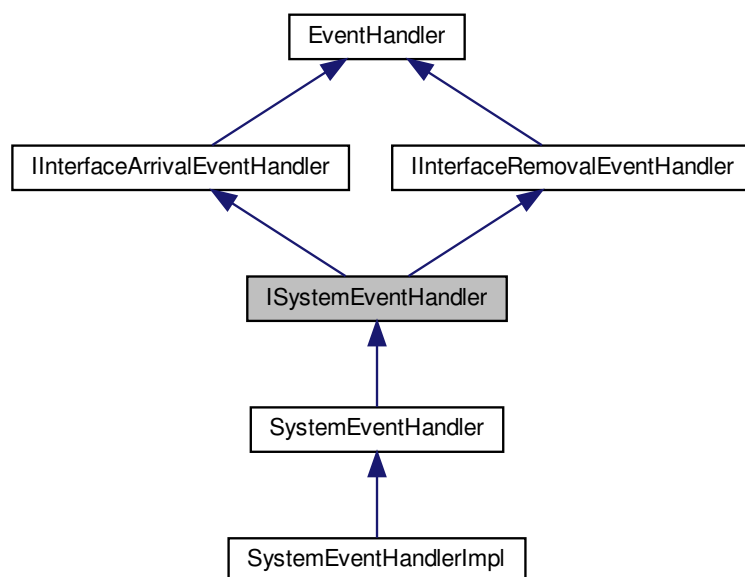
[TransportLayerSystem](#) TLSystem

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

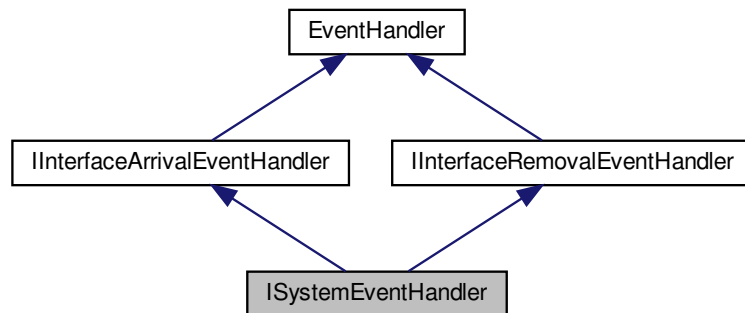
- [include/Interface/ISystem.h](#)

15.145 ISystemEventHandler Class Reference

Inheritance diagram for ISystemEventHandler:



Collaboration diagram for ISystemEventHandler:



Public Member Functions

- virtual `~ISystemEventHandler()`
- virtual void `OnInterfaceArrival (InterfacePtr pInterface)=0`
- virtual void `OnInterfaceRemoval (InterfacePtr pInterface)=0`

Protected Member Functions

- `ISystemEventHandler()`
- `ISystemEventHandler (const ISystemEventHandler &)`
- `ISystemEventHandler & operator= (const ISystemEventHandler &)`

Additional Inherited Members

15.145.1 Constructor & Destructor Documentation

15.145.1.1 ~ISystemEventHandler()

```
virtual ~ISystemEventHandler ( ) [inline], [virtual]
```

15.145.1.2 ISystemEventHandler() [1/2]

```
ISystemEventHandler ( ) [inline], [protected]
```

15.145.1.3 `ISystemEventHandler()` [2/2]

```
ISystemEventHandler (
    const ISystemEventHandler & ) [inline], [protected]
```

15.145.2 Member Function Documentation

15.145.2.1 `OnInterfaceArrival()`

```
virtual void OnInterfaceArrival (
    InterfacePtr pInterface ) [pure virtual]
```

Implements [IInterfaceArrivalEventHandler](#).

Implemented in [SystemEventHandler](#), and [SystemEventHandlerImpl](#).

15.145.2.2 `OnInterfaceRemoval()`

```
virtual void OnInterfaceRemoval (
    InterfacePtr pInterface ) [pure virtual]
```

Implements [IInterfaceRemovalEventHandler](#).

Implemented in [SystemEventHandler](#), and [SystemEventHandlerImpl](#).

15.145.2.3 `operator=()`

```
ISystemEventHandler& operator= (
    const ISystemEventHandler & ) [protected]
```

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

- [include/Interface/ISystemEventHandler.h](#)

15.146 JPEGOption Struct Reference

Options for saving JPEG image.

Public Member Functions

- [JPEGOption](#) ()

Public Attributes

- bool [progressive](#)
Whether to save as a progressive JPEG file.
- unsigned int [quality](#)
JPEG image quality in range (0-100).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.146.1 Detailed Description

Options for saving JPEG image.

15.146.2 Constructor & Destructor Documentation

15.146.2.1 JPEGOption()

```
JPEGOption ( ) [inline]
```

15.146.3 Member Data Documentation

15.146.3.1 progressive

```
bool progressive
```

Whether to save as a progressive JPEG file.

15.146.3.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

15.146.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

15.147 JPG2Option Struct Reference

Options for saving JPEG2000 image.

Public Member Functions

- [JPG2Option](#) ()

Public Attributes

- unsigned int [quality](#)
JPEG saving quality in range (1-512).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.147.1 Detailed Description

Options for saving JPEG2000 image.

15.147.2 Constructor & Destructor Documentation

15.147.2.1 JPG2Option()

```
JPG2Option ( ) [inline]
```

15.147.3 Member Data Documentation

15.147.3.1 quality

```
unsigned int quality
```

JPEG saving quality in range (1-512).

15.147.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

15.148 LibraryVersion Struct Reference

Provides easier access to the current version of [Spinnaker](#).

Public Attributes

- unsigned int [major](#)
Major version of the library.
- unsigned int [minor](#)
Minor version of the library.
- unsigned int [type](#)
Version type of the library.
- unsigned int [build](#)
Build number of the library.

15.148.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

15.148.2 Member Data Documentation

15.148.2.1 build

```
unsigned int build
```

Build number of the library.

15.148.2.2 major

```
unsigned int major
```

Major version of the library.

15.148.2.3 minor

```
unsigned int minor
```

Minor version of the library.

15.148.2.4 type

```
unsigned int type
```

Version type of the library.

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

- [include/SpinnakerDefs.h](#)

15.149 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

Public Member Functions

- [Lock](#) (const [LockableObject](#)< Object > &obj)
- [~Lock](#) ()

15.149.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >::Lock
```

A scopelevel [Lock](#) class.

Automatically acquires the lock when created and releases it when destroyed.

15.149.2 Constructor & Destructor Documentation

15.149.2.1 Lock()

```
Lock (
    const LockableObject< Object > & obj ) [inline]
```

15.149.2.2 ~Lock()

```
~Lock ( ) [inline]
```

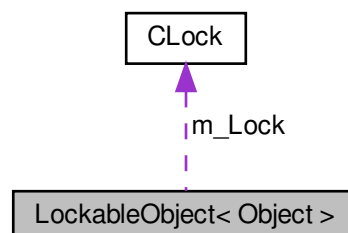
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

15.150 LockableObject< Object > Class Template Reference

Instance-Lock for an object.

Collaboration diagram for LockableObject< Object >:



Classes

- class [Lock](#)
A scopelevel [Lock](#) class.

Public Member Functions

- [Lock](#) [GetLock](#) () const
Get a new lock.

Public Attributes

- [CLOCK](#) [m_Lock](#)

Friends

- class [Lock](#)

15.150.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

15.150.2 Member Function Documentation

15.150.2.1 GetLock()

```
Lock GetLock ( ) const [inline]
```

Get a new lock.

15.150.3 Friends And Related Function Documentation

15.150.3.1 Lock

```
friend class Lock [friend]
```


15.150.4 Member Data Documentation

15.150.4.1 m_Lock

`CLock m_Lock [mutable]`

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

- `include/SpinGenApi/GCSynch.h`

15.151 LoggingEventData Class Reference

The `LoggingEventData` object.

Public Member Functions

- `~LoggingEventData ()`
Default Destructor.
- `const char * GetCategoryName ()`
Gets the logging event category name.
- `const char * GetLogMessage ()`
Gets the logging event message.
- `const char * GetNDC ()`
Gets the logging event's Nested Diagnostic Context (NDC).
- `const int GetPriority ()`
Gets the logging event priority.
- `const char * GetThreadName ()`
Gets the logging event thread name.
- `const char * GetTimestamp ()`
Gets the logging event time stamp.
- `const char * GetPriorityName ()`
Gets the logging event priority name.

Protected Member Functions

- `LoggingEventData (void *data)`
Default Constructor.

Friends

- class `SystemImpl`

15.151.1 Detailed Description

The [LoggingEventData](#) object.

15.151.2 Constructor & Destructor Documentation

15.151.2.1 ~LoggingEventData()

```
~LoggingEventData ( )
```

Default Destructor.

15.151.2.2 LoggingEventData()

```
LoggingEventData (
    void * data ) [protected]
```

Default Constructor.

15.151.3 Member Function Documentation

15.151.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

Returns

The category name

15.151.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

Returns

The log message

15.151.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

Returns

The log event's NDC

15.151.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

Returns

The log priority

15.151.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

Returns

The priority name of the log

15.151.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

Returns

The thread name

15.151.3.7 GetTimestamp()

```
const char* GetTimestamp ( )
```

Gets the logging event time stamp.

Returns

The time stamp of the log

15.151.4 Friends And Related Function Documentation

15.151.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

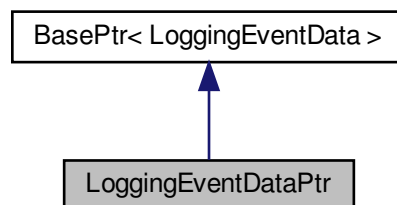
The documentation for this class was generated from the following file:

- include/[LoggingEventData.h](#)

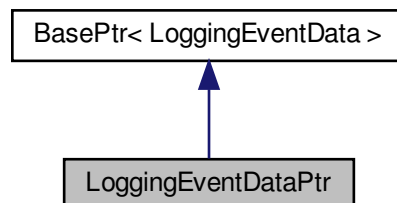
15.152 LoggingEventDataPtr Class Reference

A reference tracked pointer to the LoggingEvent object.

Inheritance diagram for LoggingEventDataPtr:



Collaboration diagram for LoggingEventDataPtr:



Public Member Functions

- [LoggingEventDataPtr](#) () throw ()
Default Constructor.
- [LoggingEventDataPtr](#) (const int) throw ()
Default Constructor with argument.
- [LoggingEventDataPtr](#) (const long) throw ()
Default Constructor with argument.
- [LoggingEventDataPtr](#) (const std::nullptr_t) throw ()
Default Constructor with argument.

Additional Inherited Members

15.152.1 Detailed Description

A reference tracked pointer to the LoggingEvent object.

15.152.2 Constructor & Destructor Documentation

15.152.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw ( )    [inline]
```

Default Constructor.

15.152.2.2 LoggingEventDataPtr() [2/4]

```
LoggingEventDataPtr (
    const int ) throw ( )    [inline]
```

Default Constructor with argument.

15.152.2.3 LoggingEventDataPtr() [3/4]

```
LoggingEventDataPtr (
    const long ) throw ( )    [inline]
```

Default Constructor with argument.

15.152.2.4 LoggingEventDataPtr() [4/4]

```
LoggingEventDataPtr (
    const std::nullptr_t ) throw ( )    [inline]
```

Default Constructor with argument.

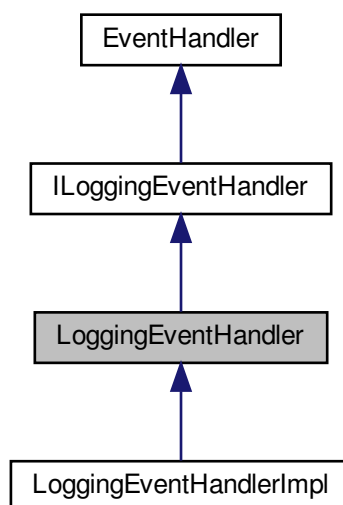
The documentation for this class was generated from the following file:

- [include/LoggingEventDataPtr.h](#)

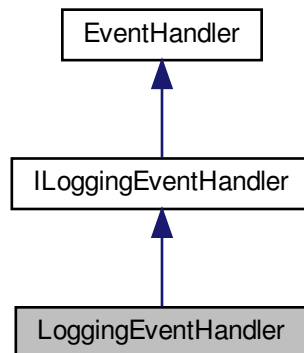
15.153 LoggingEventHandler Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEventHandler:



Collaboration diagram for LoggingEventHandler:



Public Member Functions

- [LoggingEventHandler \(\)](#)
Default constructor.
- [~LoggingEventHandler \(\)](#)
Virtual destructor.
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0
The callback for the log event.

Protected Member Functions

- [LoggingEventHandler & operator=](#) (const [LoggingEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.153.1 Detailed Description

An event handler for capturing the device logging event.

15.153.2 Constructor & Destructor Documentation

15.153.2.1 LoggingEventHandler()

```
LoggingEventHandler ( )
```

Default constructor.

15.153.2.2 ~LoggingEventHandler()

```
~LoggingEventHandler ( )
```

Virtual destructor.

15.153.3 Member Function Documentation

15.153.3.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

The callback for the log event.

Parameters

<i>eventPtr</i>	The logging event pointer

Implements [ILoggingEventHandler](#).

15.153.3.2 operator=()

```
LoggingEventHandler& operator= (
    const LoggingEventHandler & ) [protected]
```

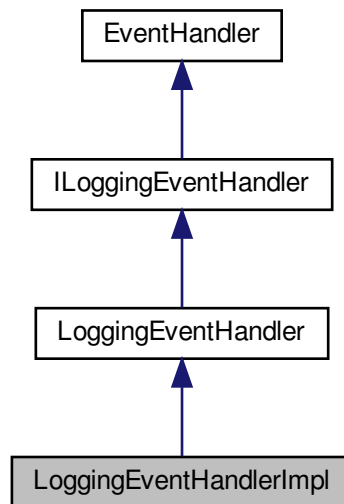
Assignment operator.

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

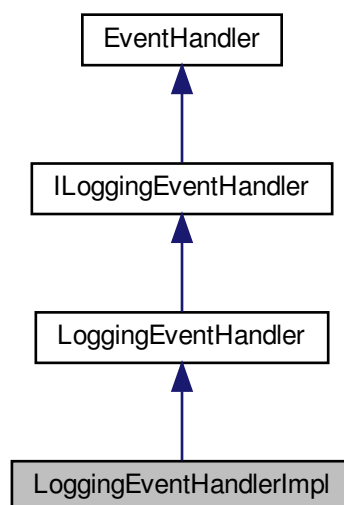
- include/[LoggingEventHandler.h](#)

15.154 LoggingEventHandlerImpl Class Reference

Inheritance diagram for LoggingEventHandlerImpl:



Collaboration diagram for LoggingEventHandlerImpl:



Additional Inherited Members

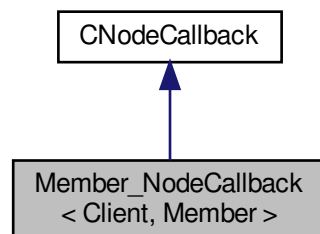
The documentation for this class was generated from the following file:

- src/Logging/[Logging.cpp](#)

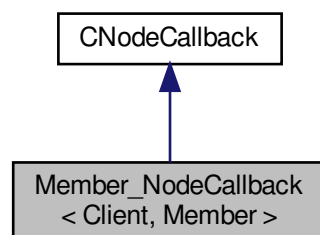
15.155 Member_NodeCallback< Client, Member > Class Template Reference

Container for a member function pointer.

Inheritance diagram for Member_NodeCallback< Client, Member >:



Collaboration diagram for Member_NodeCallback< Client, Member >:



Public Types

- typedef void(Client::* [PMEMBERFUNC](#)) (INode *)
Member function type.

Public Member Functions

- [Member_NodeCallback](#) (INode *pNode, Client &client, Member member, ECallbackType CallbackType)
Constructor.
- virtual void [operator\(\)](#) (ECallbackType CallbackType) const
execute operation
- virtual void [Destroy](#) ()
destroys the object

Additional Inherited Members

15.155.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

15.155.2 Member Typedef Documentation

15.155.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC (INode *))
```

Member function type.

15.155.3 Constructor & Destructor Documentation

15.155.3.1 Member_NodeCallback()

```
Member_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType ) [inline]
```

Constructor.

15.155.4 Member Function Documentation

15.155.4.1 Destroy()

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys the object

Implements [CNodeCallback](#).

15.155.4.2 operator>()

```
virtual void operator() (
    ECallbackType CallbackType ) const [inline], [virtual]
```

execute operation

Implements [CNodeCallback](#).

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

- include/SpinGenApi/[NodeCallback.h](#)

15.156 MJPGOption Struct Reference

Options for saving MJPG files.

Public Member Functions

- [MJPGOption](#) ()

Public Attributes

- float [frameRate](#)
Frame rate of the stream.
- unsigned int [quality](#)
Image quality (1-100)
- unsigned int [width](#)
Width of source image.
- unsigned int [height](#)
Height of source image.
- unsigned int [reserved](#) [192]

15.156.1 Detailed Description

Options for saving MJPG files.

15.156.2 Constructor & Destructor Documentation

15.156.2.1 MJPGOption()

```
MJPGOption ( ) [inline]
```

15.156.3 Member Data Documentation

15.156.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

15.156.3.2 height

```
unsigned int height
```

Height of source image.

15.156.3.3 quality

```
unsigned int quality
```

Image quality (1-100)

15.156.3.4 reserved

```
unsigned int reserved[192]
```

15.156.3.5 width

unsigned int width

Width of source image.

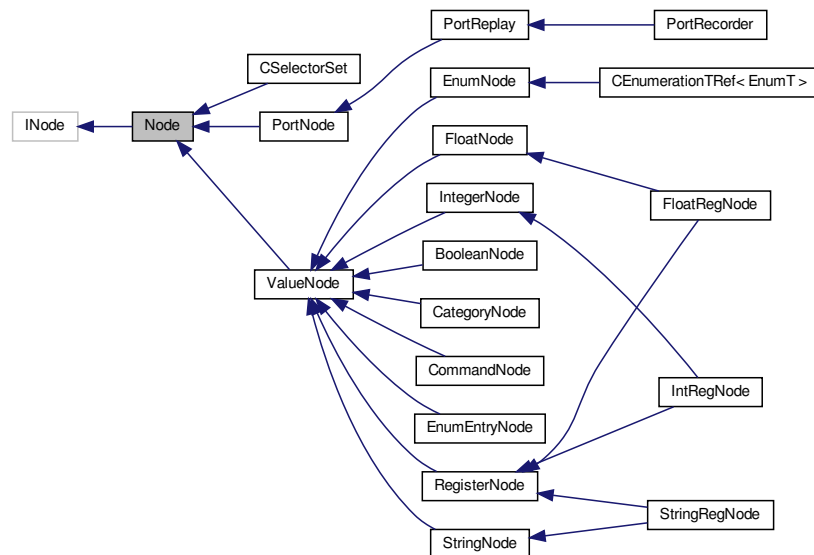
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

15.157 Node Class Reference

class common to all nodes

Inheritance diagram for Node:



Collaboration diagram for Node:



Public Member Functions

- [Node](#) ()
Constructor.
- [Node](#) (std::shared_ptr< Node::NodeImpl > pNodeHandle)
Constructor.
- [~Node](#) ()
Destructor.
- virtual [GenICam::gcstring GetName](#) (bool FullQualified=false) const
Get node name.
- virtual [GenApi::ENamespace GetNameSpace](#) () const
Get name space.
- virtual [EVisibility GetVisibility](#) () const
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const
Is the node value cacheable.
- virtual [EYesNo IsAccessModeCacheable](#) () const
True if the AccessMode can be cached.
- virtual [ECachingMode GetCachingMode](#) () const
Get Caching Mode.
- virtual [int64_t GetPollingTime](#) () const
recommended polling time (for not cacheable nodes)
- virtual [GenICam::gcstring GetToolTip](#) () const
Get a short description of the node.
- virtual [GenICam::gcstring GetDescription](#) () const
Get a long description of the node.
- virtual [GenICam::gcstring GetDisplayName](#) () const
Get a name string for display.
- virtual [GenICam::gcstring GetDeviceName](#) () const
Get a name of the device.
- virtual void [GetChildren](#) ([GenApi::NodeList_t](#) &Children, [ELinkType LinkType=ctReadingChildren](#)) const
Get all nodes this node directly depends on.
- virtual void [GetParents](#) ([GenApi::NodeList_t](#) &Parents) const
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType RegisterCallback](#) ([CNodeCallback](#) *pCallback)
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)
De register change callback Destroys CNodeCallback object.
- virtual [INodeMap * GetNodeMap](#) () const
Retrieves the central node map.
- virtual [GenICam::gcstring GetEventID](#) () const
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const
True if the node is streamable.
- virtual void [GetPropertyNames](#) ([GenICam::gcstring_vector](#) &PropertyNames) const
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)

Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.

- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)
Imposes a visibility to the natural visibility of the node.
- virtual [INode](#) * [GetAlias](#) () const
Retrieves the a node which describes the same feature in a different way.
- virtual [INode](#) * [GetCastAlias](#) () const
Retrieves the a node which describes the same feature so that it can be casted.
- virtual [GenlCam::gcstring](#) [GetDocuURL](#) () const
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const
True if the node should not be used any more.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const
True if the node can be reached via category nodes from a category node named "Root".
- void [SetNodeHandle](#) (std::shared_ptr< Node::NodeImpl > pNodeHandle)
Set [Node](#) handle.
- std::shared_ptr< Node::NodeImpl > [GetNodeHandle](#) () const
Get [Node](#) handle.
- virtual EAccessMode [GetAccessMode](#) () const
Base interface overrides.
- virtual bool [IsSelector](#) () const
Selector interface overrides.
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const
retrieve the group of features selecting this node
- virtual void [SetReference](#) ([INode](#) *pBase)
Reference interface overrides `ingroup Spinnaker_GenApi_PublicImpl`.
- virtual void [SetReference](#) ([ISelector](#) *pBase)
- void [SetNodeMap](#) ([INodeMap](#) *pNodeMap)
- virtual bool [operator==](#) (int nullPtr) const
- virtual bool [operator!=](#) (int nullPtr) const

Protected Attributes

- std::shared_ptr< Node::NodeImpl > [m_pNodeData](#)
- std::list< CallbackHandleType_t * > [m_Callbacks](#)
List of callbacks.
- [INodeMap](#) * [m_pNodeMap](#)

15.157.1 Detailed Description

class common to all nodes

15.157.2 Constructor & Destructor Documentation

15.157.2.1 Node() [1/2]

`Node ( )`

Constructor.

15.157.2.2 Node() [2/2]

`Node (`
`std::shared_ptr< Node::NodeImpl > pNodeHandle )`

Constructor.

15.157.2.3 ~Node()

`~Node ( )`

Destructor.

15.157.3 Member Function Documentation

15.157.3.1 DeregisterCallback()

```
virtual bool DeregisterCallback (
    CallbackHandleType hCallback ) [virtual]
```

De register change callback Destroys [CNodeCallback](#) object.

Returns

true if the callback handle was valid

15.157.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.157.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature in a different way.

15.157.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const [virtual]
```

Get Caching Mode.

15.157.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const [virtual]
```

Retrieves the a node which describes the same feature so that it can be casted.

15.157.3.6 GetChildren()

```
virtual void GetChildren (
    GenApi::NodeList_t & Children,
    ELinkType LinkType = ctReadingChildren ) const [virtual]
```

Get all nodes this node directly depends on.

Parameters

out	<i>Children</i>	List of children nodes
	<i>LinkType</i>	The link type

15.157.3.7 GetDescription()

```
virtual GenICam::gcstring GetDescription ( ) const [virtual]
```

Get a long description of the node.

15.157.3.8 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) const [virtual]
```

Get a name of the device.

15.157.3.9 GetDisplayName()

```
virtual GenICam::gcstring GetDisplayName ( ) const [virtual]
```

Get a name string for display.

15.157.3.10 GetDocuURL()

```
virtual GenICam::gcstring GetDocuURL ( ) const [virtual]
```

Gets a URL pointing to the documentation of that feature.

15.157.3.11 GetEventID()

```
virtual GenICam::gcstring GetEventID ( ) const [virtual]
```

Get the EventId of the node.

15.157.3.12 GetName()

```
virtual GenICam::gcstring GetName (
    bool FullQualified = false ) const [virtual]
```

Get node name.

15.157.3.13 GetNameSpace()

```
virtual GenApi::ENamespace GetNameSpace ( ) const [virtual]
```

Get name space.

15.157.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const
```

Get [Node](#) handle.

15.157.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap ( ) const [virtual]
```

Retrieves the central node map.

15.157.3.16 GetParents()

```
virtual void GetParents (
    GenApi::NodeList_t & Parents ) const [virtual]
```

Gets all nodes this node is directly depending on.

Parameters

out	<i>Parents</i>	List of parent nodes

15.157.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const [virtual]
```

recommended polling time (for not cacheable nodes)

15.157.3.18 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

15.157.3.19 GetProperty()

```
virtual bool GetProperty (
    const GenICam::gcstring & PropertyName,
    GenICam::gcstring & ValueStr,
    GenICam::gcstring & AttributeStr ) [virtual]
```

Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.

15.157.3.20 GetPropertyNames()

```
virtual void GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [virtual]
```

Returns a list of the names all properties set during initialization.

15.157.3.21 GetSelectedFeatures()

```
virtual void GetSelectedFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of selected features

15.157.3.22 GetSelectingFeatures()

```
virtual void GetSelectingFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of features selecting this node

15.157.3.23 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) const [virtual]
```

Get a short description of the node.

15.157.3.24 GetVisibility()

```
virtual EVisibility GetVisibility ( ) const [virtual]
```

Get the recommended visibility of the node.

15.157.3.25 ImposeAccessMode()

```
virtual void ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [virtual]
```

Imposes an access mode to the natural access mode of the node.

15.157.3.26 ImposeVisibility()

```
virtual void ImposeVisibility (
    EVisibility ImposedVisibility ) [virtual]
```

Imposes a visibility to the natural visibility of the node.

15.157.3.27 InvalidateNode()

```
virtual void InvalidateNode ( ) [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

15.157.3.28 IsAccessModeCacheable()

```
virtual EYesNo IsAccessModeCacheable ( ) const [virtual]
```

True if the AccessMode can be cached.

15.157.3.29 IsCachable()

```
virtual bool IsCachable ( ) const [virtual]
```

Is the node value cacheable.

15.157.3.30 IsDeprecated()

```
virtual bool IsDeprecated ( ) const [virtual]
```

True if the node should not be used any more.

15.157.3.31 IsFeature()

```
virtual bool IsFeature ( ) const [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

15.157.3.32 IsSelector()

```
virtual bool IsSelector ( ) const [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

15.157.3.33 IsStreamable()

```
virtual bool IsStreamable ( ) const [virtual]
```

True if the node is streamable.

15.157.3.34 operator"!="()

```
virtual bool operator!= (
    int nullptr ) const [virtual]
```

15.157.3.35 operator==()

```
virtual bool operator==(
    int nullPtr ) const [virtual]
```

15.157.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
    CNodeCallback * pCallback ) [virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

15.157.3.37 SetNodeHandle()

```
void SetNodeHandle (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set [Node](#) handle.

15.157.3.38 SetNodeMap()

```
void SetNodeMap (
    INodeMap * pNodeMap )
```

15.157.3.39 SetReference() [1/2]

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.

Reimplemented in [FloatNode](#), [PortNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [ValueNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

15.157.3.40 SetReference() [2/2]

```
virtual void SetReference (
    ISelector * pBase ) [virtual]
```


15.157.4 Member Data Documentation

15.157.4.1 m_Callbacks

```
std::list<CallbackHandleType_t*> m_Callbacks [protected]
```

List of callbacks.

15.157.4.2 m_pNodeData

```
std::shared_ptr<Node::NodeImpl> m_pNodeData [protected]
```

15.157.4.3 m_pNodeMap

```
INodeMap* m_pNodeMap [protected]
```

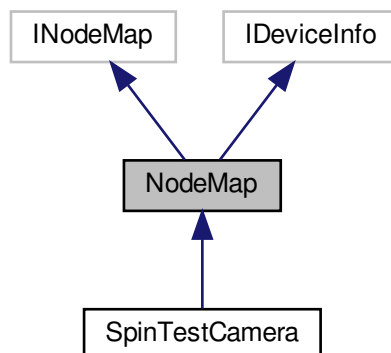
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Node.h](#)

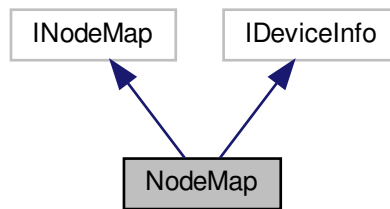
15.158 NodeMap Class Reference

Smart pointer template for NodeMaps with create function.

Inheritance diagram for NodeMap:



Collaboration diagram for NodeMap:



Public Member Functions

- **NodeMap** (**GenICam::gcstring** DeviceName="Device")
Constructor.
- virtual **~NodeMap** ()
Destructor.
- void **Destroy** ()
Destroys the node map.
- void **LoadXMLFromFile** (**GenICam::gcstring** FileName)
Creates the object from a XML file with given file name.
- void **LoadXMLFromZIPFile** (**GenICam::gcstring** ZipFileName)
Creates the object from a ZIP'd XML file with given file name.
- void **LoadXMLFromZIPData** (const void *zipData, size_t zipSize)
Creates the object from a ZIP'd XML file given in a string.
- void **LoadXMLFromFileInject** (**GenICam::gcstring** TargetFileName, **GenICam::gcstring** InjectFileName)
Creates the object from a XML target and an inject file with given file name.
- void **LoadXMLFromString** (const **GenICam::gcstring** &XMLData)
Creates the object from XML data given in a string.
- void **LoadXMLFromStringInject** (const **GenICam::gcstring** &TargetXMLDataconst, const **GenICam::gcstring** &InjectXMLData)
Creates the object from XML data given in a string with injection.
- virtual void **GetSupportedSchemaVersions** (**GenICam::gcstring_vector** &SchemaVersions)
Gets a list of supported schema versions.
- virtual **GenICam::gcstring** **GetDeviceName** ()
Get device name.
- virtual void **Poll** (int64_t ElapsedTime)
Fires nodes which have a polling time.
- virtual void **GetNodes** (**NodeList_t** &Nodes) const
Retrieves all nodes in the node map.
- virtual **INode** * **GetNode** (const **GenICam::gcstring** &key) const
Retrieves the node from the central map by name.
- virtual void **InvalidateNodes** () const
Invalidates all nodes.
- virtual bool **Connect** (**IPort** *pPort, const **GenICam::gcstring** &PortName) const
Connects a port to a port node with given name.

- virtual bool [Connect](#) ([IPort](#) *pPort) const
Connects a port to the standard port "Device".
- virtual [CLock](#) & [GetLock](#) () const
Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const
Get the number of nodes in the map.
- void * [GetNodeMapHandle](#) () const
- virtual [GenICam::gcstring](#) [GetModelName](#) ()
Get the model name.
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()
Get the vendor name.
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()
Get tool tip.
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()
Get the standard name space.
- virtual void [GetGenApiVersion](#) ([GenICam::Version_t](#) &Version, uint16_t &Build)
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) ([GenICam::Version_t](#) &Version)
Get the schema version number.
- virtual void [GetDeviceVersion](#) ([GenICam::Version_t](#) &Version)
Get the version of the device description file.
- virtual [GenICam::gcstring](#) [GetProductGuid](#) ()
Get the GUID describing the product.
- virtual [GenICam::gcstring](#) [GetVersionGuid](#) ()
Get the GUID describing the product version.

Static Public Member Functions

- static bool [ClearXMLCache](#) ()
Clears the cache of the camera description files.

Public Attributes

- [INodeMap](#) * [_Ptr](#)
Pointer to the [NodeMap](#).

15.158.1 Detailed Description

Smart pointer template for NodeMaps with create function.

Parameters

<i>TCameraParams</i>	The camera specific parameter class (auto generated from camera xml file)

15.158.2 Constructor & Destructor Documentation

15.158.2.1 NodeMap()

```
NodeMap (
    GenICam::gcstring DeviceName = "Device" )
```

Constructor.

15.158.2.2 ~NodeMap()

```
virtual ~NodeMap ( ) [virtual]
```

Destructor.

15.158.3 Member Function Documentation

15.158.3.1 ClearXMLCache()

```
static bool ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

15.158.3.2 Connect() [1/2]

```
virtual bool Connect (
    IPort * pPort ) const [virtual]
```

Connects a port to the standard port "Device".

15.158.3.3 Connect() [2/2]

```
virtual bool Connect (
    IPort * pPort,
    const GenICam::gcstring & PortName ) const [virtual]
```

Connects a port to a port node with given name.

15.158.3.4 Destroy()

```
void Destroy ( )
```

Destroys the node map.

15.158.3.5 GetDeviceName()

```
virtual GenICam::gcstring GetDeviceName ( ) [virtual]
```

Get device name.

15.158.3.6 GetDeviceVersion()

```
virtual void GetDeviceVersion (
    GenICam::Version_t & Version ) [virtual]
```

Get the version of the device description file.

15.158.3.7 GetGenApiVersion()

```
virtual void GetGenApiVersion (
    GenICam::Version_t & Version,
    uint16_t & Build ) [virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

15.158.3.8 GetLock()

```
virtual CLock& GetLock ( ) const [virtual]
```

Returns the lock which guards the node map.

15.158.3.9 GetModelName()

```
virtual GenICam::gcstring GetModelName ( ) [virtual]
```

Get the model name.

15.158.3.10 GetNode()

```
virtual INode* GetNode (
    const GenICam::gcstring & key ) const [virtual]
```

Retrieves the node from the central map by name.

15.158.3.11 GetNodeMapHandle()

```
void* GetNodeMapHandle ( ) const
```

15.158.3.12 GetNodes()

```
virtual void GetNodes (
    NodeList_t & Nodes ) const [virtual]
```

Retrieves all nodes in the node map.

15.158.3.13 GetNumNodes()

```
virtual uint64_t GetNumNodes ( ) const [virtual]
```

Get the number of nodes in the map.

15.158.3.14 GetProductGuid()

```
virtual GenICam::gcstring GetProductGuid ( ) [virtual]
```

Get the GUID describing the product.

15.158.3.15 GetSchemaVersion()

```
virtual void GetSchemaVersion (
    GenICam::Version_t & Version ) [virtual]
```

Get the schema version number.

15.158.3.16 GetStandardNameSpace()

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

15.158.3.17 GetSupportedSchemaVersions()

```
virtual void GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

! Loads an XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromFile(const [GenICam::gcstring](#)& XMLFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Loads a Zipped XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromZIPFile(const [GenICam::gcstring](#)& ZIPFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Injects an XML file into a target file virtual void MergeXMLFiles(const [GenICam::gcstring](#)& TargetFileName, *< Name of the target XML file to process const [GenICam::gcstring](#)& InjectedFileName, *< Name of the Injected XML file to process const [GenICam::gcstring](#)& OutputFileName *< Name of the output file);

! Extract independent subtree virtual void ExtractIndependentSubtree(const [GenICam::gcstring](#)& XMLData, *< The XML data the subtree is extracted from. const [GenICam::gcstring](#)& InjectXMLData, *< Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed. const [GenICam::gcstring](#)& SubTreeRootNodeName,*< The name of the node that represents the root of the subtree that shall be extracted. [GenICam::gcstring](#)& ExtractedSubtree *< The returned extracted subtree as string.);

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

15.158.3.18 GetToolTip()

```
virtual GenICam::gcstring GetToolTip ( ) [virtual]
```

Get tool tip.

15.158.3.19 GetVendorName()

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

15.158.3.20 GetVersionGuid()

```
virtual GenICam::gcstring GetVersionGuid ( ) [virtual]
```

Get the GUID describing the product version.

15.158.3.21 InvalidateNodes()

```
virtual void InvalidateNodes ( ) const [virtual]
```

Invalidates all nodes.

15.158.3.22 LoadXMLFromFile()

```
void LoadXMLFromFile (
    GenICam::gcstring FileName )
```

Creates the object from a XML file with given file name.

! Creates the object from the default DLL ! note Can only be used if the class TCameraParams was auto generated from a specific camera xml file void LoadDLL(void);

! Creates the object from a DLL whose name is deduced from vendor and model name void LoadDLL(GenICam::gcstring VendorName, GenICam::gcstring ModelName);

! Creates the object from a DLL with given file name void LoadDLL(GenICam::gcstring FileName);

15.158.3.23 LoadXMLFromFileInject()

```
void LoadXMLFromFileInject (
    GenICam::gcstring TargetFileName,
    GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

15.158.3.24 LoadXMLFromString()

```
void LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

15.158.3.25 LoadXMLFromStringInject()

```
void LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

15.158.3.26 LoadXMLFromZIPData()

```
void LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

15.158.3.27 LoadXMLFromZIPFile()

```
void LoadXMLFromZIPFile (
    GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

15.158.3.28 Poll()

```
virtual void Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

15.158.4 Member Data Documentation

15.158.4.1 _Ptr

`INodeMap* _Ptr`

Pointer to the [NodeMap](#).

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

- `include/SpinGenApi/NodeMap.h`

15.159 CNodeMapFactory::NodeStatistics_t Struct Reference

Public Attributes

- uint32_t [NumNodes](#)
- uint32_t [NumProperties](#)
- uint32_t [NumLinks](#)
- uint32_t [NumStrings](#)

15.159.1 Member Data Documentation

15.159.1.1 NumLinks

uint32_t NumLinks

15.159.1.2 NumNodes

uint32_t NumNodes

15.159.1.3 NumProperties

uint32_t NumProperties

15.159.1.4 NumStrings

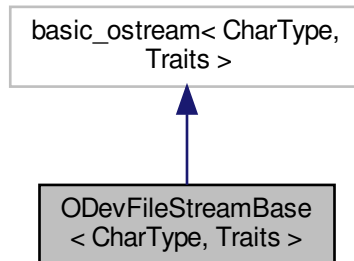
uint32_t NumStrings

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

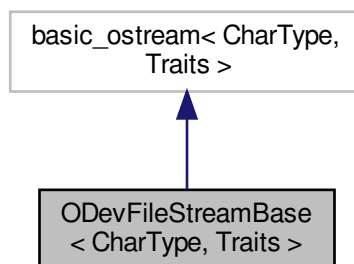
- include/SpinGenApi/[NodeMapFactory.h](#)

15.160 ODevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBase< CharType, Traits >:



Collaboration diagram for ODevFileStreamBase< CharType, Traits >:



Public Types

- typedef `ODEvFileStreamBuf< CharType, Traits >` `filebuf_type`
- typedef `std::basic_ios< CharType, Traits >` `ios_type`
- typedef `std::basic_ostream< CharType, Traits >` `ostream_type`

Public Member Functions

- `filebuf_type * rdbuf () const`
- `bool is_open () const`
- `void open (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out|std::ios_base::trunc)`
Open file on device in write mode.
- `void close ()`
Close the file on device.

15.160.1 Member Typedef Documentation

15.160.1.1 filebuf_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

15.160.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

15.160.1.3 ostream_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

15.160.2 Member Function Documentation

15.160.2.1 close()

```
void close ( ) [inline]
```

Close the file on device.

15.160.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.160.2.3 open()

```
void open (
    INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::out | std::ios_base::trunc ) [inline]
```

Open file on device in write mode.

Parameters

<i>pInterface</i>	NodeMap of the device to which the FileProtocolAdapter is attached
<i>pFileName</i>	Name of the file to open
<i>mode</i>	open mode

15.160.2.4 rdbuf()

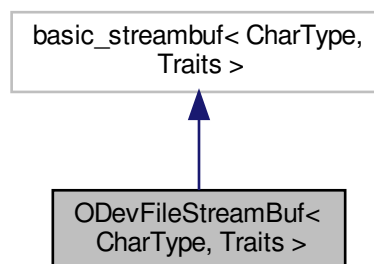
```
filebuf_type* rdbuf ( ) const [inline]
```

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

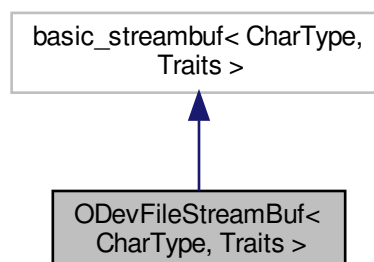
- include/SpinGenApi/[Filestream.h](#)

15.161 ODevFileStreamBuf< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBuf< CharType, Traits >:



Collaboration diagram for ODevFileStreamBuf< CharType, Traits >:



Public Member Functions

- [ODevFileStreamBuf](#) ()
- [~ODevFileStreamBuf](#) ()
- [filebuf_type](#) * [open](#) ([Spinnaker::GenApi::INodeMap](#) *pInterface, const char *pFileName, std::ios_base::openmode mode)
- bool [is_open](#) () const
- [filebuf_type](#) * [close](#) ()

Protected Member Functions

- std::streamsize [xsputn](#) (const char_type *s, std::streamsize n)
- int_type [overflow](#) (int_type c=traits_type::eof())
- int [sync](#) ()

15.161.1 Constructor & Destructor Documentation

15.161.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

15.161.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

15.161.2 Member Function Documentation

15.161.2.1 close()

```
filebuf_type* close ( ) [inline]
```

15.161.2.2 is_open()

```
bool is_open ( ) const [inline]
```

15.161.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode ) [inline]
```

15.161.2.4 overflow()

```
int_type overflow (
    int_type c = traits_type::eof() ) [inline], [protected]
```

15.161.2.5 sync()

```
int sync ( ) [inline], [protected]
```

15.161.2.6 xsputn()

```
std::streamsize xsputn (
    const char_type * s,
    std::streamsize n ) [inline], [protected]
```

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

- include/SpinGenApi/[Filestream.h](#)

15.162 PGMOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PGMOption](#) ()

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

15.162.1 Detailed Description

Options for saving PGM images.

15.162.2 Constructor & Destructor Documentation

15.162.2.1 PGMOption()

```
PGMOption ( ) [inline]
```

15.162.3 Member Data Documentation

15.162.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

15.162.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

15.163 PNGOption Struct Reference

[Spinnaker](#) struct definitions.

Public Member Functions

- [PNGOption](#) ()

Public Attributes

- bool [interlaced](#)
Whether to save the PNG as interlaced.
- unsigned int [compressionLevel](#)
Compression level (0-9).
- unsigned int [reserved](#) [16]
Reserved for future use.

15.163.1 Detailed Description

[Spinnaker](#) struct definitions.

Options for saving PNG images.

15.163.2 Constructor & Destructor Documentation

15.163.2.1 PNGOption()

```
PNGOption ( ) [inline]
```

15.163.3 Member Data Documentation

15.163.3.1 compressionLevel

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

15.163.3.2 interlaced

```
bool interlaced
```

Whether to save the PNG as interlaced.

15.163.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

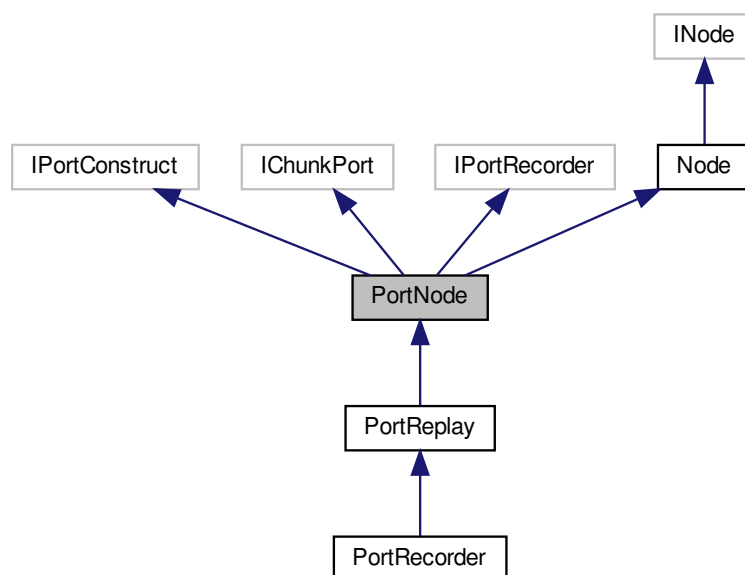
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

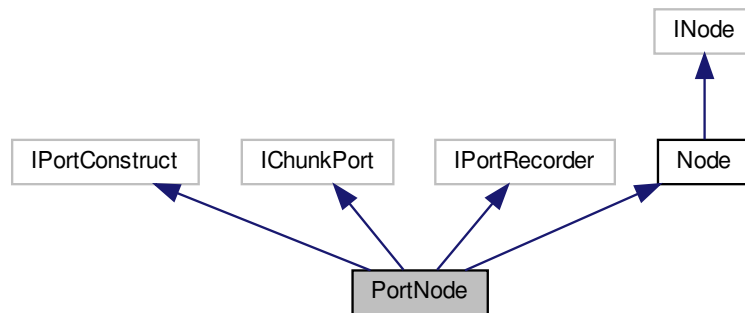
15.164 PortNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for PortNode:



Collaboration diagram for PortNode:



Public Member Functions

- [PortNode](#) ()
Constructor.
- [PortNode](#) (std::shared_ptr< Node::NodeImpl > pValue)
constructor with [GenICam](#) IValue
- [~PortNode](#) ()
Destructor.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual EYesNo [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual [Spinnaker::GenICam::gcstring](#) [GetChunkID](#) () const
Get the Id of the chunk the port should be attached to.
- virtual EYesNo [CacheChunkData](#) () const
Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual void [StartRecording](#) (IPortWriteList *pPortRecorder)
Starts logging all WriteRegister commands to a list.
- virtual void [StopRecording](#) ()
Stops recording.
- virtual void [Replay](#) (IPortWriteList *pPortRecorder, bool [Invalidate](#)=true)
Sends the commands to the camera.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IPort *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IChunkPort *pBase)
overload SetReference for Value
- std::shared_ptr< Node::NodeImpl > [GetPortHandle](#) ()
- virtual void [SetReference](#) (INode *pBase)
Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.
- virtual void [SetReference](#) (ISelector *pBase)

Additional Inherited Members

15.164.1 Detailed Description

[Interface](#) for value properties.

15.164.2 Constructor & Destructor Documentation

15.164.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

15.164.2.2 PortNode() [2/2]

```
PortNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenlCam](#) IValue

15.164.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

15.164.3 Member Function Documentation

15.164.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const [virtual]
```

Indicates if the chunk a adapter must hold a cached version of the chunk data.

15.164.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const [virtual]
```

Get the Id of the chunk the port should be attached to.

15.164.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle ( ) [inline]
```

15.164.3.4 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [virtual]
```

Determines if the port adapter must perform an endianness swap.

15.164.3.5 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.164.3.6 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [virtual]
```

Sends the commands to the camera.

The default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.164.3.7 SetPortImpl()

```
void SetPortImpl (
    IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

15.164.3.8 SetReference() [1/5]

```
virtual void SetReference (
    IChunkPort * pBase ) [virtual]
```

overload SetReference for Value

15.164.3.9 SetReference() [2/5]

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

15.164.3.10 SetReference() [3/5]

```
virtual void SetReference
```

Reference interface overrides \ingroup Spinnaker_GenApi_PublicImpl.

15.164.3.11 SetReference() [4/5]

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

15.164.3.12 SetReference() [5/5]

```
virtual void SetReference
```

15.164.3.13 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

15.164.3.14 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in [PortRecorder](#).

15.164.3.15 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented in [PortRecorder](#), and [PortReplay](#).

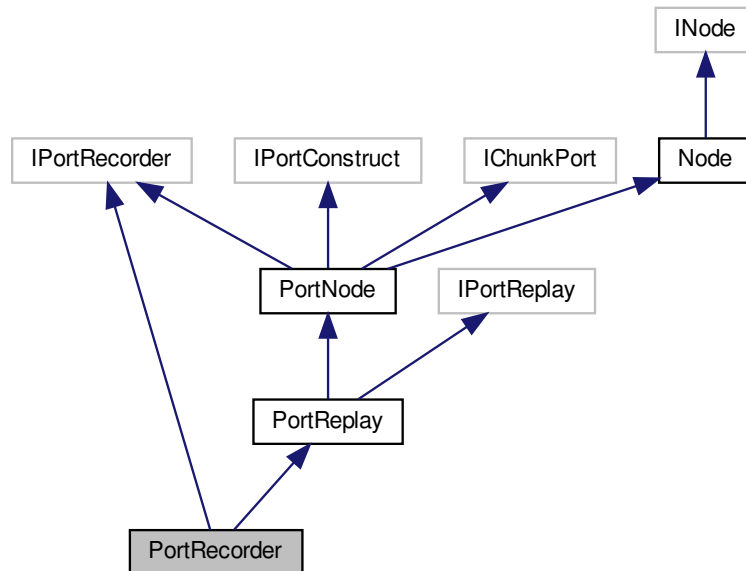
The documentation for this class was generated from the following file:

- [include/SpinGenApi/PortNode.h](#)

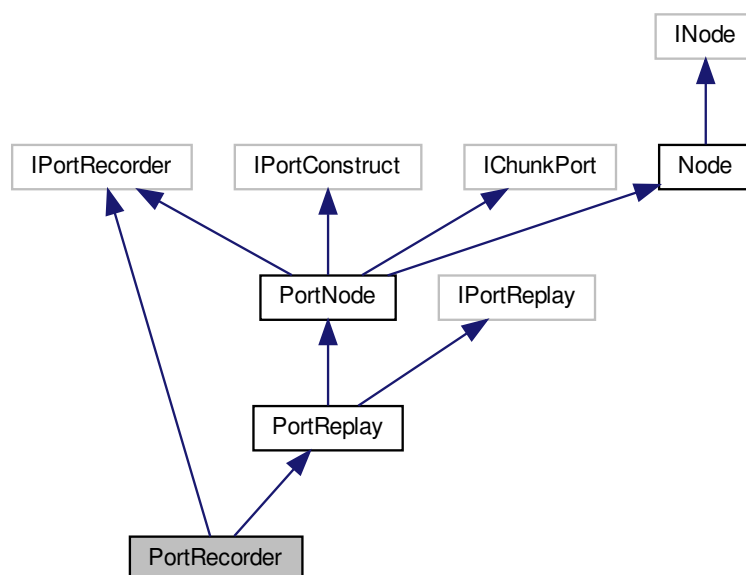
15.165 PortRecorder Class Reference

[Interface](#) for recording write commands on a port.

Inheritance diagram for PortRecorder:



Collaboration diagram for PortRecorder:



Public Member Functions

- [PortRecorder](#) ()
- virtual [~PortRecorder](#) ()
- virtual void [StartRecording](#) ([IPortWriteList](#) *pPortRecorder)
starts logging all WriteRegister commands to a list
- virtual void [StopRecording](#) ()
stops recording
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual void [SetReference](#) ([IPort](#) *pBase)
overload SetReference for Value
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.

Additional Inherited Members

15.165.1 Detailed Description

[Interface](#) for recording write commands on a port.

15.165.2 Constructor & Destructor Documentation

15.165.2.1 PortRecorder()

[PortRecorder](#) ()

15.165.2.2 ~PortRecorder()

virtual [~PortRecorder](#) () [virtual]

15.165.3 Member Function Documentation

15.165.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

Reimplemented from [PortReplay](#).

15.165.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from [PortReplay](#).

15.165.3.3 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [inline], [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from [PortReplay](#).

15.165.3.4 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortReplay](#).

15.165.3.5 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

15.165.3.6 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

stops recording

Reimplemented from [PortNode](#).

15.165.3.7 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from [PortReplay](#).

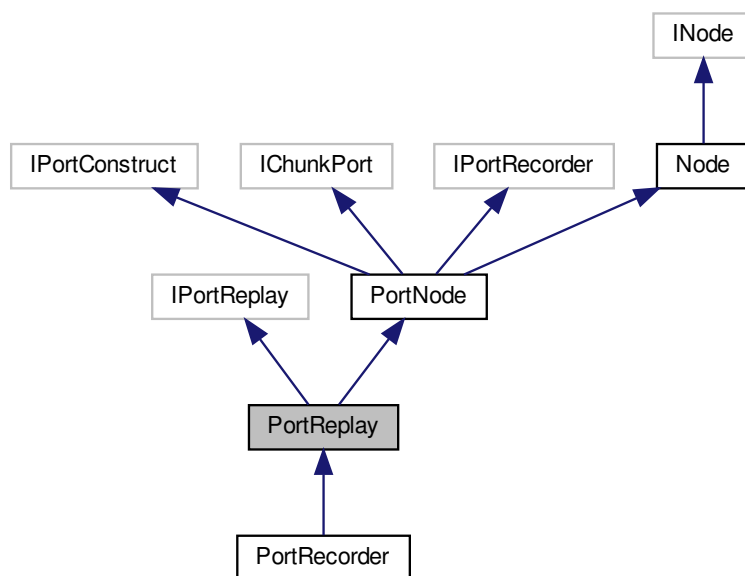
The documentation for this class was generated from the following file:

- [include/SpinGenApi/PortRecorder.h](#)

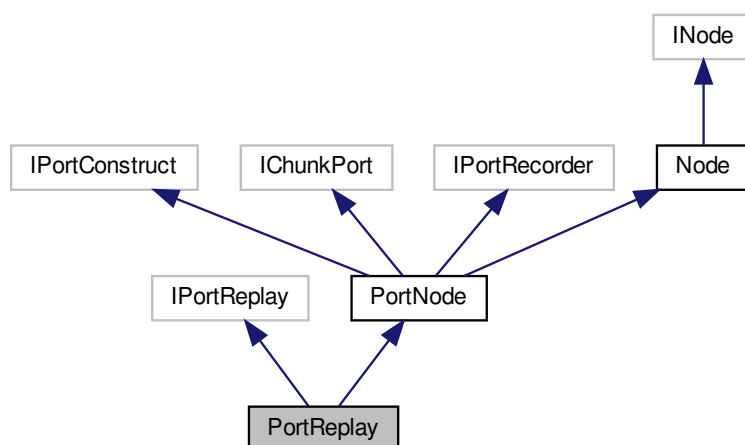
15.166 PortReplay Class Reference

[Interface](#) for replaying write commands on a port.

Inheritance diagram for PortReplay:



Collaboration diagram for PortReplay:



Public Member Functions

- [PortReplay](#) ()
- virtual [~PortReplay](#) ()
- virtual void [Replay](#) ([IPortWriteList](#) *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- virtual void [SetReference](#) ([IPort](#) *pBase)
overload SetReference for Value
- void * [GetPortReplayHandle](#) ()
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Base interface overrides.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.

Additional Inherited Members

15.166.1 Detailed Description

[Interface](#) for replaying write commands on a port.

15.166.2 Constructor & Destructor Documentation

15.166.2.1 PortReplay()

[PortReplay](#) ()

15.166.2.2 ~PortReplay()

virtual [~PortReplay](#) () [virtual]

15.166.3 Member Function Documentation

15.166.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented from [Node](#).

Reimplemented in [PortRecorder](#).

15.166.3.2 GetPortReplayHandle()

```
void* GetPortReplayHandle ( )
```

15.166.3.3 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.166.3.4 Replay()

```
virtual void Replay (
    IPortWriteList * pPortRecorder,
    bool Invalidate = true ) [virtual]
```

sends the commands to the camera.

the default implementation just walks the list and issues each command using the WriteRegister method. Depending on the capabilities of the transport layer the implementation can however use a special command which sends all register write commands as one package.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.166.3.5 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

15.166.3.6 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Writes a chunk of bytes to the port.

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

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

- [include/SpinGenApi/PortReplay.h](#)

15.167 PPMOption Struct Reference

Options for saving PPM images.

Public Member Functions

- [PPMOption](#) ()

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

15.167.1 Detailed Description

Options for saving PPM images.

15.167.2 Constructor & Destructor Documentation

15.167.2.1 PPMOption()

```
PPMOption ( ) [inline]
```

15.167.3 Member Data Documentation

15.167.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

15.167.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

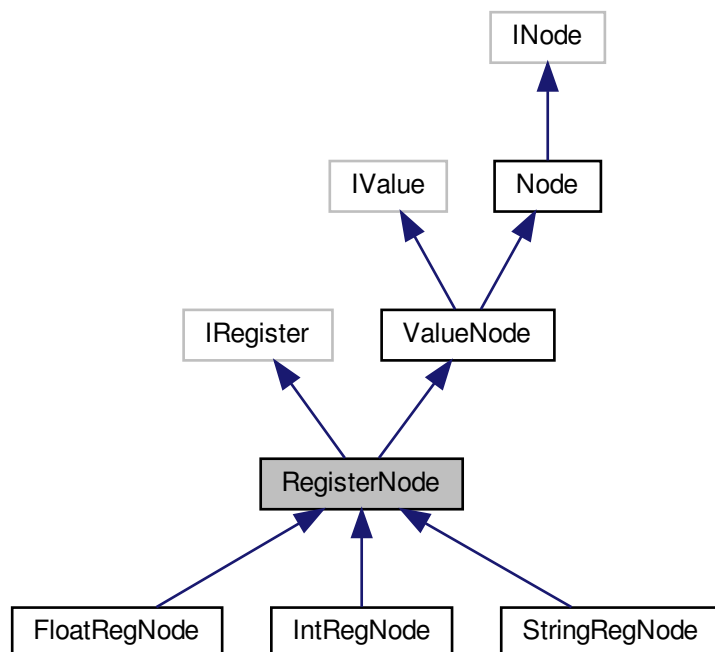
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

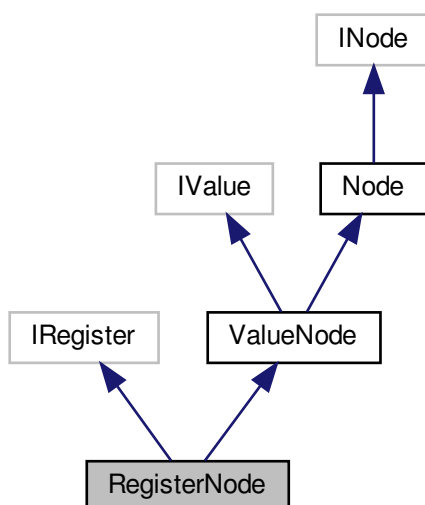
15.168 RegisterNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for RegisterNode:



Collaboration diagram for RegisterNode:



Public Member Functions

- [RegisterNode](#) ()
- [RegisterNode](#) (std::shared_ptr< Node::NodeImpl > pRegister)
- virtual [~RegisterNode](#) ()
- virtual void [Set](#) (const uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=true)
Set the register's contents.
- virtual void [Get](#) (uint8_t *pBuffer, int64_t [Length](#), bool [Verify](#)=false, bool IgnoreCache=false)
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()
Retrieves the Address of the register.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Register

Additional Inherited Members

15.168.1 Detailed Description

[Interface](#) for string properties.

15.168.2 Constructor & Destructor Documentation

15.168.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

15.168.2.2 RegisterNode() [2/2]

```
RegisterNode (
    std::shared_ptr< Node::NodeImpl > pRegister )
```

15.168.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

15.168.3 Member Function Documentation

15.168.3.1 Get()

```
virtual void Get (
    uint8_t * pBuffer,
    int64_t Length,
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Fills a buffer with the register's contents.

Parameters

<i>pBuffer</i>	The buffer receiving the data to read
<i>Length</i>	The number of bytes to retrieve
<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.168.3.2 GetAddress()

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

15.168.3.3 GetLength()

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

15.168.3.4 Set()

```
virtual void Set (
    const uint8_t * pBuffer,
    int64_t Length,
    bool Verify = true ) [virtual]
```

Set the register's contents.

Parameters

<i>pBuffer</i>	The buffer containing the data to set
<i>Length</i>	The number of bytes in pBuffer
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.168.3.5 SetReference()

```
virtual void SetReference (  
    INode * pBase ) [virtual]
```

overload SetReference for Register

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

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

- include/SpinGenApi/[RegisterNode.h](#)

15.169 SingleChunkData_t Struct Reference

Public Attributes

- uint64_t [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

15.169.1 Member Data Documentation

15.169.1.1 ChunkID

```
uint64_t ChunkID
```

15.169.1.2 ChunkLength

```
size_t ChunkLength
```

15.169.1.3 ChunkOffset

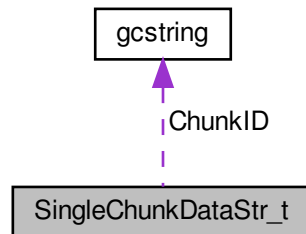
```
ptrdiff_t ChunkOffset
```

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

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

15.170 SingleChunkDataStr_t Struct Reference

Collaboration diagram for SingleChunkDataStr_t:



Public Attributes

- [GenICam::gcstring](#) [ChunkID](#)
- [ptrdiff_t](#) [ChunkOffset](#)
- [size_t](#) [ChunkLength](#)

15.170.1 Member Data Documentation

15.170.1.1 ChunkID

[GenICam::gcstring](#) [ChunkID](#)

15.170.1.2 ChunkLength

[size_t](#) [ChunkLength](#)

15.170.1.3 ChunkOffset

[ptrdiff_t](#) [ChunkOffset](#)

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

- [include/SpinGenApi/ChunkAdapterGeneric.h](#)

15.171 SIOption Struct Reference

Options for saving [Spinnaker](#) image.

Public Member Functions

- [SIOption](#) ()

Public Attributes

- unsigned int [reserved](#) [16]
Reserved for future use.

15.171.1 Detailed Description

Options for saving [Spinnaker](#) image.

15.171.2 Constructor & Destructor Documentation

15.171.2.1 SIOption()

```
SIOption ( ) [inline]
```

15.171.3 Member Data Documentation

15.171.3.1 reserved

```
unsigned int reserved[16]
```

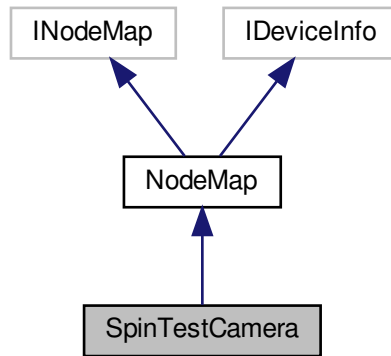
Reserved for future use.

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

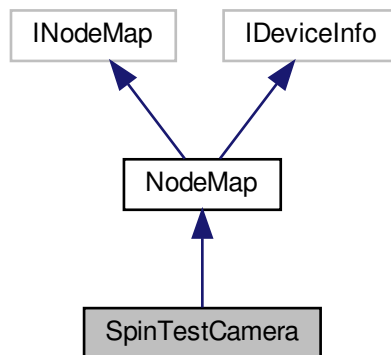
- include/[SpinnakerDefs.h](#)

15.172 SpinTestCamera Class Reference

Inheritance diagram for SpinTestCamera:



Collaboration diagram for SpinTestCamera:



Additional Inherited Members

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

- [include/SpinGenApi/SpinTestCamera.h](#)

15.173 SpinVideo Class Reference

Provides the functionality for the user to record images to an AVI/MP4 file.

Public Member Functions

- [SpinVideo](#) ()
Default constructor.
- virtual [~SpinVideo](#) ()
Default destructor.
- virtual void [Open](#) (const char *pFileName, [AVIOption](#) &pOption)
Open an video file in preparation for writing Images to disk.
- virtual void [Open](#) (const char *pFileName, [MJPGOption](#) &pOption)
Open an MJPEG video file in preparation for writing Images to disk.
- virtual void [Open](#) (const char *pFileName, [H264Option](#) &pOption)
Open an H264 MP4 video file in preparation for writing Images to disk.
- virtual void [Append](#) ([ImagePtr](#) pImage)
Append an image to the video file.
- virtual void [Close](#) ()
Close the video file.
- virtual void [SetMaximumFileSize](#) (unsigned int size)
Set the maximum file size (in megabytes) of a AVI/MP4 file.

15.173.1 Detailed Description

Provides the functionality for the user to record images to an AVI/MP4 file.

15.173.2 Constructor & Destructor Documentation

15.173.2.1 SpinVideo()

```
SpinVideo ( )
```

Default constructor.

15.173.2.2 ~SpinVideo()

```
virtual ~SpinVideo ( ) [virtual]
```

Default destructor.

15.173.3 Member Function Documentation

15.173.3.1 Append()

```
virtual void Append (  
    ImagePtr pImage ) [virtual]
```

Append an image to the video file.

When using the H264 encoder, several images are required to be appended before the encoder is able to output the first encoded frame.

Parameters

<i>pImage</i>	The image to append.

15.173.3.2 Close()

```
virtual void Close ( ) [virtual]
```

Close the video file.

This function will throw an exception when the H264 encoder was unable to output any encoded frames, in which case the output video should be considered invalid.

See also

[Open\(\)](#)[Append\(ImagePtr pImage\)](#)**15.173.3.3 Open() [1/3]**

```
virtual void Open (
    const char * pFileName,
    AVIOption & pOption ) [virtual]
```

Open an video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the video file.
<i>pOption</i>	Options to apply to the video file.

See also

[Close\(\)](#)**15.173.3.4 Open() [2/3]**

```
virtual void Open (
    const char * pFileName,
    H264Option & pOption ) [virtual]
```

Open an H264 MP4 video file in preparation for writing Images to disk.

The size of MP4 files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the MP4 video file.
<i>pOption</i>	H264 options to apply to the MP4 video file.

See also

[Close\(\)](#)[H264Option](#)**15.173.3.5 Open() [3/3]**

```
virtual void Open (  
    const char * pFileName,  
    MJPGOption & pOption ) [virtual]
```

Open an MJPEG video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

<i>pFileName</i>	The filename of the video file.
<i>pOption</i>	MJPEG options to apply to the video file.

See also

[Close\(\)](#)[MJPGOption](#)**15.173.3.6 SetMaximumFileSize()**

```
virtual void SetMaximumFileSize (  
    unsigned int size ) [virtual]
```

Set the maximum file size (in megabytes) of a AVI/MP4 file.

A new video file is created automatically when file size limit is reached. Setting a maximum size of 0 indicates no limit on file size.

Note that this size serves only as a hint and can still be slightly exceeded in some cases after the video trailer is written.

Parameters

<i>size</i>	The maximum video file size in MB.

See also

[Append\(ImagePtr pImage\)](#)

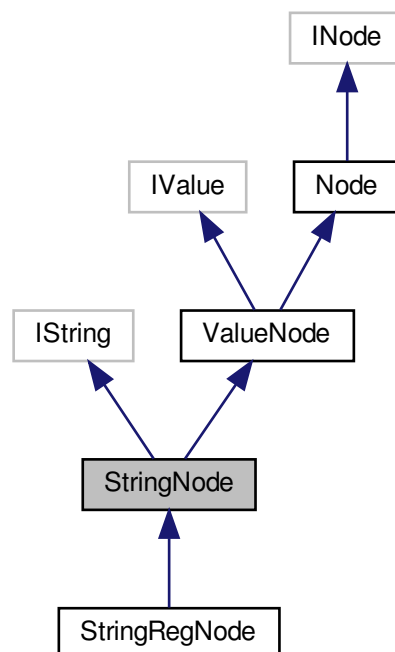
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

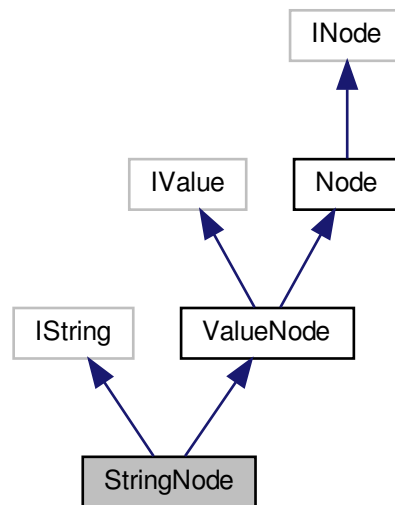
15.174 StringNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringNode:



Collaboration diagram for StringNode:



Public Member Functions

- [StringNode](#) ()
- [StringNode](#) (std::shared_ptr< Node::NodeImpl > pString)
- virtual [~StringNode](#) ()
- virtual void [SetValue](#) (const [GenICam::gcstring](#) &Value, bool [Verify](#)=true)
Set node value.
- virtual [IString](#) & [operator=](#) (const [GenICam::gcstring](#) &Value)
Set node value.
- virtual [GenICam::gcstring](#) [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get node value.
- virtual [GenICam::gcstring](#) [operator\(\)](#) ()
Get node value.
- virtual [GenICam::gcstring](#) [operator*](#) ()
Get node value.
- virtual int64_t [GetMaxLength](#) ()
Retrieves the maximum length of the string in bytes.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Value

Additional Inherited Members

15.174.1 Detailed Description

[Interface](#) for string properties.

15.174.2 Constructor & Destructor Documentation

15.174.2.1 StringNode() [1/2]

```
StringNode ( )
```

15.174.2.2 StringNode() [2/2]

```
StringNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

15.174.2.3 ~StringNode()

```
virtual ~StringNode ( ) [virtual]
```

15.174.3 Member Function Documentation

15.174.3.1 GetMaxLength()

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

15.174.3.2 GetValue()

```
virtual GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

15.174.3.3 operator>()()

```
virtual GenICam::gcstring operator() ( ) [virtual]
```

Get node value.

15.174.3.4 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get node value.

15.174.3.5 operator=()

```
virtual IString& operator= (
    const GenICam::gcstring & Value ) [virtual]
```

Set node value.

15.174.3.6 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

15.174.3.7 SetValue()

```
virtual void SetValue (
    const GenICam::gcstring & Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

<i>Value</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

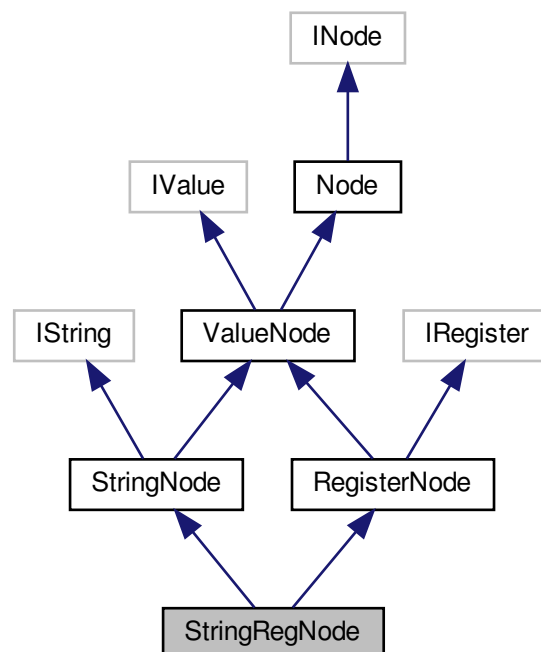
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringNode.h](#)

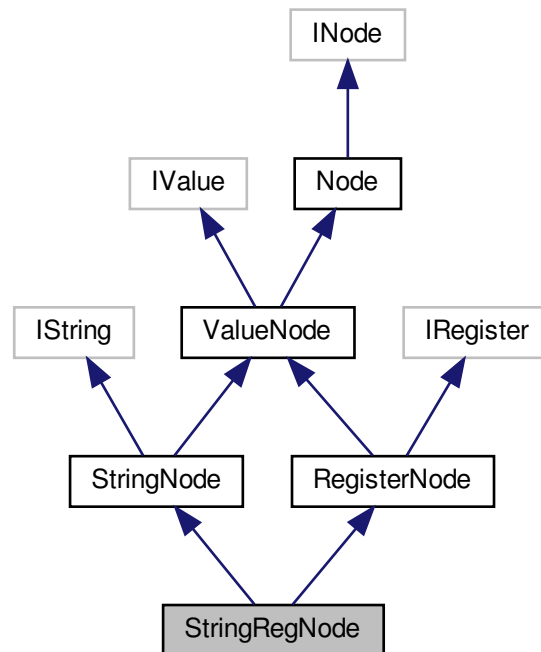
15.175 StringRegNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for StringRegNode:



Collaboration diagram for StringRegNode:



Public Member Functions

- [StringRegNode](#) ()
- [StringRegNode](#) (std::shared_ptr< Node::NodeImpl > pString)
- virtual [~StringRegNode](#) ()
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

15.175.1 Detailed Description

[Interface](#) for string properties.

15.175.2 Constructor & Destructor Documentation

15.175.2.1 StringRegNode() [1/2]

```
StringRegNode ( )
```

15.175.2.2 StringRegNode() [2/2]

```
StringRegNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

15.175.2.3 ~StringRegNode()

```
virtual ~StringRegNode ( ) [virtual]
```

15.175.3 Member Function Documentation

15.175.3.1 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [RegisterNode](#).

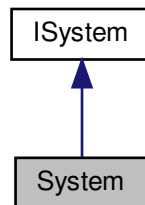
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringRegNode.h](#)

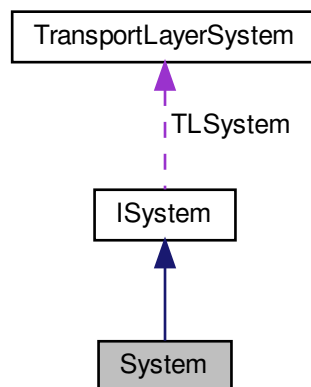
15.176 System Class Reference

The system object is used to retrieve the list of interfaces and cameras available.

Inheritance diagram for System:



Collaboration diagram for System:



Public Member Functions

- virtual [~System](#) ()
Default destructor.
- virtual void [ReleaseInstance](#) ()
This call releases the referenced instance of the [System](#) Singleton.
- virtual [InterfaceList](#) [GetInterfaces](#) (bool updateInterface=true)
Returns a list of interfaces available on the system.
- virtual void [UpdateInterfaceList](#) ()
Updates the list of interfaces on the system.

- virtual [CameraList GetCameras](#) (bool updateInterfaces=true, bool updateCameras=true)
Returns a list of cameras that are available on the system.
- virtual bool [UpdateCameras](#) (bool updateInterfaces=true)
Updates the list of cameras on the system.
- void [RegisterEventHandler](#) ([EventHandler](#) &evtHandlerToRegister, bool updateInterface=false)
Registers an event handler for the system.
- void [UnregisterEventHandler](#) ([EventHandler](#) &evtHandlerToUnregister)
Unregisters an event handler for the system.
- virtual void [RegisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)
Registers a logging event.
- virtual void [UnregisterAllLoggingEventHandlers](#) ()
Unregisters all previously registered logging events.
- virtual void [UnregisterLoggingEventHandler](#) ([LoggingEventHandler](#) &handler)
Unregisters a logging event.
- virtual void [SetLoggingEventPriorityLevel](#) ([SpinnakerLogLevel](#) level)
Sets a threshold priority level for logging event.
- virtual [SpinnakerLogLevel GetLoggingEventPriorityLevel](#) ()
Retrieves the current logging event priority level.
- virtual bool [IsInUse](#) ()
Checks if the system is in use by any interface or camera objects.
- virtual void [SendActionCommand](#) (unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, [ActionCommandResult](#) results[]=NULL)
Broadcast an Action Command to all devices on system.
- virtual const [LibraryVersion GetLibraryVersion](#) ()
Get current library version of [Spinnaker](#).
- virtual [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const
Gets a reference to the system node map.

Static Public Member Functions

- static [SystemPtr GetInstance](#) ()
Returns a pointer to a Singleton instance of a [System](#) object.

Protected Member Functions

- [System](#) ()
Default constructor.

Additional Inherited Members

15.176.1 Detailed Description

The system object is used to retrieve the list of interfaces and cameras available.

15.176.2 Constructor & Destructor Documentation

15.176.2.1 ~System()

```
virtual ~System ( ) [virtual]
```

Default destructor.

15.176.2.2 System()

```
System ( ) [protected]
```

Default constructor.

15.176.3 Member Function Documentation

15.176.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [virtual]
```

Returns a list of cameras that are available on the system.

This call returns both GigE Vision and Usb3 Vision cameras from all interfaces. The camera list object will reference count the cameras it returns. It is important that the camera list is destroyed or is cleared before calling system->[ReleaseInstance\(\)](#) or else the call to system->[ReleaseInstance\(\)](#) will result in an error message thrown that a reference to the camera is still held.

See also

[ReleaseInstance\(\)](#)

[CameraList::Clear\(\)](#)

Parameters

<i>updateInterfaces</i>	Determines whether or not updateInterfaceList() is called before getting cameras from available interfaces on the system
<i>updateCameras</i>	Determines whether or not UpdateCameras() is called before getting cameras from available interfaces on the system

Returns

An [CameraList](#) object that contains a list of all cameras.

Implements [ISystem](#).

15.176.3.2 GetInstance()

```
static SystemPtr GetInstance ( ) [static]
```

Returns a pointer to a Singleton instance of a [System](#) object.

The [System](#) object may be used to get cameras or interfaces. When an application is done using the cameras it is necessary to free the [System](#) by calling [ReleaseInstance\(\)](#).

See also

[ReleaseInstance\(\)](#)

Returns

A const ref to a system object.

15.176.3.3 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [virtual]
```

Returns a list of interfaces available on the system.

This call returns GigE and Usb2 and Usb3 interfaces. Note that on MacOS only active GigE interfaces will be stored in the returned [InterfaceList](#).

See also

[UpdateInterfaceList\(\)](#)

Parameters

<i>updateInterface</i>	Determines whether or not UpdateInterfaceList() is called before getting available interfaces

Returns

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

15.176.3.4 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( ) [virtual]
```

Get current library version of [Spinnaker](#).

Returns

A struct containing the current version of [Spinnaker](#) (major, minor, type, build).

Implements [ISystem](#).

15.176.3.5 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [virtual]
```

Retrieves the current logging event priority level.

[Spinnaker](#) uses multiple levels of logging:

- Fatal - failures that are non-recoverable without user intervention.
 - Error - failures that may or may not be recoverable without user intervention (use case dependent).
 - Warning - failures that are recoverable without user intervention.
 - Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
 - Info - information about recurring events that are generated regularly such as information on individual images.
 - Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

Returns

Level The threshold level

Implements [ISystem](#).

15.176.3.6 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [virtual]
```

Gets a reference to the system node map.

The system must be initialized by a call to [System::GetInstance\(\)](#) first before a node map reference can be successfully acquired.

Returns

A reference to the [System](#) [INodeMap](#).

Implements [ISystem](#).

15.176.3.7 IsInUse()

```
virtual bool IsInUse ( ) [virtual]
```

Checks if the system is in use by any interface or camera objects.

Returns

Returns true if the system is in use and false otherwise.

Implements [ISystem](#).

15.176.3.8 RegisterEventHandler()

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = false ) [virtual]
```

Registers an event handler for the system.

See also

[SystemEventHandler](#)

Parameters

<i>evtHandlerToRegister</i>	The event handler to register for the system
<i>updateInterface</i>	Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

Implements [ISystem](#).

15.176.3.9 RegisterLoggingEventHandler()

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Registers a logging event.

Parameters

<i>handler</i>	The logging event handler to register

Implements [ISystem](#).

15.176.3.10 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [virtual]
```

This call releases the referenced instance of the [System](#) Singleton.

After successfully releasing the [System](#) instance, the pointer returned by [GetInstance\(\)](#) will be invalid. Once the final [System](#) instance is released, all remaining [Spinnaker](#) resources will be released. If the final [System](#) instance is released while an interface or camera reference is still held, this function will throw an error of type SPINNAKE←R_ERR_RESOURCE_IN_USE.

See also

[Error](#)

[GetInstance\(\)](#)

Implements [ISystem](#).

15.176.3.11 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [virtual]
```

Broadcast an Action Command to all devices on system.

Parameters

<i>deviceKey</i>	The Action Command's device key
<i>groupKey</i>	The Action Command's group key
<i>groupMask</i>	The Action Command's group mask
<i>actionTime</i>	(Optional) Time when to assert a future action. Zero means immediate action.
<i>pResultSize</i>	(Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted.
<i>results</i>	(Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

Implements [ISystem](#).

15.176.3.12 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [virtual]
```

Sets a threshold priority level for logging event.

Logging events below such level will not trigger callbacks.

[Spinnaker](#) uses multiple levels of logging:

- Fatal - failures that are non-recoverable without user intervention.
 - Error - failures that may or may not be recoverable without user intervention (use case dependent).
 - Warning - failures that are recoverable without user intervention.
 - Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.
 - Info - information about recurring events that are generated regularly such as information on individual images.
 - Debug - information that can be used to troubleshoot the system.

See also

[SpinnakerLogLevel](#)

Parameters

<i>level</i>	The threshold level

Implements [ISystem](#).

15.176.3.13 UnregisterAllLoggingEventHandlers()

```
virtual void UnregisterAllLoggingEventHandlers ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements [ISystem](#).

15.176.3.14 UnregisterEventHandler()

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the system.

See also

[SystemEventHandler](#)

Parameters

<i>evtHandlerToUnregister</i>	The event handler to unregister from the system

Implements [ISystem](#).

15.176.3.15 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Unregisters a logging event.

Parameters

<i>handler</i>	The logging event handler to unregister

Implements [ISystem](#).

15.176.3.16 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [virtual]
```

Updates the list of cameras on the system.

Note that [System::GetCameras\(\)](#) internally calls [UpdateCameras\(\)](#) for each interface it enumerates. If the list changed between this call and the last time [UpdateCameras](#) was called then the return value will be true, otherwise it is false.

See also

[GetCameras\(\)](#)

Parameters

<i>updateInterfaces</i>	Determines whether or not UpdateInterfaceList() is called before updating cameras for available interfaces on the system

Returns

True if cameras changed on interface and false otherwise.

Implements [ISystem](#).

15.176.3.17 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [virtual]
```

Updates the list of interfaces on the system.

If desired, local copies of [InterfaceList](#) should be updated by calling `GetInterfaces`.

See also

[GetInterfaces\(\)](#)

Implements [ISystem](#).

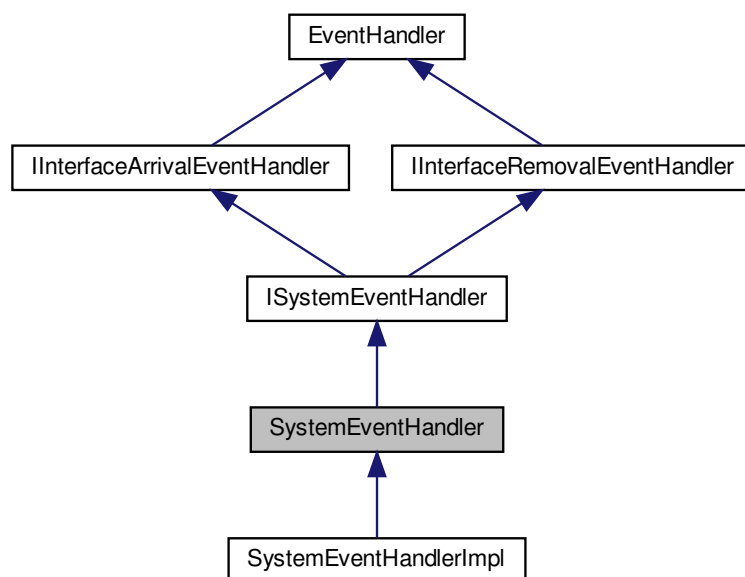
The documentation for this class was generated from the following file:

- `include/System.h`

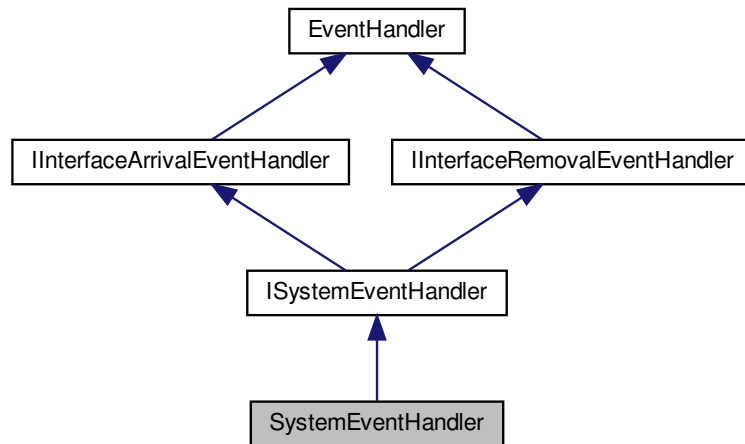
15.177 SystemEventHandler Class Reference

A handler to interface arrival and removal events on the system.

Inheritance diagram for SystemEventHandler:



Collaboration diagram for SystemEventHandler:



Public Member Functions

- [SystemEventHandler](#) ()
Default constructor.
- virtual [~SystemEventHandler](#) ()
Virtual destructor.
- virtual void [OnInterfaceArrival](#) ([InterfacePtr](#) pInterface)=0
Interface arrival event callback.
- virtual void [OnInterfaceRemoval](#) ([InterfacePtr](#) pInterface)=0
Interface removal event callback.

Protected Member Functions

- [SystemEventHandler](#) & [operator=](#) (const [SystemEventHandler](#) &)
Assignment operator.

Additional Inherited Members

15.177.1 Detailed Description

A handler to interface arrival and removal events on the system.

Note that only GEV interface arrivals and removals are currently handled.

15.177.2 Constructor & Destructor Documentation

15.177.2.1 SystemEventHandler()

```
SystemEventHandler ( )
```

Default constructor.

15.177.2.2 ~SystemEventHandler()

```
virtual ~SystemEventHandler ( ) [virtual]
```

Virtual destructor.

15.177.3 Member Function Documentation

15.177.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    InterfacePtr pInterface ) [pure virtual]
```

[Interface](#) arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter attached to the system

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

15.177.3.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    InterfacePtr pInterface ) [pure virtual]
```

[Interface](#) removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter removed from the system

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

15.177.3.3 operator=()

```
SystemEventHandler& operator= (
    const SystemEventHandler & ) [protected]
```

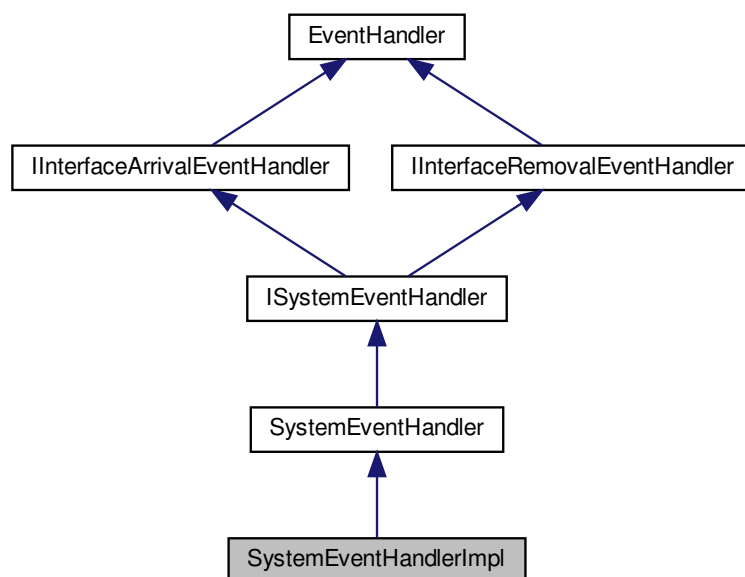
Assignment operator.

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

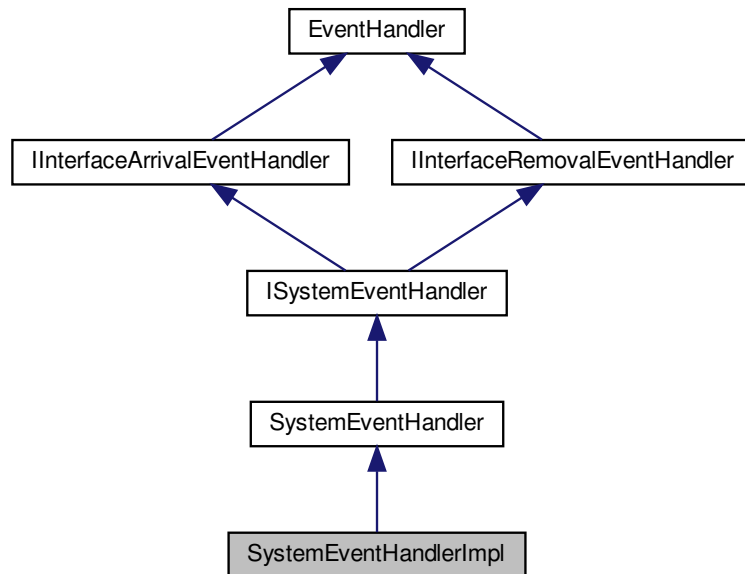
- include/[SystemEventHandler.h](#)

15.178 SystemEventHandlerImpl Class Reference

Inheritance diagram for SystemEventHandlerImpl:



Collaboration diagram for SystemEventHandlerImpl:



Public Member Functions

- [SystemEventHandlerImpl](#) ([SystemPtr](#) system)
- [~SystemEventHandlerImpl](#) ()
- void [LockEventHandlerMutex](#) ()
- void [UnlockEventHandlerMutex](#) ()
- void [OnInterfaceArrival](#) ([InterfacePtr](#) pInterface)
Interface arrival event callback.
- void [OnInterfaceRemoval](#) ([InterfacePtr](#) pInterface)
Interface removal event callback.
- void [RegisterInterfaceEventToSystem](#) ()
- void [UnregisterInterfaceEventFromSystem](#) ()
- void [RegisterAllInterfaceEvents](#) ()
- void [UnregisterAllInterfaceEvents](#) ()

Additional Inherited Members

15.178.1 Constructor & Destructor Documentation

15.178.1.1 SystemEventHandlerImpl()

```

SystemEventHandlerImpl (
    SystemPtr system ) [inline]
  
```

15.178.1.2 ~SystemEventHandlerImpl()

```
~SystemEventHandlerImpl ( ) [inline]
```

15.178.2 Member Function Documentation

15.178.2.1 LockEventHandlerMutex()

```
void LockEventHandlerMutex ( ) [inline]
```

15.178.2.2 OnInterfaceArrival()

```
void OnInterfaceArrival (
    InterfacePtr pInterface ) [inline], [virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter attached to the system

Implements [SystemEventHandler](#).

15.178.2.3 OnInterfaceRemoval()

```
void OnInterfaceRemoval (
    InterfacePtr pInterface ) [inline], [virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

Parameters

<i>pInterface</i>	Reference tracked pointer to Interface object of the adapter removed from the system

Implements [SystemEventHandler](#).

15.178.2.4 RegisterAllInterfaceEvents()

```
void RegisterAllInterfaceEvents ( ) [inline]
```

15.178.2.5 RegisterInterfaceEventToSystem()

```
void RegisterInterfaceEventToSystem ( ) [inline]
```

15.178.2.6 UnlockEventHandlerMutex()

```
void UnlockEventHandlerMutex ( ) [inline]
```

15.178.2.7 UnregisterAllInterfaceEvents()

```
void UnregisterAllInterfaceEvents ( ) [inline]
```

15.178.2.8 UnregisterInterfaceEventFromSystem()

```
void UnregisterInterfaceEventFromSystem ( ) [inline]
```

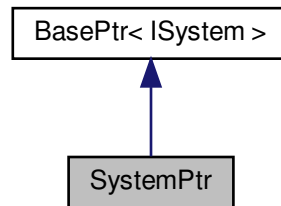
The documentation for this class was generated from the following file:

- src/EnumerationEvents/[EnumerationEvents.cpp](#)

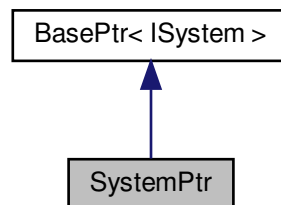
15.179 SystemPtr Class Reference

A reference tracked pointer to a system object.

Inheritance diagram for SystemPtr:



Collaboration diagram for SystemPtr:



Public Member Functions

- [SystemPtr](#) ()
Default constructor.
- [SystemPtr](#) (const int)
Copy constructor.
- [SystemPtr](#) (const long)
Copy constructor.
- [SystemPtr](#) (const std::nullptr_t)
Copy constructor.
- virtual [~SystemPtr](#) (void)
Virtual destructor.

Additional Inherited Members

15.179.1 Detailed Description

A reference tracked pointer to a system object.

15.179.2 Constructor & Destructor Documentation

15.179.2.1 SystemPtr() [1/4]

```
SystemPtr ( ) [inline]
```

Default constructor.

15.179.2.2 SystemPtr() [2/4]

```
SystemPtr (
    const int ) [inline]
```

Copy constructor.

15.179.2.3 SystemPtr() [3/4]

```
SystemPtr (
    const long ) [inline]
```

Copy constructor.

15.179.2.4 SystemPtr() [4/4]

```
SystemPtr (
    const std::nullptr_t ) [inline]
```

Copy constructor.

15.179.2.5 ~SystemPtr()

```
virtual ~SystemPtr (
    void ) [inline], [virtual]
```

Virtual destructor.

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

- include/SystemPtr.h

15.180 TIFFOption Struct Reference

Options for saving TIFF images.

Public Member Functions

- [TIFFOption](#) ()

Public Attributes

- [TIFFCompressionMethod](#) *compression*
Compression method to use for encoding TIFF images.
- unsigned int [reserved](#) [16]
Reserved for future use.

15.180.1 Detailed Description

Options for saving TIFF images.

15.180.2 Constructor & Destructor Documentation

15.180.2.1 TIFFOption()

```
TIFFOption ( ) [inline]
```

15.180.3 Member Data Documentation

15.180.3.1 compression

[TIFFCompressionMethod](#) compression

Compression method to use for encoding TIFF images.

15.180.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

15.181 TransportLayerDevice Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerDevice](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerDevice](#) ()

Public Attributes

- [GenApi::IString](#) & [DeviceID](#)
Description: Interface-wide unique identifier of this device.
- [GenApi::IString](#) & [DeviceSerialNumber](#)
Description: Serial number of the remote device.
- [GenApi::IString](#) & [DeviceUserID](#)
Description: User Defined Name.
- [GenApi::IString](#) & [DeviceVendorName](#)
Description: Name of the remote device vendor.
- [GenApi::IString](#) & [DeviceModelName](#)
Description: Name of the remote device model.
- [GenApi::IString](#) & [DeviceVersion](#)
Description: Version of the device.
- [GenApi::Integer](#) & [DeviceBootloaderVersion](#)
Description: Bootloader version for FLIR devices.
- [GenApi::IEnumerationT](#) < [DeviceTypeEnum](#) > & [DeviceType](#)
Description: Transport layer type of the device.
- [GenApi::IString](#) & [DeviceDisplayName](#)
Description: User readable name of the device.

- [GenApi::IEnumerationT< DeviceAccessStatusEnum > & DeviceAccessStatus](#)
Description: Gets the access status the transport layer Producer has on the device.
- [GenApi::Integer & DeviceLinkSpeed](#)
Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.
- [GenApi::IString & DeviceDriverVersion](#)
Description: Version of the device driver.
- [GenApi::IBoolean & DeviceIsUpdater](#)
Description: Indicates whether the device is in updater mode.
- [GenApi::IEnumerationT< GenICamXMLLocationEnum > & GenICamXMLLocation](#)
Description: Sets the location to load [GenICam](#) XML.
- [GenApi::IString & GenICamXMLPath](#)
Description: [GenICam](#) XML Path.
- [GenApi::IEnumerationT< GUIXMLLocationEnum > & GUIXMLLocation](#)
Description: Sets the location to load GUI XML.
- [GenApi::IString & GUIXMLPath](#)
Description: GUI XML Path.
- [GenApi::IEnumerationT< GevCCPEnum > & GevCCP](#)
Description: Controls the device access privilege of an application.
- [GenApi::Integer & GevDeviceMACAddress](#)
Description: 48-bit MAC address of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceIPAddress](#)
Description: Current IP address of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceSubnetMask](#)
Description: Current subnet mask of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceGateway](#)
Description: Current gateway IP address of the GVCP interface of the remote device.
- [GenApi::Integer & GevVersionMajor](#)
Description: Major version of the specification.
- [GenApi::Integer & GevVersionMinor](#)
Description: Minor version of the specification.
- [GenApi::IBoolean & GevDeviceModelsBigEndian](#)
Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).
- [GenApi::Integer & GevDeviceReadAndWriteTimeout](#)
Description: The timeout in us for read/write operations to the camera.
- [GenApi::Integer & GevDeviceMaximumRetryCount](#)
Description: Maximum number of times to retry a read/write operation.
- [GenApi::Integer & GevDevicePort](#)
Description: Current IP port of the GVCP interface of the selected remote device.
- [GenApi::ICommand & GevDeviceDiscoverMaximumPacketSize](#)
Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.
- [GenApi::Integer & GevDeviceMaximumPacketSize](#)
Description: The maximum packet size that can be safely used by the device on the current interface.
- [GenApi::IBoolean & GevDeviceIsWrongSubnet](#)
Description: Indicates whether the device is on the wrong subnet.
- [GenApi::ICommand & GevDeviceAutoForceIP](#)
Description: Forces the camera to be on the same subnet as its corresponding interface.
- [GenApi::ICommand & GevDeviceForceIP](#)
Description: Apply the force IP settings ([GevDeviceForceIPAddress](#), [GevDeviceForceSubnetMask](#) and [GevDeviceForceGateway](#)) in the Device using [ForceIP](#) command.

- [GenApi::Integer](#) & [GevDeviceForceIPAddress](#)
Description: Static IP address to set for the GVCP interface of the remote device.
- [GenApi::Integer](#) & [GevDeviceForceSubnetMask](#)
Description: Static subnet mask to set for GVCP interface of the remote device.
- [GenApi::Integer](#) & [GevDeviceForceGateway](#)
Description: Static gateway IP address to set for the GVCP interface of the remote device.
- [GenApi::Boolean](#) & [DeviceMulticastMonitorMode](#)
Description: Controls and indicates if the device is operating in as a Multicast Monitor.
- [GenApi::EnumerationT< DeviceEndiannessMechanismEnum >](#) & [DeviceEndiannessMechanism](#)
Description: Identifies the endianness handling mode.
- [GenApi::Command](#) & [DeviceReset](#)
Description: Resets and reboots the device.
- [GenApi::String](#) & [DeviceInstanceId](#)
Description: Device Instance Id.
- [GenApi::String](#) & [DeviceLocation](#)
Description: Device Location.
- [GenApi::EnumerationT< DeviceCurrentSpeedEnum >](#) & [DeviceCurrentSpeed](#)
Description: The USB Speed that the device is currently operating at.
- [GenApi::Boolean](#) & [DeviceU3VProtocol](#)
Description: Indicates whether the device is communicating in U3V Protocol.
- [GenApi::String](#) & [DevicePortId](#)
Description: Device Port ID.

Protected Member Functions

- [TransportLayerDevice](#) ()

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

15.181.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.181.2 Constructor & Destructor Documentation

15.181.2.1 TransportLayerDevice() [1/2]

```
TransportLayerDevice (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.181.2.2 ~TransportLayerDevice()

```
~TransportLayerDevice ( )
```

15.181.2.3 TransportLayerDevice() [2/2]

```
TransportLayerDevice ( ) [protected]
```

15.181.3 Friends And Related Function Documentation

15.181.3.1 CameraBase

```
friend class CameraBase [friend]
```

15.181.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

15.181.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

15.181.4 Member Data Documentation

15.181.4.1 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

15.181.4.2 DeviceBootloaderVersion

`GenApi::Integer& DeviceBootloaderVersion`

Description: Bootloader version for FLIR devices.

Visibility: Expert

15.181.4.3 DeviceCurrentSpeed

`GenApi::EnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed`

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

15.181.4.4 DeviceDisplayName

`GenApi::String& DeviceDisplayName`

Description: User readable name of the device.

If this is not defined in the device this should be "VENDOR MODEL (ID)". Visibility: Expert

15.181.4.5 DeviceDriverVersion

`GenApi::String& DeviceDriverVersion`

Description: Version of the device driver.

Visibility: Expert

15.181.4.6 DeviceEndiannessMechanism

`GenApi::EnumerationT<DeviceEndiannessMechanismEnum>& DeviceEndiannessMechanism`

Description: Identifies the endianness handling mode.

Visibility: Expert

15.181.4.7 DeviceID

`GenApi::String& DeviceID`

Description: Interface-wide unique identifier of this device.

Visibility: Expert

15.181.4.8 DeviceInstanceId

`GenApi::IString& DeviceInstanceId`

Description: Device Instance Id.

Visibility: Invisible

15.181.4.9 DeviceIsUpdater

`GenApi::IBoolean& DeviceIsUpdater`

Description: Indicates whether the device is in updater mode.

Visibility: Expert

15.181.4.10 DeviceLinkSpeed

`GenApi::IInteger& DeviceLinkSpeed`

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

15.181.4.11 DeviceLocation

`GenApi::IString& DeviceLocation`

Description: Device Location.

Visibility: Expert

15.181.4.12 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the remote device model.

Visibility: Beginner

15.181.4.13 DeviceMulticastMonitorMode

`GenApi::IBoolean& DeviceMulticastMonitorMode`

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

15.181.4.14 DevicePortId

`GenApi::IString& DevicePortId`

Description: Device Port ID.

Visibility: Expert

15.181.4.15 DeviceReset

`GenApi::ICommand& DeviceReset`

Description: Resets and reboots the device.

Visibility: Beginner

15.181.4.16 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the remote device.

Visibility: Expert

15.181.4.17 DeviceType

`GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType`

Description: Transport layer type of the device.

Visibility: Expert

15.181.4.18 DeviceU3VProtocol

`GenApi::IBoolean& DeviceU3VProtocol`

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

15.181.4.19 DeviceUserID

`GenApi::IString& DeviceUserID`

Description: User Defined Name.

Visibility: Expert

15.181.4.20 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the remote device vendor.

Visibility: Beginner

15.181.4.21 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility: Beginner

15.181.4.22 GenICamXMLLocation

`GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation`

Description: Sets the location to load [GenICam](#) XML.

Visibility: Beginner

15.181.4.23 GenICamXMLPath

`GenApi::IString& GenICamXMLPath`

Description: [GenICam](#) XML Path.

Visibility: Beginner

15.181.4.24 GevCCP

`GenApi::IEnumerationT<GevCCPEnum>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility: Beginner

15.181.4.25 GevDeviceAutoForceIP

`GenApi::ICommand& GevDeviceAutoForceIP`

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

15.181.4.26 **GevDeviceDiscoverMaximumPacketSize**

`GenApi::ICommand& GevDeviceDiscoverMaximumPacketSize`

Description: Discovers and updates the maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

15.181.4.27 **GevDeviceForceGateway**

`GenApi::IInteger& GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the remote device.

Visibility: Expert

15.181.4.28 **GevDeviceForceIP**

`GenApi::ICommand& GevDeviceForceIP`

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the Device using ForceIP command.

Visibility: Expert

15.181.4.29 **GevDeviceForceIPAddress**

`GenApi::IInteger& GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the remote device.

Visibility: Expert

15.181.4.30 **GevDeviceForceSubnetMask**

`GenApi::IInteger& GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the remote device.

Visibility: Expert

15.181.4.31 **GevDeviceGateway**

`GenApi::IInteger& GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

15.181.4.32 GevDeviceIPAddress

`GenApi::Integer&   GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.181.4.33 GevDeviceIsWrongSubnet

`GenApi::Boolean&   GevDeviceIsWrongSubnet`

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

15.181.4.34 GevDeviceMACAddress

`GenApi::Integer&   GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

15.181.4.35 GevDeviceMaximumPacketSize

`GenApi::Integer&   GevDeviceMaximumPacketSize`

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

15.181.4.36 GevDeviceMaximumRetryCount

`GenApi::Integer&   GevDeviceMaximumRetryCount`

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

15.181.4.37 GevDeviceModeIsBigEndian

`GenApi::Boolean&   GevDeviceModeIsBigEndian`

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

15.181.4.38 GevDevicePort

`GenApi::Integer& GevDevicePort`

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

15.181.4.39 GevDeviceReadAndWriteTimeout

`GenApi::Integer& GevDeviceReadAndWriteTimeout`

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

15.181.4.40 GevDeviceSubnetMask

`GenApi::Integer& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

15.181.4.41 GevVersionMajor

`GenApi::Integer& GevVersionMajor`

Description: Major version of the specification.

Visibility: Expert

15.181.4.42 GevVersionMinor

`GenApi::Integer& GevVersionMinor`

Description: Minor version of the specification.

Visibility: Expert

15.181.4.43 GUIXMLLocation

`GenApi::EnumerationT<GUIXMLLocationEnum>& GUIXMLLocation`

Description: Sets the location to load GUI XML.

Visibility: Beginner

15.181.4.44 GUIXMLPath

`GenApi::IString& GUIXMLPath`

Description: GUI XML Path.

Visibility: Beginner

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

- include/[TransportLayerDevice.h](#)

15.182 TransportLayerInterface Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerInterface](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerInterface](#) ()

Public Attributes

- [GenApi::IString](#) & [InterfaceID](#)
Description: Transport layer Producer wide unique identifier of the selected interface.
- [GenApi::IString](#) & [InterfaceDisplayName](#)
Description: User readable name of the selected interface.
- [GenApi::IEnumerationT](#) < [InterfaceTypeEnum](#) > & [InterfaceType](#)
Description: Transport layer type of the interface.
- [GenApi::Integer](#) & [GevInterfaceGatewaySelector](#)
Description: Selector for the different gateway entries for this interface.
- [GenApi::Integer](#) & [GevInterfaceGateway](#)
Description: IP address of the selected gateway entry of this interface.
- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)
Description: 48-bit MAC address of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetSelector](#)
Description: Selector for the subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetIPAddress](#)
Description: IP address of the selected subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetMask](#)
Description: Subnet mask of the selected subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceTransmitLinkSpeed](#)
Description: Transmit link speed of this interface in bits per second.
- [GenApi::Integer](#) & [GevInterfaceReceiveLinkSpeed](#)
Description: Receive link speed of this interface in bits per second.
- [GenApi::Integer](#) & [GevInterfaceMTU](#)
Description: Maximum transmission unit of this interface.
- [GenApi::IEnumerationT](#) < [POEStatusEnum](#) > & [POEStatus](#)

- Description: Reports and controls the interface's power over Ethernet status.*

 - [GenApi::IEnumerationT < FilterDriverStatusEnum > & FilterDriverStatus](#)

Description: Reports whether FLIR Light Weight Filter Driver is enabled, disabled, or not installed.
- [GenApi::Integer & GevActionDeviceKey](#)

Description: Key to authorize the action for the device.
- [GenApi::Integer & GevActionGroupKey](#)

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer & GevActionGroupMask](#)

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer & GevActionTime](#)

Description: Provides the time in nanoseconds when the action is to be executed.
- [GenApi::ICommand & ActionCommand](#)

Description: Issues an Action Command to attached GEV devices on interface.
- [GenApi::IString & DeviceUnlock](#)

Description: Unlocks devices for internal use.
- [GenApi::ICommand & DeviceUpdateList](#)

Description: Updates the internal device list.
- [GenApi::Integer & DeviceCount](#)

Description: Number of compatible devices detected on current interface.
- [GenApi::Integer & DeviceSelector](#)

Description: Selector for the different devices on this interface.
- [GenApi::IString & DeviceID](#)

Description: [Interface](#) wide unique identifier of the selected device.
- [GenApi::IString & DeviceVendorName](#)

Description: Name of the device vendor.
- [GenApi::IString & DeviceModelName](#)

Description: Name of the device model.
- [GenApi::IString & DeviceSerialNumber](#)

Description: Serial number of the selected remote device.
- [GenApi::IEnumerationT < DeviceAccessStatusEnum > & DeviceAccessStatus](#)

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".
- [GenApi::Integer & GevDeviceIPAddress](#)

Description: Current IP address of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceSubnetMask](#)

Description: Current subnet mask of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceGateway](#)

Description: Current gateway IP address of the GVCP interface of the selected remote device.
- [GenApi::Integer & GevDeviceMACAddress](#)

Description: 48-bit MAC address of the GVCP interface of the selected remote device.
- [GenApi::Integer & IncompatibleDeviceCount](#)

Description: Number of incompatible devices detected on current interface.
- [GenApi::Integer & IncompatibleDeviceSelector](#)

Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.
- [GenApi::IString & IncompatibleDeviceID](#)

Description: [Interface](#) wide unique identifier of the selected incompatible device.
- [GenApi::IString & IncompatibleDeviceVendorName](#)

Description: Name of the incompatible device vendor.
- [GenApi::IString & IncompatibleDeviceModelName](#)

Description: Name of the incompatible device model.

- [GenApi::Integer](#) & [IncompatibleGevDeviceIPAddress](#)
Description: Current IP address of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceSubnetMask](#)
Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.
- [GenApi::Integer](#) & [IncompatibleGevDeviceMACAddress](#)
Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.
- [GenApi::ICommand](#) & [GevDeviceForceIP](#)
Description: Apply the force IP settings ([GevDeviceForceIPAddress](#), [GevDeviceForceSubnetMask](#) and [GevDeviceForceGateway](#)) in the selected remote device using ForceIP command.
- [GenApi::Integer](#) & [GevDeviceForceIPAddress](#)
Description: Static IP address to set for the GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceForceSubnetMask](#)
Description: Static subnet mask to set for GVCP interface of the selected remote device.
- [GenApi::Integer](#) & [GevDeviceForceGateway](#)
Description: Static gateway IP address to set for the GVCP interface of the selected remote device.
- [GenApi::ICommand](#) & [GevDeviceAutoForceIP](#)
Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.
- [GenApi::IString](#) & [HostAdapterName](#)
Description: User readable name of the interface's host adapter.
- [GenApi::IString](#) & [HostAdapterVendor](#)
Description: User readable name of the host adapter's vendor.
- [GenApi::IString](#) & [HostAdapterDriverVersion](#)
Description: Driver version of the interface's host adapter.

Protected Member Functions

- [TransportLayerInterface](#) ()

Friends

- class [Interface](#)
- class [IInterface](#)
- class [InterfaceInternal](#)

15.182.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.182.2 Constructor & Destructor Documentation

15.182.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.182.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

15.182.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

15.182.3 Friends And Related Function Documentation

15.182.3.1 IInterface

```
friend class IInterface [friend]
```

15.182.3.2 Interface

```
friend class Interface [friend]
```

15.182.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

15.182.4 Member Data Documentation

15.182.4.1 ActionCommand

```
GenApi::ICommand& ActionCommand
```

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

15.182.4.2 DeviceAccessStatus

`GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus`

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.3 DeviceCount

`GenApi::IInteger& DeviceCount`

Description: Number of compatible devices detected on current interface.

Visibility: Expert

15.182.4.4 DeviceID

`GenApi::IString& DeviceID`

Description: [Interface](#) wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.5 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.6 DeviceSelector

`GenApi::IInteger& DeviceSelector`

Description: Selector for the different devices on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

15.182.4.7 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the selected remote device.

Visibility: Expert

15.182.4.8 DeviceUnlock

`GenApi::IString& DeviceUnlock`

Description: Unlocks devices for internal use.

Visibility: Expert

15.182.4.9 DeviceUpdateList

`GenApi::ICommand& DeviceUpdateList`

Description: Updates the internal device list.

Visibility: Expert

15.182.4.10 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.11 FilterDriverStatus

`GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus`

Description: Reports whether FLIR Light Weight Filter Driver is enabled, disabled, or not installed.

Visibility: Expert

15.182.4.12 GevActionDeviceKey

`GenApi::IInteger& GevActionDeviceKey`

Description: Key to authorize the action for the device.

Visibility: Expert

15.182.4.13 GevActionGroupKey

`GenApi::IInteger& GevActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

15.182.4.14 **GevActionGroupMask**

`GenApi::Integer& GevActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

15.182.4.15 **GevActionTime**

`GenApi::Integer& GevActionTime`

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

15.182.4.16 **GevDeviceAutoForceIP**

`GenApi::ICommand& GevDeviceAutoForceIP`

Description: Automatically forces the selected remote device to an IP Address on the same subnet as the GVCP interface.

Visibility: Expert

15.182.4.17 **GevDeviceForceGateway**

`GenApi::Integer& GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.18 **GevDeviceForceIP**

`GenApi::ICommand& GevDeviceForceIP`

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the selected remote device using ForceIP command.

Visibility: Expert

15.182.4.19 **GevDeviceForceIPAddress**

`GenApi::Integer& GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.20 **GevDeviceForceSubnetMask**

`GenApi::Integer& GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.21 **GevDeviceGateway**

`GenApi::Integer& GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.22 **GevDeviceIPAddress**

`GenApi::Integer& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.23 **GevDeviceMACAddress**

`GenApi::Integer& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.24 **GevDeviceSubnetMask**

`GenApi::Integer& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

15.182.4.25 **GevInterfaceGateway**

`GenApi::Integer& GevInterfaceGateway`

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

15.182.4.26 GevInterfaceGatewaySelector

`GenApi::Integer&   GevInterfaceGatewaySelector`

Description: Selector for the different gateway entries for this interface.

Visibility: Expert

15.182.4.27 GevInterfaceMACAddress

`GenApi::Integer&   GevInterfaceMACAddress`

Description: 48-bit MAC address of this interface.

Visibility: Expert

15.182.4.28 GevInterfaceMTU

`GenApi::Integer&   GevInterfaceMTU`

Description: Maximum transmission unit of this interface.

Visibility: Expert

15.182.4.29 GevInterfaceReceiveLinkSpeed

`GenApi::Integer&   GevInterfaceReceiveLinkSpeed`

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

15.182.4.30 GevInterfaceSubnetIPAddress

`GenApi::Integer&   GevInterfaceSubnetIPAddress`

Description: IP address of the selected subnet of this interface.

Visibility: Expert

15.182.4.31 GevInterfaceSubnetMask

`GenApi::Integer&   GevInterfaceSubnetMask`

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

15.182.4.32 GevInterfaceSubnetSelector

`GenApi::Integer&   GevInterfaceSubnetSelector`

Description: Selector for the subnet of this interface.

Visibility: Expert

15.182.4.33 GevInterfaceTransmitLinkSpeed

`GenApi::Integer&   GevInterfaceTransmitLinkSpeed`

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

15.182.4.34 HostAdapterDriverVersion

`GenApi::String&   HostAdapterDriverVersion`

Description: Driver version of the interface's host adapter.

Visibility: Expert

15.182.4.35 HostAdapterName

`GenApi::String&   HostAdapterName`

Description: User readable name of the interface's host adapter.

Visibility: Expert

15.182.4.36 HostAdapterVendor

`GenApi::String&   HostAdapterVendor`

Description: User readable name of the host adapter's vendor.

Visibility: Expert

15.182.4.37 IncompatibleDeviceCount

`GenApi::Integer&   IncompatibleDeviceCount`

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

15.182.4.38 IncompatibleDeviceID

`GenApi::IString& IncompatibleDeviceID`

Description: [Interface](#) wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.39 IncompatibleDeviceModelName

`GenApi::IString& IncompatibleDeviceModelName`

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.40 IncompatibleDeviceSelector

`GenApi::IInteger& IncompatibleDeviceSelector`

Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.

This value only changes on execution of "DeviceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Expert

15.182.4.41 IncompatibleDeviceVendorName

`GenApi::IString& IncompatibleDeviceVendorName`

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

15.182.4.42 IncompatibleGevDeviceIPAddress

`GenApi::IInteger& IncompatibleGevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.182.4.43 IncompatibleGevDeviceMACAddress

`GenApi::IInteger& IncompatibleGevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.182.4.44 IncompatibleGevDeviceSubnetMask

`GenApi::Integer& IncompatibleGevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

15.182.4.45 InterfaceDisplayName

`GenApi::IString& InterfaceDisplayName`

Description: User readable name of the selected interface.

Visibility: Expert

15.182.4.46 InterfaceID

`GenApi::IString& InterfaceID`

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

15.182.4.47 InterfaceType

`GenApi::IEnumerationT<InterfaceTypeEnum>& InterfaceType`

Description: Transport layer type of the interface.

Visibility: Expert

15.182.4.48 POEStatus

`GenApi::IEnumerationT<POEStatusEnum>& POEStatus`

Description: Reports and controls the interface's power over Ethernet status.

Visibility: Expert

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

- [include/TransportLayerInterface.h](#)

15.183 TransportLayerStream Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerStream](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerStream](#) ()

Public Attributes

- [GenApi::IString](#) & [StreamID](#)
Description: Device unique ID for the data stream, e.g.
- [GenApi::IEnumerationT](#)< [StreamTypeEnum](#) > & [StreamType](#)
Description: Stream type of the device.
- [GenApi::IEnumerationT](#)< [StreamModeEnum](#) > & [StreamMode](#)
Description: Stream mode of the device.
- [GenApi::IInteger](#) & [StreamBufferCountManual](#)
Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.
- [GenApi::IInteger](#) & [StreamBufferCountResult](#)
Description: Displays the number of buffers to be used on this stream upon acquisition start.
- [GenApi::IInteger](#) & [StreamBufferCountMax](#)
Description: Controls the maximum number of buffers that should be used on this stream.
- [GenApi::IEnumerationT](#)< [StreamBufferCountModeEnum](#) > & [StreamBufferCountMode](#)
Description: Controls access to setting the number of buffers used for the stream.
- [GenApi::IEnumerationT](#)< [StreamBufferHandlingModeEnum](#) > & [StreamBufferHandlingMode](#)
Description: Available buffer handling modes of this data stream: Visibility: Beginner.
- [GenApi::IInteger](#) & [StreamAnnounceBufferMinimum](#)
Description: Minimal number of buffers to announce to enable selected buffer handling mode.
- [GenApi::IInteger](#) & [StreamAnnouncedBufferCount](#)
Description: Number of announced (known) buffers on this stream.
- [GenApi::IInteger](#) & [StreamStartedFrameCount](#)
Description: Number of frames started in the acquisition engine.
- [GenApi::IInteger](#) & [StreamDeliveredFrameCount](#)
Description: Number of delivered frames since last acquisition start.
- [GenApi::IInteger](#) & [StreamReceivedFrameCount](#)
Description: Number of successful GVSP data blocks received.
- [GenApi::IInteger](#) & [StreamIncompleteFrameCount](#)
Description: Displays number of images with missing packet.
- [GenApi::IInteger](#) & [StreamLostFrameCount](#)
Description: Number of times new data could not be acquired and was lost because there was no free buffer in the input buffer pool, or the frame data was never received fully due to data lost.
- [GenApi::IInteger](#) & [StreamDroppedFrameCount](#)
Description: Number of frames dropped from the output buffer queue before being processed by the application.
- [GenApi::IInteger](#) & [StreamInputBufferCount](#)
Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.
- [GenApi::IInteger](#) & [StreamOutputBufferCount](#)
Description: Number of buffers in the output buffer queue.
- [GenApi::IBoolean](#) & [StreamIsGrabbing](#)
Description: Flag indicating whether the acquisition engine is started or not.
- [GenApi::IInteger](#) & [StreamChunkCountMaximum](#)
Description: Maximum number of chunks to be expected in a buffer.
- [GenApi::IInteger](#) & [StreamBufferAlignment](#)
Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.
- [GenApi::IBoolean](#) & [StreamCRCCheckEnable](#)

- Description: Enables or disables CRC checks on received images.*
- [GenApi::Integer](#) & [StreamReceivedPacketCount](#)
Description: Displays number of packets received on this stream.
- [GenApi::Integer](#) & [StreamMissedPacketCount](#)
Description: Displays number of packets missed by this stream.
- [GenApi::Boolean](#) & [StreamPacketResendEnable](#)
Description: Enables or disables the packet resend mechanism.
- [GenApi::Integer](#) & [StreamPacketResendTimeout](#)
Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.
- [GenApi::Integer](#) & [StreamPacketResendMaxRequests](#)
Description: Maximum number of resend requests per image.
- [GenApi::Integer](#) & [StreamPacketResendRequestCount](#)
Description: Displays number of packet resend requests transmitted to the camera.
- [GenApi::Integer](#) & [StreamPacketResendRequestSuccessCount](#)
Description: Displays number of packet resend requests successfully transmitted to the camera.
- [GenApi::Integer](#) & [StreamPacketResendRequestedPacketCount](#)
Description: Displays number of packets requested to be retransmitted on this stream.
- [GenApi::Integer](#) & [StreamPacketResendReceivedPacketCount](#)
Description: Displays number of retransmitted packets received on this stream.
- [GenApi::Integer](#) & [StreamBlockTransferSize](#)
Description: Controls the image breakup size that should be used on this stream.

Protected Member Functions

- [TransportLayerStream](#) ()

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

15.183.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.183.2 Constructor & Destructor Documentation

15.183.2.1 TransportLayerStream() [1/2]

```
TransportLayerStream (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.183.2.2 ~TransportLayerStream()

```
~TransportLayerStream ( )
```

15.183.2.3 TransportLayerStream() [2/2]

```
TransportLayerStream ( ) [protected]
```

15.183.3 Friends And Related Function Documentation

15.183.3.1 CameraBase

```
friend class CameraBase [friend]
```

15.183.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

15.183.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

15.183.4 Member Data Documentation

15.183.4.1 StreamAnnounceBufferMinimum

```
GenApi::Integer& StreamAnnounceBufferMinimum
```

Description: Minimal number of buffers to announce to enable selected buffer handling mode.

Visibility: Expert

15.183.4.2 StreamAnnouncedBufferCount

`GenApi::Integer& StreamAnnouncedBufferCount`

Description: Number of announced (known) buffers on this stream.

This value is volatile. It may change if additional buffers are announced and/or buffers are revoked by the GenTL Consumer. Visibility: Expert

15.183.4.3 StreamBlockTransferSize

`GenApi::Integer& StreamBlockTransferSize`

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

15.183.4.4 StreamBufferAlignment

`GenApi::Integer& StreamBufferAlignment`

Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.

Visibility: Expert

15.183.4.5 StreamBufferCountManual

`GenApi::Integer& StreamBufferCountManual`

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

15.183.4.6 StreamBufferCountMax

`GenApi::Integer& StreamBufferCountMax`

Description: Controls the maximum number of buffers that should be used on this stream.

This value is calculated based on the available system memory. Visibility: Expert

15.183.4.7 StreamBufferCountMode

`GenApi::EnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode`

Description: Controls access to setting the number of buffers used for the stream.

Visibility: Expert

15.183.4.8 StreamBufferCountResult

`GenApi::Integer& StreamBufferCountResult`

Description: Displays the number of buffers to be used on this stream upon acquisition start.

Recalculated on acquisition start if in auto mode. Visibility: Expert

15.183.4.9 StreamBufferHandlingMode

`GenApi::EnumerationT<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode`

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

15.183.4.10 StreamChunkCountMaximum

`GenApi::Integer& StreamChunkCountMaximum`

Description: Maximum number of chunks to be expected in a buffer.

Visibility: Expert

15.183.4.11 StreamCRCCheckEnable

`GenApi::Boolean& StreamCRCCheckEnable`

Description: Enables or disables CRC checks on received images.

Visibility: Expert

15.183.4.12 StreamDeliveredFrameCount

`GenApi::Integer& StreamDeliveredFrameCount`

Description: Number of delivered frames since last acquisition start.

It is not reset until the stream is closed. Visibility: Expert

15.183.4.13 StreamDroppedFrameCount

`GenApi::Integer& StreamDroppedFrameCount`

Description: Number of frames dropped from the output buffer queue before being processed by the application.

This count usually indicates buffer starvation. To reduce dropped frame count, improve the application's image processing speed, release the buffer (`image->Release()`) as soon as it is not needed, or add more buffers before streaming. This number is initialized with zero when stream is opened and incremented when old data is dropped. It is reset when the stream is closed. Visibility: Expert

15.183.4.14 StreamID

`GenApi::IString& StreamID`

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

15.183.4.15 StreamIncompleteFrameCount

`GenApi::IInteger& StreamIncompleteFrameCount`

Description: Displays number of images with missing packet.

Visibility: Expert

15.183.4.16 StreamInputBufferCount

`GenApi::IInteger& StreamInputBufferCount`

Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.

Visibility: Expert

15.183.4.17 StreamIsGrabbing

`GenApi::IBoolean& StreamIsGrabbing`

Description: Flag indicating whether the acquisition engine is started or not.

Visibility: Expert

15.183.4.18 StreamLostFrameCount

`GenApi::IInteger& StreamLostFrameCount`

Description: Number of times new data could not be acquired and was lost because there was no free buffer in the input buffer pool, or the frame data was never received fully due to data lost.

This count usually indicates buffer starvation as well as some other underlying issues causing data lost in transit from the imaging device to the host. To reduce lost frame count, improve the application's image processing speed, release the buffer (`image->Release()`) as soon as it is not needed, improve the data connection, or use buffer handling modes such as Oldest First Overwrite or Newest First. This number is initialized with zero when the stream is opened and incremented when new data is lost. It is reset when the stream is closed. Visibility: Expert

15.183.4.19 StreamMissedPacketCount

`GenApi::IInteger& StreamMissedPacketCount`

Description: Displays number of packets missed by this stream.

Visibility: Expert

15.183.4.20 StreamMode

`GenApi::IEnumerationT<StreamModeEnum>& StreamMode`

Description: Stream mode of the device.

Visibility: Expert

15.183.4.21 StreamOutputBufferCount

`GenApi::IInteger& StreamOutputBufferCount`

Description: Number of buffers in the output buffer queue.

Visibility: Expert

15.183.4.22 StreamPacketResendEnable

`GenApi::IBoolean& StreamPacketResendEnable`

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

15.183.4.23 StreamPacketResendMaxRequests

`GenApi::IInteger& StreamPacketResendMaxRequests`

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

15.183.4.24 StreamPacketResendReceivedPacketCount

`GenApi::IInteger& StreamPacketResendReceivedPacketCount`

Description: Displays number of retransmitted packets received on this stream.

Visibility: Expert

15.183.4.25 StreamPacketResendRequestCount

`GenApi::IInteger& StreamPacketResendRequestCount`

Description: Displays number of packet resend requests transmitted to the camera.

Visibility: Expert

15.183.4.26 StreamPacketResendRequestedPacketCount

`GenApi::Integer& StreamPacketResendRequestedPacketCount`

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

15.183.4.27 StreamPacketResendRequestSuccessCount

`GenApi::Integer& StreamPacketResendRequestSuccessCount`

Description: Displays number of packet resend requests successfully transmitted to the camera.

Visibility: Expert

15.183.4.28 StreamPacketResendTimeout

`GenApi::Integer& StreamPacketResendTimeout`

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

15.183.4.29 StreamReceivedFrameCount

`GenApi::Integer& StreamReceivedFrameCount`

Description: Number of successful GVSP data blocks received.

Only valid while stream is active. Visibility: Expert

15.183.4.30 StreamReceivedPacketCount

`GenApi::Integer& StreamReceivedPacketCount`

Description: Displays number of packets received on this stream.

Visibility: Expert

15.183.4.31 StreamStartedFrameCount

`GenApi::Integer& StreamStartedFrameCount`

Description: Number of frames started in the acquisition engine.

This number is incremented every time in case of a new buffer is started and then to be filled (data written to) regardless even if the buffer is later delivered to the user or discarded for any reason. This number is initialized with 0 at the time of the stream is opened. It is not reset until the stream is closed. Visibility: Expert

15.183.4.32 StreamType

`GenApi::IEnumerationT<StreamTypeEnum>& StreamType`

Description: Stream type of the device.

Visibility: Expert

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

- include/[TransportLayerStream.h](#)

15.184 TransportLayerSystem Class Reference

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Public Member Functions

- [TransportLayerSystem](#) ([GenApi::INodeMap](#) *nodeMapTLDevice)
- [~TransportLayerSystem](#) ()

Public Attributes

- [GenApi::IString](#) & [TLID](#)
Description: Unique identifier of the GenTL Producer like a GUID.
- [GenApi::IString](#) & [TLVendorName](#)
Description: Name of the GenTL Producer vendor.
- [GenApi::IString](#) & [TLModelName](#)
Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.
- [GenApi::IString](#) & [TLVersion](#)
Description: Vendor specific version string.
- [GenApi::IString](#) & [TLFileName](#)
Description: Filename including extension of the GenTL Producer.
- [GenApi::IString](#) & [TLDisplayName](#)
Description: User readable name of the GenTL Producer.
- [GenApi::IString](#) & [TLPath](#)
Description: Full path to the GenTL Producer including filename and extension.
- [GenApi::IEnumerationT< TLTypeEnum >](#) & [TLType](#)
Description: Transport layer type of the GenTL Producer implementation.
- [GenApi::Integer](#) & [GenTLVersionMajor](#)
Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.
- [GenApi::Integer](#) & [GenTLVersionMinor](#)
Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.
- [GenApi::Integer](#) & [GenTLSFNCVersionMajor](#)
Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.
- [GenApi::Integer](#) & [GenTLSFNCVersionMinor](#)

Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- [GenApi::Integer](#) & [GenTLSFNCVersionSubMinor](#)

Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- [GenApi::Integer](#) & [GevVersionMajor](#)

Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

- [GenApi::Integer](#) & [GevVersionMinor](#)

Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.

- [GenApi::ICommand](#) & [InterfaceUpdateList](#)

Description: Updates the internal list of the interfaces.

- [GenApi::Integer](#) & [InterfaceSelector](#)

Description: Selector for the different GenTL Producer interfaces.

- [GenApi::IString](#) & [InterfaceID](#)

Description: GenTL Producer wide unique identifier of the selected interface.

- [GenApi::IString](#) & [InterfaceDisplayName](#)

Description: A user-friendly name of the selected [Interface](#).

- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)

Description: 48-bit MAC address of the selected interface.

- [GenApi::Integer](#) & [GevInterfaceDefaultIPAddress](#)

Description: IP address of the first subnet of the selected interface.

- [GenApi::Integer](#) & [GevInterfaceDefaultSubnetMask](#)

Description: Subnet mask of the first subnet of the selected interface.

- [GenApi::Integer](#) & [GevInterfaceDefaultGateway](#)

Description: Gateway of the selected interface.

- [GenApi::IBoolean](#) & [EnumerateGEVInterfaces](#)

Description: Enables or disables enumeration of GEV Interfaces.

- [GenApi::IBoolean](#) & [EnumerateUSBInterfaces](#)

Description: Enables or disables enumeration of USB Interfaces.

- [GenApi::IBoolean](#) & [EnumerateGen2Cameras](#)

Description: Enables or disables the enumeration of USB3 and GigE based Generation 2 cameras.

Protected Member Functions

- [TransportLayerSystem](#) ()

Friends

- class [System](#)
- class [ISystem](#)
- class [SystemPtrInternal](#)

15.184.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

15.184.2 Constructor & Destructor Documentation

15.184.2.1 TransportLayerSystem() [1/2]

```
TransportLayerSystem (
    GenApi::INodeMap * nodeMapTLDevice )
```

15.184.2.2 ~TransportLayerSystem()

```
~TransportLayerSystem ( )
```

15.184.2.3 TransportLayerSystem() [2/2]

```
TransportLayerSystem ( ) [protected]
```

15.184.3 Friends And Related Function Documentation**15.184.3.1 ISystem**

```
friend class ISystem [friend]
```

15.184.3.2 System

```
friend class System [friend]
```

15.184.3.3 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

15.184.4 Member Data Documentation

15.184.4.1 EnumerateGen2Cameras

`GenApi::IBoolean& EnumerateGen2Cameras`

Description: Enables or disables the enumeration of USB3 and GigE based Generation 2 cameras.

This includes the CM3, FL3, GS3, and BFLY families. Visibility: Expert

15.184.4.2 EnumerateGEVInterfaces

`GenApi::IBoolean& EnumerateGEVInterfaces`

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

15.184.4.3 EnumerateUSBInterfaces

`GenApi::IBoolean& EnumerateUSBInterfaces`

Description: Enables or disables enumeration of USB Interfaces.

Visibility: Expert

15.184.4.4 GenTL_SFNC_Version_Major

`GenApi::IInteger& GenTL_SFNC_Version_Major`

Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.184.4.5 GenTL_SFNC_Version_Minor

`GenApi::IInteger& GenTL_SFNC_Version_Minor`

Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.184.4.6 GenTL_SFNC_Version_SubMinor

`GenApi::IInteger& GenTL_SFNC_Version_SubMinor`

Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

15.184.4.7 GenTLVersionMajor

`GenApi::Integer& GenTLVersionMajor`

Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

15.184.4.8 GenTLVersionMinor

`GenApi::Integer& GenTLVersionMinor`

Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

15.184.4.9 GevInterfaceDefaultGateway

`GenApi::Integer& GevInterfaceDefaultGateway`

Description: Gateway of the selected interface.

Visibility: Expert

15.184.4.10 GevInterfaceDefaultIPAddress

`GenApi::Integer& GevInterfaceDefaultIPAddress`

Description: IP address of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.184.4.11 GevInterfaceDefaultSubnetMask

`GenApi::Integer& GevInterfaceDefaultSubnetMask`

Description: Subnet mask of the first subnet of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.184.4.12 GevInterfaceMACAddress

`GenApi::Integer& GevInterfaceMACAddress`

Description: 48-bit MAC address of the selected interface.

Note that for a GenTL Producer implementation supporting GigE Vision this feature is mandatory. Visibility: Expert

15.184.4.13 **GevVersionMajor**

`GenApi::Integer& GevVersionMajor`

Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

15.184.4.14 **GevVersionMinor**

`GenApi::Integer& GevVersionMinor`

Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

15.184.4.15 **InterfaceDisplayName**

`GenApi::String& InterfaceDisplayName`

Description: A user-friendly name of the selected [Interface](#).

Visibility: Beginner

15.184.4.16 **InterfaceID**

`GenApi::String& InterfaceID`

Description: GenTL Producer wide unique identifier of the selected interface.

Visibility: Beginner

15.184.4.17 **InterfaceSelector**

`GenApi::Integer& InterfaceSelector`

Description: Selector for the different GenTL Producer interfaces.

This interface list only changes on execution of "InterfaceUpdateList". The selector is 0-based in order to match the index of the C interface. Visibility: Beginner

15.184.4.18 **InterfaceUpdateList**

`GenApi::Command& InterfaceUpdateList`

Description: Updates the internal list of the interfaces.

This feature is readable even if the execution cannot be performed immediately. The command then returns and the status can be polled. This function interacts with the `TLUpdateInterfaceList` function of the GenTL producer. It is up to the GenTL consumer to handle access in case both methods are used. Visibility: Beginner

15.184.4.19 TLDisplayName

`GenApi::IString& TLDisplayName`

Description: User readable name of the GenTL Producer.

Visibility: Expert

15.184.4.20 TLFileName

`GenApi::IString& TLFileName`

Description: Filename including extension of the GenTL Producer.

Visibility: Expert

15.184.4.21 TLID

`GenApi::IString& TLID`

Description: Unique identifier of the GenTL Producer like a GUID.

Visibility: Expert

15.184.4.22 TLModelName

`GenApi::IString& TLModelName`

Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.

Visibility: Beginner

15.184.4.23 TLPath

`GenApi::IString& TLPath`

Description: Full path to the GenTL Producer including filename and extension.

Visibility: Expert

15.184.4.24 TLType

`GenApi::IEnumerationT<TLTypeEnum>& TLType`

Description: Transport layer type of the GenTL Producer implementation.

Visibility: Expert

15.184.4.25 TLVendorName

`GenApi::IString& TLVendorName`

Description: Name of the GenTL Producer vendor.

Visibility: Beginner

15.184.4.26 TLVersion

`GenApi::IString& TLVersion`

Description: Vendor specific version string.

Visibility: Beginner

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

- include/[TransportLayerSystem.h](#)

15.185 U3V_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

15.185.1 Detailed Description

header of a GVCP request packet

15.185.2 Member Data Documentation

15.185.2.1 ChunkID

uint32_t ChunkID

15.185.2.2 ChunkLength

uint32_t ChunkLength

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

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

15.186 U3V_COMMAND_HEADER Struct Reference

U3V/GenCP command header.

Public Attributes

- uint32_t [Prefix](#)
- uint16_t [Flags](#)
- uint16_t [CommandId](#)
- uint16_t [Length](#)
- uint16_t [ReqId](#)

15.186.1 Detailed Description

U3V/GenCP command header.

15.186.2 Member Data Documentation

15.186.2.1 CommandId

uint16_t CommandId

15.186.2.2 Flags

uint16_t Flags

15.186.2.3 Length

uint16_t Length

15.186.2.4 Prefix

`uint32_t Prefix`

15.186.2.5 ReqId

`uint16_t ReqId`

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

- `include/SpinGenApi/EventAdapterU3V.h`

15.187 U3V_EVENT_DATA Struct Reference

U3V/GenCP EVENT_CMD specific command data.

Public Attributes

- `uint16_t` [Reserved](#)
- `uint16_t` [EventId](#)
- `uint64_t` [Timestamp](#)

15.187.1 Detailed Description

U3V/GenCP EVENT_CMD specific command data.

15.187.2 Member Data Documentation

15.187.2.1 EventId

`uint16_t EventId`

15.187.2.2 Reserved

`uint16_t Reserved`

15.187.2.3 Timestamp

`uint64_t` Timestamp

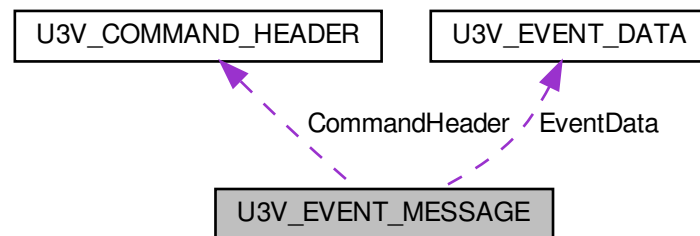
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

15.188 U3V_EVENT_MESSAGE Struct Reference

Entire event data message (without the variable-sized data field)

Collaboration diagram for U3V_EVENT_MESSAGE:



Public Attributes

- `U3V_COMMAND_HEADER` CommandHeader
- `U3V_EVENT_DATA` EventData

15.188.1 Detailed Description

Entire event data message (without the variable-sized data field)

15.188.2 Member Data Documentation

15.188.2.1 CommandHeader

`U3V_COMMAND_HEADER` CommandHeader

15.188.2.2 EventData

[U3V_EVENT_DATA](#) `EventData`

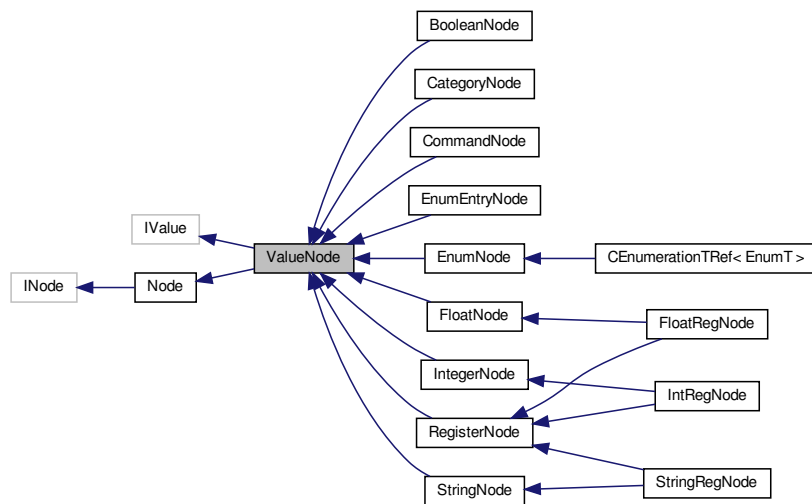
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

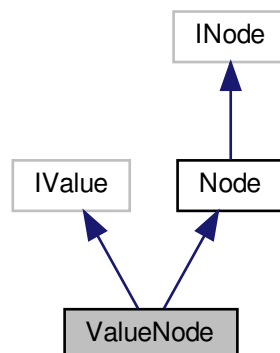
15.189 ValueNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for ValueNode:



Collaboration diagram for ValueNode:



Public Member Functions

- [ValueNode](#) ()
Constructor.
- [ValueNode](#) (std::shared_ptr< Node::NodeImpl > pValue)
constructor with [GenICam](#) IValue
- [~ValueNode](#) ()
Destructor.
- virtual [INode](#) * [GetNode](#) ()
- virtual [GenICam::gcstring](#) [ToString](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get content of the node as string.
- virtual void [FromString](#) (const [GenICam::gcstring](#) &ValueStr, bool [Verify](#)=true)
Set content of the node as string.
- virtual bool [IsValueCacheValid](#) () const
Checks if the value comes from cache or is requested from another node.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Value

Additional Inherited Members

15.189.1 Detailed Description

[Interface](#) for value properties.

15.189.2 Constructor & Destructor Documentation

15.189.2.1 ValueNode() [1/2]

[ValueNode](#) ()

Constructor.

15.189.2.2 ValueNode() [2/2]

[ValueNode](#) (
std::shared_ptr< Node::NodeImpl > pValue)

constructor with [GenICam](#) IValue

15.189.2.3 ~ValueNode()

```
~ValueNode ( )
```

Destructor.

15.189.3 Member Function Documentation**15.189.3.1 FromString()**

```
virtual void FromString (
    const GenICam::gcstring & ValueStr,
    bool Verify = true ) [virtual]
```

Set content of the node as string.

Parameters

<i>ValueStr</i>	The value to set
<i>Verify</i>	Enables AccessMode and Range verification (default = true)

15.189.3.2 GetNode()

```
virtual INode* GetNode ( ) [virtual]
```

15.189.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid ( ) const [virtual]
```

Checks if the value comes from cache or is requested from another node.

15.189.3.4 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

Reimplemented in [FloatNode](#), [IntegerNode](#), [EnumNode](#), [CEnumerationTRef< EnumT >](#), [StringNode](#), [RegisterNode](#), [BooleanNode](#), [CommandNode](#), [EnumEntryNode](#), [CategoryNode](#), [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

15.189.3.5 ToString()

```
virtual GenICam::gcstring ToString (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get content of the node as string.

Parameters

<i>Verify</i>	Enables Range verification (default = false). The AccessMode is always checked
<i>IgnoreCache</i>	If true the value is read ignoring any caches (default = false)

Returns

The value read

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

- include/SpinGenApi/[ValueNode.h](#)

15.190 Version_t Struct Reference

Version.

Public Attributes

- uint16_t [Major](#)
- uint16_t [Minor](#)
a is incompatible with b if a != b
- uint16_t [SubMinor](#)
a is incompatible b a > b

15.190.1 Detailed Description

Version.

15.190.2 Member Data Documentation

15.190.2.1 Major

uint16_t Major

15.190.2.2 Minor

uint16_t Minor

a is incompatible with b if $a \neq b$

15.190.2.3 SubMinor

uint16_t SubMinor

a is incompatible b $a > b$

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

- include/SpinGenApi/[GCTypes.h](#)

Chapter 16

File Documentation

16.1 [doc/spindocs/C++/GettingStarted.dox](#) File Reference

16.2 [doc/spindocs/C++/ProgrammerGuide.dox](#) File Reference

16.3 [doc/spindocs/shared/Benefits.dox](#) File Reference

16.4 [doc/spindocs/shared/FlyCapture2Comparison.dox](#) File Reference

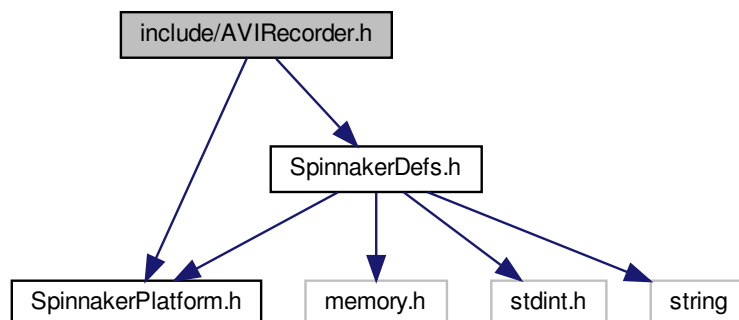
16.5 [doc/spindocs/shared/GenICamGenTL.dox](#) File Reference

16.6 [doc/spindocs/shared/Licensing.dox](#) File Reference

16.7 [doc/spindocs/shared/Maintenance.dox](#) File Reference

16.8 [include/AVIRecorder.h](#) File Reference

Include dependency graph for AVIRecorder.h:



Namespaces

- [Spinnaker](#)

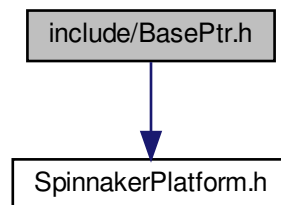
Functions

- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API AVIRecorder

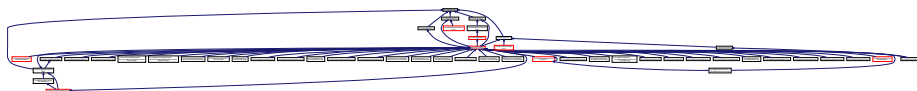
Provides the functionality for the user to record images to an AVI file.

16.9 include/BasePtr.h File Reference

Include dependency graph for BasePtr.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [BasePtr< T, B >](#)

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Namespaces

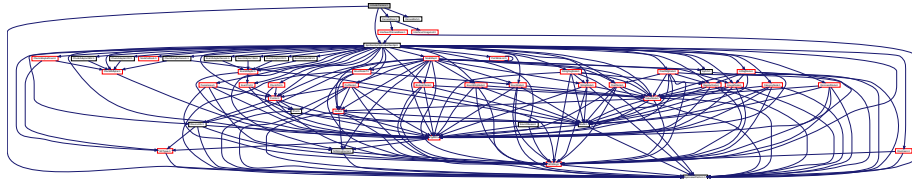
- [Spinnaker](#)

Functions

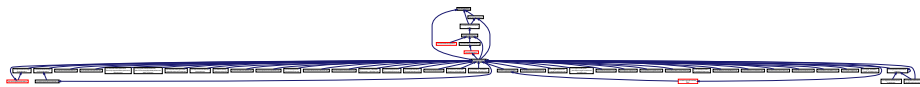
- template<class T, class B>
bool [operator==](#) (const std::nullptr_t, const BasePtr< T, B > &rhs)
Pointer equal.

16.10 include/Camera.h File Reference

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



Classes

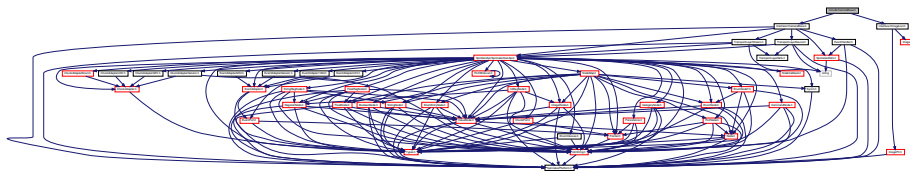
- class [Camera](#)
The camera object class.

Namespaces

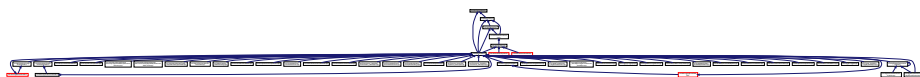
- [Spinnaker](#)

16.11 include/CameraBase.h File Reference

Include dependency graph for CameraBase.h:



This graph shows which files directly or indirectly include this file:



Classes

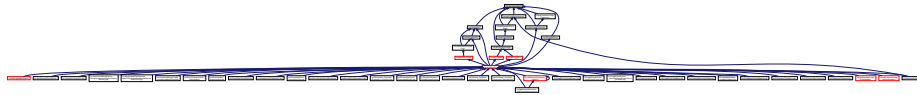
- class [CameraBase](#)
The base class for the camera object.

Namespaces

- [Spinnaker](#)

16.12 include/CameraDefs.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)

Enumerations

- enum [LUTSelectorEnums](#) {
[LUTSelector_LUT1](#),
[NUM_LUTSELECTOR](#) }

The enum definitions for camera nodes from the Standard Feature Naming Convention (SFNC) .xml files.

- enum [ExposureModeEnums](#) {
[ExposureMode_Timed](#),
[ExposureMode_TriggerWidth](#),
[NUM_EXPOSUREMODE](#) }
- enum [AcquisitionModeEnums](#) {
[AcquisitionMode_Continuous](#),
[AcquisitionMode_SingleFrame](#),
[AcquisitionMode_MultiFrame](#),
[NUM_ACQUISITIONMODE](#) }
- enum [TriggerSourceEnums](#) {
[TriggerSource_Software](#),
[TriggerSource_Line0](#),
[TriggerSource_Line1](#),
[TriggerSource_Line2](#),
[TriggerSource_Line3](#),
[TriggerSource_UserOutput0](#),
[TriggerSource_UserOutput1](#),
[TriggerSource_UserOutput2](#),
[TriggerSource_UserOutput3](#),
[TriggerSource_Counter0Start](#),
[TriggerSource_Counter1Start](#),
[TriggerSource_Counter0End](#),
[TriggerSource_Counter1End](#),
[TriggerSource_LogicBlock0](#),
[TriggerSource_LogicBlock1](#),
[TriggerSource_Action0](#),
[NUM_TRIGGERSOURCE](#) }

- enum [TriggerActivationEnums](#) {
 [TriggerActivation_LevelLow](#),
 [TriggerActivation_LevelHigh](#),
 [TriggerActivation_FallingEdge](#),
 [TriggerActivation_RisingEdge](#),
 [TriggerActivation_AnyEdge](#),
 [NUM_TRIGGERACTIVATION](#) }
- enum [SensorShutterModeEnums](#) {
 [SensorShutterMode_Global](#),
 [SensorShutterMode_Rolling](#),
 [SensorShutterMode_GlobalReset](#),
 [NUM_SENSORSHUTTERMODE](#) }
- enum [TriggerModeEnums](#) {
 [TriggerMode_Off](#),
 [TriggerMode_On](#),
 [NUM_TRIGGERMODE](#) }
- enum [TriggerOverlapEnums](#) {
 [TriggerOverlap_Off](#),
 [TriggerOverlap_ReadOut](#),
 [TriggerOverlap_PreviousFrame](#),
 [NUM_TRIGGEROVERLAP](#) }
- enum [TriggerSelectorEnums](#) {
 [TriggerSelector_AcquisitionStart](#),
 [TriggerSelector_FrameStart](#),
 [TriggerSelector_FrameBurstStart](#),
 [NUM_TRIGGERSELECTOR](#) }
- enum [ExposureAutoEnums](#) {
 [ExposureAuto_Off](#),
 [ExposureAuto_Once](#),
 [ExposureAuto_Continuous](#),
 [NUM_EXPOSUREAUTO](#) }
- enum [EventSelectorEnums](#) {
 [EventSelector_Error](#),
 [EventSelector_ExposureEnd](#),
 [EventSelector_SerialPortReceive](#),
 [NUM_EVENTSELECTOR](#) }
- enum [EventNotificationEnums](#) {
 [EventNotification_On](#),
 [EventNotification_Off](#),
 [NUM_EVENTNOTIFICATION](#) }
- enum [LogicBlockSelectorEnums](#) {
 [LogicBlockSelector_LogicBlock0](#),
 [LogicBlockSelector_LogicBlock1](#),
 [NUM_LOGICBLOCKSELECTOR](#) }
- enum [LogicBlockLUTInputActivationEnums](#) {
 [LogicBlockLUTInputActivation_LevelLow](#),
 [LogicBlockLUTInputActivation_LevelHigh](#),
 [LogicBlockLUTInputActivation_FallingEdge](#),
 [LogicBlockLUTInputActivation_RisingEdge](#),
 [LogicBlockLUTInputActivation_AnyEdge](#),
 [NUM_LOGICBLOCKLUTINPUTACTIVATION](#) }
- enum [LogicBlockLUTInputSelectorEnums](#) {
 [LogicBlockLUTInputSelector_Input0](#),
 [LogicBlockLUTInputSelector_Input1](#),
 [LogicBlockLUTInputSelector_Input2](#),
 [LogicBlockLUTInputSelector_Input3](#),
 [NUM_LOGICBLOCKLUTINPUTSELECTOR](#) }

- enum LogicBlockLUTInputSourceEnums {
 - LogicBlockLUTInputSource_Zero,
 - LogicBlockLUTInputSource_Line0,
 - LogicBlockLUTInputSource_Line1,
 - LogicBlockLUTInputSource_Line2,
 - LogicBlockLUTInputSource_Line3,
 - LogicBlockLUTInputSource_UserOutput0,
 - LogicBlockLUTInputSource_UserOutput1,
 - LogicBlockLUTInputSource_UserOutput2,
 - LogicBlockLUTInputSource_UserOutput3,
 - LogicBlockLUTInputSource_Counter0Start,
 - LogicBlockLUTInputSource_Counter1Start,
 - LogicBlockLUTInputSource_Counter0End,
 - LogicBlockLUTInputSource_Counter1End,
 - LogicBlockLUTInputSource_LogicBlock0,
 - LogicBlockLUTInputSource_LogicBlock1,
 - LogicBlockLUTInputSource_ExposureStart,
 - LogicBlockLUTInputSource_ExposureEnd,
 - LogicBlockLUTInputSource_FrameTriggerWait,
 - LogicBlockLUTInputSource_AcquisitionActive,
 - NUM_LOGICBLOCKLUTINPUTSOURCE }
- enum LogicBlockLUTSelectorEnums {
 - LogicBlockLUTSelector_Value,
 - LogicBlockLUTSelector_Enable,
 - NUM_LOGICBLOCKLUTSELECTOR }
- enum ColorTransformationSelectorEnums {
 - ColorTransformationSelector_RGBtoRGB,
 - ColorTransformationSelector_RGBtoYUV,
 - NUM_COLORTRANSFORMATIONSELECTOR }
- enum RgbTransformLightSourceEnums {
 - RgbTransformLightSource_General,
 - RgbTransformLightSource_Tungsten2800K,
 - RgbTransformLightSource_WarmFluorescent3000K,
 - RgbTransformLightSource_CoolFluorescent4000K,
 - RgbTransformLightSource_Daylight5000K,
 - RgbTransformLightSource_Cloudy6500K,
 - RgbTransformLightSource_Shade8000K,
 - RgbTransformLightSource_Custom,
 - NUM_RGBTRANSFORMLIGHTSOURCE }
- enum ColorTransformationValueSelectorEnums {
 - ColorTransformationValueSelector_Gain00,
 - ColorTransformationValueSelector_Gain01,
 - ColorTransformationValueSelector_Gain02,
 - ColorTransformationValueSelector_Gain10,
 - ColorTransformationValueSelector_Gain11,
 - ColorTransformationValueSelector_Gain12,
 - ColorTransformationValueSelector_Gain20,
 - ColorTransformationValueSelector_Gain21,
 - ColorTransformationValueSelector_Gain22,
 - ColorTransformationValueSelector_Offset0,
 - ColorTransformationValueSelector_Offset1,
 - ColorTransformationValueSelector_Offset2,
 - NUM_COLORTRANSFORMATIONVALUESELECTOR }
- enum DeviceRegistersEndiannessEnums {
 - DeviceRegistersEndianness_Little,
 - DeviceRegistersEndianness_Big,
 - NUM_DEVICEREGISTERSENDIANNESSE }
- enum DeviceScanTypeEnums {

- DeviceScanType_Areascan,
NUM_DEVICESCANTYPE }
- enum DeviceCharacterSetEnums {
DeviceCharacterSet_UTF8,
DeviceCharacterSet_ASCII,
NUM_DEVICECHARACTERSET }
- enum DeviceTLTypeEnums {
DeviceTLType_GigEVision,
DeviceTLType_CameraLink,
DeviceTLType_CameraLinkHS,
DeviceTLType_CoaXPress,
DeviceTLType_USB3Vision,
DeviceTLType_Custom,
NUM_DEVICETLTYPE }
- enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }
- enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTOR }
- enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICEINDICATORMODE }
- enum AutoExposureControlPriorityEnums {
AutoExposureControlPriority_Gain,
AutoExposureControlPriority_ExposureTime,
NUM_AUTOEXPOSURECONTROLPRIORITY }
- enum AutoExposureMeteringModeEnums {
AutoExposureMeteringMode_Average,
AutoExposureMeteringMode_Spot,
AutoExposureMeteringMode_Partial,
AutoExposureMeteringMode_CenterWeighted,
AutoExposureMeteringMode_HistogramPeak,
NUM_AUTOEXPOSUREMETERINGMODE }
- enum BalanceWhiteAutoProfileEnums {
BalanceWhiteAutoProfile_Indoor,
BalanceWhiteAutoProfile_Outdoor,
NUM_BALANCEWHITEAUTOPROFILE }
- enum AutoAlgorithmSelectorEnums {
AutoAlgorithmSelector_Awb,
AutoAlgorithmSelector_Ae,
NUM_AUTOALGORITHMSELECTOR }
- enum AutoExposureTargetGreyValueAutoEnums {
AutoExposureTargetGreyValueAuto_Off,
AutoExposureTargetGreyValueAuto_Continuous,
NUM_AUTOEXPOSURETARGETGREYVALUEAUTO }
- enum AutoExposureLightingModeEnums {
AutoExposureLightingMode_AutoDetect,
AutoExposureLightingMode_Backlight,
AutoExposureLightingMode_Frontlight,
AutoExposureLightingMode_Normal,
NUM_AUTOEXPOSURELIGHTINGMODE }
- enum GevIEEE1588StatusEnums {
GevIEEE1588Status_Initializing,
GevIEEE1588Status_Faulty,
GevIEEE1588Status_Disabled,

```

    GevIEEE1588Status_Listening,
    GevIEEE1588Status_PreMaster,
    GevIEEE1588Status_Master,
    GevIEEE1588Status_Passive,
    GevIEEE1588Status_Uncalibrated,
    GevIEEE1588Status_Slave,
    NUM_GEVIEEE1588STATUS }
• enum GevIEEE1588ModeEnums {
    GevIEEE1588Mode_Auto,
    GevIEEE1588Mode_SlaveOnly,
    NUM_GEVIEEE1588MODE }
• enum GevIEEE1588ClockAccuracyEnums {
    GevIEEE1588ClockAccuracy_Unknown,
    NUM_GEVIEEE1588CLOCKACCURACY }
• enum GevCCPEnums {
    GevCCP_OpenAccess,
    GevCCP_ExclusiveAccess,
    GevCCP_ControlAccess,
    NUM_GEVCCP }
• enum GevSupportedOptionSelectorEnums {
    GevSupportedOptionSelector_UserDefinedName,
    GevSupportedOptionSelector_SerialNumber,
    GevSupportedOptionSelector_HeartbeatDisable,
    GevSupportedOptionSelector_LinkSpeed,
    GevSupportedOptionSelector_CCPApplicationSocket,
    GevSupportedOptionSelector_ManifestTable,
    GevSupportedOptionSelector_TestData,
    GevSupportedOptionSelector_DiscoveryAckDelay,
    GevSupportedOptionSelector_DiscoveryAckDelayWritable,
    GevSupportedOptionSelector_ExtendedStatusCodes,
    GevSupportedOptionSelector_Action,
    GevSupportedOptionSelector_PendingAck,
    GevSupportedOptionSelector_EventData,
    GevSupportedOptionSelector_Event,
    GevSupportedOptionSelector_PacketResend,
    GevSupportedOptionSelector_WriteMem,
    GevSupportedOptionSelector_CommandsConcatenation,
    GevSupportedOptionSelector_IPConfigurationLLA,
    GevSupportedOptionSelector_IPConfigurationDHCP,
    GevSupportedOptionSelector_IPConfigurationPersistentIP,
    GevSupportedOptionSelector_StreamChannelSourceSocket,
    GevSupportedOptionSelector_MessageChannelSourceSocket,
    NUM_GEVSUPPORTEDOPTIONSELECTOR }
• enum BlackLevelSelectorEnums {
    BlackLevelSelector_All,
    BlackLevelSelector_Analog,
    BlackLevelSelector_Digital,
    NUM_BLACKLEVELSELECTOR }
• enum BalanceWhiteAutoEnums {
    BalanceWhiteAuto_Off,
    BalanceWhiteAuto_Once,
    BalanceWhiteAuto_Continuous,
    NUM_BALANCEWHITEAUTO }
• enum GainAutoEnums {
    GainAuto_Off,
    GainAuto_Once,
    GainAuto_Continuous,
    NUM_GAINAUTO }

```

- enum [BalanceRatioSelectorEnums](#) {
 [BalanceRatioSelector_Red](#),
 [BalanceRatioSelector_Blue](#),
 [NUM_BALANCERATIOSELECTOR](#) }
- enum [GainSelectorEnums](#) {
 [GainSelector_All](#),
 [NUM_GAINSELECTOR](#) }
- enum [DefectCorrectionModeEnums](#) {
 [DefectCorrectionMode_Average](#),
 [DefectCorrectionMode_Highlight](#),
 [DefectCorrectionMode_Zero](#),
 [NUM_DEFECTCORRECTIONMODE](#) }
- enum [UserSetSelectorEnums](#) {
 [UserSetSelector_Default](#),
 [UserSetSelector_UserSet0](#),
 [UserSetSelector_UserSet1](#),
 [NUM_USERSETSELECTOR](#) }
- enum [UserSetDefaultEnums](#) {
 [UserSetDefault_Default](#),
 [UserSetDefault_UserSet0](#),
 [UserSetDefault_UserSet1](#),
 [NUM_USERSETDEFAULT](#) }
- enum [SerialPortBaudRateEnums](#) {
 [SerialPortBaudRate_Baud300](#),
 [SerialPortBaudRate_Baud600](#),
 [SerialPortBaudRate_Baud1200](#),
 [SerialPortBaudRate_Baud2400](#),
 [SerialPortBaudRate_Baud4800](#),
 [SerialPortBaudRate_Baud9600](#),
 [SerialPortBaudRate_Baud14400](#),
 [SerialPortBaudRate_Baud19200](#),
 [SerialPortBaudRate_Baud38400](#),
 [SerialPortBaudRate_Baud57600](#),
 [SerialPortBaudRate_Baud115200](#),
 [SerialPortBaudRate_Baud230400](#),
 [SerialPortBaudRate_Baud460800](#),
 [SerialPortBaudRate_Baud921600](#),
 [NUM_SERIALPORTBAUDRATE](#) }
- enum [SerialPortParityEnums](#) {
 [SerialPortParity_None](#),
 [SerialPortParity_Odd](#),
 [SerialPortParity_Even](#),
 [SerialPortParity_Mark](#),
 [SerialPortParity_Space](#),
 [NUM_SERIALPORTPARITY](#) }
- enum [SerialPortSelectorEnums](#) {
 [SerialPortSelector_SerialPort0](#),
 [NUM_SERIALPORTSELECTOR](#) }
- enum [SerialPortStopBitsEnums](#) {
 [SerialPortStopBits_Bits1](#),
 [SerialPortStopBits_Bits1AndAHalf](#),
 [SerialPortStopBits_Bits2](#),
 [NUM_SERIALPORTSTOPBITS](#) }
- enum [SerialPortSourceEnums](#) {
 [SerialPortSource_Line0](#),
 [SerialPortSource_Line1](#),
 [SerialPortSource_Line2](#),
 [SerialPortSource_Line3](#),

```
SerialPortSource_Off,  
NUM_SERIALPORTSOURCE }  
• enum SequencerModeEnums {  
    SequencerMode_Off,  
    SequencerMode_On,  
    NUM_SEQUENCERMODE }  
• enum SequencerConfigurationValidEnums {  
    SequencerConfigurationValid_No,  
    SequencerConfigurationValid_Yes,  
    NUM_SEQUENCERCONFIGURATIONVALID }  
• enum SequencerSetValidEnums {  
    SequencerSetValid_No,  
    SequencerSetValid_Yes,  
    NUM_SEQUENCERSETVALID }  
• enum SequencerTriggerActivationEnums {  
    SequencerTriggerActivation_RisingEdge,  
    SequencerTriggerActivation_FallingEdge,  
    SequencerTriggerActivation_AnyEdge,  
    SequencerTriggerActivation_LevelHigh,  
    SequencerTriggerActivation_LevelLow,  
    NUM_SEQUENCERTRIGGERACTIVATION }  
• enum SequencerConfigurationModeEnums {  
    SequencerConfigurationMode_Off,  
    SequencerConfigurationMode_On,  
    NUM_SEQUENCERCONFIGURATIONMODE }  
• enum SequencerTriggerSourceEnums {  
    SequencerTriggerSource_Off,  
    SequencerTriggerSource_FrameStart,  
    NUM_SEQUENCERTRIGGERSOURCE }  
• enum TransferQueueModeEnums {  
    TransferQueueMode_FirstInFirstOut,  
    NUM_TRANSFERQUEUEMODE }  
• enum TransferOperationModeEnums {  
    TransferOperationMode_Continuous,  
    TransferOperationMode_MultiBlock,  
    NUM_TRANSFEROPERATIONMODE }  
• enum TransferControlModeEnums {  
    TransferControlMode_Basic,  
    TransferControlMode_Automatic,  
    TransferControlMode_UserControlled,  
    NUM_TRANSFERCONTROLMODE }  
• enum ChunkGainSelectorEnums {  
    ChunkGainSelector_All,  
    ChunkGainSelector_Red,  
    ChunkGainSelector_Green,  
    ChunkGainSelector_Blue,  
    NUM_CHUNKGAINSELECTOR }  
• enum ChunkSelectorEnums {  
    ChunkSelector_Image,  
    ChunkSelector_CRC,  
    ChunkSelector_FrameID,  
    ChunkSelector_OffsetX,  
    ChunkSelector_OffsetY,  
    ChunkSelector_Width,  
    ChunkSelector_Height,  
    ChunkSelector_ExposureTime,  
    ChunkSelector_Gain,  
    ChunkSelector_BlackLevel,
```



```

    ChunkSelector_PixelFormat,
    ChunkSelector_Timestamp,
    ChunkSelector_SequencerSetActive,
    ChunkSelector_SerialData,
    ChunkSelector_ExposureEndLineStatusAll,
    NUM_CHUNKSELECTOR }

• enum ChunkBlackLevelSelectorEnums {
    ChunkBlackLevelSelector_All,
    NUM_CHUNKBLACKLEVELSELECTOR }

• enum ChunkPixelFormatEnums {
    ChunkPixelFormat_Mono8,
    ChunkPixelFormat_Mono12Packed,
    ChunkPixelFormat_Mono16,
    ChunkPixelFormat_RGB8Packed,
    ChunkPixelFormat_YUV422Packed,
    ChunkPixelFormat_BayerGR8,
    ChunkPixelFormat_BayerRG8,
    ChunkPixelFormat_BayerGB8,
    ChunkPixelFormat_BayerBG8,
    ChunkPixelFormat_YCbCr601_422_8_CbYCrY,
    NUM_CHUNKPIXELFORMAT }

• enum FileOperationStatusEnums {
    FileOperationStatus_Success,
    FileOperationStatus_Failure,
    FileOperationStatus_Overflow,
    NUM_FILEOPERATIONSTATUS }

• enum FileOpenModeEnums {
    FileOpenMode_Read,
    FileOpenMode_Write,
    FileOpenMode_ReadWrite,
    NUM_FILEOPENMODE }

• enum FileOperationSelectorEnums {
    FileOperationSelector_Open,
    FileOperationSelector_Close,
    FileOperationSelector_Read,
    FileOperationSelector_Write,
    FileOperationSelector_Delete,
    NUM_FILEOPERATIONSELECTOR }

• enum FileSelectorEnums {
    FileSelector_UserSetDefault,
    FileSelector_UserSet0,
    FileSelector_UserSet1,
    FileSelector_UserFile1,
    FileSelector_SerialPort0,
    NUM_FILESELECTOR }

• enum BinningSelectorEnums {
    BinningSelector_All,
    BinningSelector_Sensor,
    BinningSelector_ISP,
    NUM_BINNINGSELECTOR }

• enum TestPatternGeneratorSelectorEnums {
    TestPatternGeneratorSelector_Sensor,
    TestPatternGeneratorSelector_PipelineStart,
    NUM_TESTPATTERNGENERATORSELECTOR }

• enum CompressionSaturationPriorityEnums {
    CompressionSaturationPriority_DropFrame,
    CompressionSaturationPriority_ReduceFrameRate,
    NUM_COMPRESSIONSATURATIONPRIORITY }

```

- enum [TestPatternEnums](#) {
 [TestPattern_Off](#),
 [TestPattern_Increment](#),
 [TestPattern_SensorTestPattern](#),
 [NUM_TESTPATTERN](#) }
- enum [PixelColorFilterEnums](#) {
 [PixelColorFilter_None](#),
 [PixelColorFilter_BayerRG](#),
 [PixelColorFilter_BayerGB](#),
 [PixelColorFilter_BayerGR](#),
 [PixelColorFilter_BayerBG](#),
 [NUM_PIXELCOLORFILTER](#) }
- enum [AdcBitDepthEnums](#) {
 [AdcBitDepth_Bit8](#),
 [AdcBitDepth_Bit10](#),
 [AdcBitDepth_Bit12](#),
 [AdcBitDepth_Bit14](#),
 [NUM_ADCBITDEPTH](#) }
- enum [DecimationHorizontalModeEnums](#) {
 [DecimationHorizontalMode_Discard](#),
 [NUM_DECIMATIONHORIZONTALMODE](#) }
- enum [BinningVerticalModeEnums](#) {
 [BinningVerticalMode_Sum](#),
 [BinningVerticalMode_Average](#),
 [NUM_BINNINGVERTICALMODE](#) }
- enum [PixelSizeEnums](#) {
 [PixelSize_Bpp1](#),
 [PixelSize_Bpp2](#),
 [PixelSize_Bpp4](#),
 [PixelSize_Bpp8](#),
 [PixelSize_Bpp10](#),
 [PixelSize_Bpp12](#),
 [PixelSize_Bpp14](#),
 [PixelSize_Bpp16](#),
 [PixelSize_Bpp20](#),
 [PixelSize_Bpp24](#),
 [PixelSize_Bpp30](#),
 [PixelSize_Bpp32](#),
 [PixelSize_Bpp36](#),
 [PixelSize_Bpp48](#),
 [PixelSize_Bpp64](#),
 [PixelSize_Bpp96](#),
 [NUM_PIXELSIZE](#) }
- enum [DecimationSelectorEnums](#) {
 [DecimationSelector_All](#),
 [DecimationSelector_Sensor](#),
 [NUM_DECIMATIONSELECTOR](#) }
- enum [ImageCompressionModeEnums](#) {
 [ImageCompressionMode_Off](#),
 [ImageCompressionMode_Lossless](#),
 [NUM_IMAGECOMPRESSIONMODE](#) }
- enum [BinningHorizontalModeEnums](#) {
 [BinningHorizontalMode_Sum](#),
 [BinningHorizontalMode_Average](#),
 [NUM_BINNINGHORIZONTALMODE](#) }
- enum [PixelFormatEnums](#) {
 [PixelFormat_Mono8](#),
 [PixelFormat_Mono16](#),

[PixelFormat_RGB8Packed,](#)
[PixelFormat_BayerGR8,](#)
[PixelFormat_BayerRG8,](#)
[PixelFormat_BayerGB8,](#)
[PixelFormat_BayerBG8,](#)
[PixelFormat_BayerGR16,](#)
[PixelFormat_BayerRG16,](#)
[PixelFormat_BayerGB16,](#)
[PixelFormat_BayerBG16,](#)
[PixelFormat_Mono12Packed,](#)
[PixelFormat_BayerGR12Packed,](#)
[PixelFormat_BayerRG12Packed,](#)
[PixelFormat_BayerGB12Packed,](#)
[PixelFormat_BayerBG12Packed,](#)
[PixelFormat_YUV411Packed,](#)
[PixelFormat_YUV422Packed,](#)
[PixelFormat_YUV444Packed,](#)
[PixelFormat_Mono12p,](#)
[PixelFormat_BayerGR12p,](#)
[PixelFormat_BayerRG12p,](#)
[PixelFormat_BayerGB12p,](#)
[PixelFormat_BayerBG12p,](#)
[PixelFormat_YCbCr8,](#)
[PixelFormat_YCbCr422_8,](#)
[PixelFormat_YCbCr411_8,](#)
[PixelFormat_BGR8,](#)
[PixelFormat_BGRa8,](#)
[PixelFormat_Mono10Packed,](#)
[PixelFormat_BayerGR10Packed,](#)
[PixelFormat_BayerRG10Packed,](#)
[PixelFormat_BayerGB10Packed,](#)
[PixelFormat_BayerBG10Packed,](#)
[PixelFormat_Mono10p,](#)
[PixelFormat_BayerGR10p,](#)
[PixelFormat_BayerRG10p,](#)
[PixelFormat_BayerGB10p,](#)
[PixelFormat_BayerBG10p,](#)
[PixelFormat_Mono1p,](#)
[PixelFormat_Mono2p,](#)
[PixelFormat_Mono4p,](#)
[PixelFormat_Mono8s,](#)
[PixelFormat_Mono10,](#)
[PixelFormat_Mono12,](#)
[PixelFormat_Mono14,](#)
[PixelFormat_Mono16s,](#)
[PixelFormat_Mono32f,](#)
[PixelFormat_BayerBG10,](#)
[PixelFormat_BayerBG12,](#)
[PixelFormat_BayerGB10,](#)
[PixelFormat_BayerGB12,](#)
[PixelFormat_BayerGR10,](#)
[PixelFormat_BayerGR12,](#)
[PixelFormat_BayerRG10,](#)
[PixelFormat_BayerRG12,](#)
[PixelFormat_RGBa8,](#)
[PixelFormat_RGBa10,](#)
[PixelFormat_RGBa10p,](#)
[PixelFormat_RGBa12,](#)

PixelFormat_RGBA12p,
PixelFormat_RGBA14,
PixelFormat_RGBA16,
PixelFormat_RGB8,
PixelFormat_RGB8_Planar,
PixelFormat_RGB10,
PixelFormat_RGB10_Planar,
PixelFormat_RGB10p,
PixelFormat_RGB10p32,
PixelFormat_RGB12,
PixelFormat_RGB12_Planar,
PixelFormat_RGB12p,
PixelFormat_RGB14,
PixelFormat_RGB16,
PixelFormat_RGB16s,
PixelFormat_RGB32f,
PixelFormat_RGB16_Planar,
PixelFormat_RGB565p,
PixelFormat_BGRA10,
PixelFormat_BGRA10p,
PixelFormat_BGRA12,
PixelFormat_BGRA12p,
PixelFormat_BGRA14,
PixelFormat_BGRA16,
PixelFormat_RGBA32f,
PixelFormat_BGR10,
PixelFormat_BGR10p,
PixelFormat_BGR12,
PixelFormat_BGR12p,
PixelFormat_BGR14,
PixelFormat_BGR16,
PixelFormat_BGR565p,
PixelFormat_R8,
PixelFormat_R10,
PixelFormat_R12,
PixelFormat_R16,
PixelFormat_G8,
PixelFormat_G10,
PixelFormat_G12,
PixelFormat_G16,
PixelFormat_B8,
PixelFormat_B10,
PixelFormat_B12,
PixelFormat_B16,
PixelFormat_Coord3D_ABC8,
PixelFormat_Coord3D_ABC8_Planar,
PixelFormat_Coord3D_ABC10p,
PixelFormat_Coord3D_ABC10p_Planar,
PixelFormat_Coord3D_ABC12p,
PixelFormat_Coord3D_ABC12p_Planar,
PixelFormat_Coord3D_ABC16,
PixelFormat_Coord3D_ABC16_Planar,
PixelFormat_Coord3D_ABC32f,
PixelFormat_Coord3D_ABC32f_Planar,
PixelFormat_Coord3D_AC8,
PixelFormat_Coord3D_AC8_Planar,
PixelFormat_Coord3D_AC10p,
PixelFormat_Coord3D_AC10p_Planar,

PixelFormat_Coord3D_AC12p,
PixelFormat_Coord3D_AC12p_Planar,
PixelFormat_Coord3D_AC16,
PixelFormat_Coord3D_AC16_Planar,
PixelFormat_Coord3D_AC32f,
PixelFormat_Coord3D_AC32f_Planar,
PixelFormat_Coord3D_A8,
PixelFormat_Coord3D_A10p,
PixelFormat_Coord3D_A12p,
PixelFormat_Coord3D_A16,
PixelFormat_Coord3D_A32f,
PixelFormat_Coord3D_B8,
PixelFormat_Coord3D_B10p,
PixelFormat_Coord3D_B12p,
PixelFormat_Coord3D_B16,
PixelFormat_Coord3D_B32f,
PixelFormat_Coord3D_C8,
PixelFormat_Coord3D_C10p,
PixelFormat_Coord3D_C12p,
PixelFormat_Coord3D_C16,
PixelFormat_Coord3D_C32f,
PixelFormat_Confidence1,
PixelFormat_Confidence1p,
PixelFormat_Confidence8,
PixelFormat_Confidence16,
PixelFormat_Confidence32f,
PixelFormat_BiColorBGRG8,
PixelFormat_BiColorBGRG10,
PixelFormat_BiColorBGRG10p,
PixelFormat_BiColorBGRG12,
PixelFormat_BiColorBGRG12p,
PixelFormat_BiColorRGBG8,
PixelFormat_BiColorRGBG10,
PixelFormat_BiColorRGBG10p,
PixelFormat_BiColorRGBG12,
PixelFormat_BiColorRGBG12p,
PixelFormat_SCF1WBWG8,
PixelFormat_SCF1WBWG10,
PixelFormat_SCF1WBWG10p,
PixelFormat_SCF1WBWG12,
PixelFormat_SCF1WBWG12p,
PixelFormat_SCF1WBWG14,
PixelFormat_SCF1WBWG16,
PixelFormat_SCF1WGWB8,
PixelFormat_SCF1WGWB10,
PixelFormat_SCF1WGWB10p,
PixelFormat_SCF1WGWB12,
PixelFormat_SCF1WGWB12p,
PixelFormat_SCF1WGWB14,
PixelFormat_SCF1WGWB16,
PixelFormat_SCF1WGWR8,
PixelFormat_SCF1WGWR10,
PixelFormat_SCF1WGWR10p,
PixelFormat_SCF1WGWR12,
PixelFormat_SCF1WGWR12p,
PixelFormat_SCF1WGWR14,
PixelFormat_SCF1WGWR16,
PixelFormat_SCF1WRWG8,

PixelFormat_SCF1WRWG10,
PixelFormat_SCF1WRWG10p,
PixelFormat_SCF1WRWG12,
PixelFormat_SCF1WRWG12p,
PixelFormat_SCF1WRWG14,
PixelFormat_SCF1WRWG16,
PixelFormat_YCbCr8_CbYCr,
PixelFormat_YCbCr10_CbYCr,
PixelFormat_YCbCr10p_CbYCr,
PixelFormat_YCbCr12_CbYCr,
PixelFormat_YCbCr12p_CbYCr,
PixelFormat_YCbCr411_8_CbYYCrYY,
PixelFormat_YCbCr422_8_CbYCrY,
PixelFormat_YCbCr422_10,
PixelFormat_YCbCr422_10_CbYCrY,
PixelFormat_YCbCr422_10p,
PixelFormat_YCbCr422_10p_CbYCrY,
PixelFormat_YCbCr422_12,
PixelFormat_YCbCr422_12_CbYCrY,
PixelFormat_YCbCr422_12p,
PixelFormat_YCbCr422_12p_CbYCrY,
PixelFormat_YCbCr601_8_CbYCr,
PixelFormat_YCbCr601_10_CbYCr,
PixelFormat_YCbCr601_10p_CbYCr,
PixelFormat_YCbCr601_12_CbYCr,
PixelFormat_YCbCr601_12p_CbYCr,
PixelFormat_YCbCr601_411_8_CbYYCrYY,
PixelFormat_YCbCr601_422_8,
PixelFormat_YCbCr601_422_8_CbYCrY,
PixelFormat_YCbCr601_422_10,
PixelFormat_YCbCr601_422_10_CbYCrY,
PixelFormat_YCbCr601_422_10p,
PixelFormat_YCbCr601_422_10p_CbYCrY,
PixelFormat_YCbCr601_422_12,
PixelFormat_YCbCr601_422_12_CbYCrY,
PixelFormat_YCbCr601_422_12p,
PixelFormat_YCbCr601_422_12p_CbYCrY,
PixelFormat_YCbCr709_8_CbYCr,
PixelFormat_YCbCr709_10_CbYCr,
PixelFormat_YCbCr709_10p_CbYCr,
PixelFormat_YCbCr709_12_CbYCr,
PixelFormat_YCbCr709_12p_CbYCr,
PixelFormat_YCbCr709_411_8_CbYYCrYY,
PixelFormat_YCbCr709_422_8,
PixelFormat_YCbCr709_422_8_CbYCrY,
PixelFormat_YCbCr709_422_10,
PixelFormat_YCbCr709_422_10_CbYCrY,
PixelFormat_YCbCr709_422_10p,
PixelFormat_YCbCr709_422_10p_CbYCrY,
PixelFormat_YCbCr709_422_12,
PixelFormat_YCbCr709_422_12_CbYCrY,
PixelFormat_YCbCr709_422_12p,
PixelFormat_YCbCr709_422_12p_CbYCrY,
PixelFormat_YUV8_UYV,
PixelFormat_YUV411_8_UYYVYY,
PixelFormat_YUV422_8,
PixelFormat_YUV422_8_UYVY,
PixelFormat_Polarized8,

```

PixelFormat_Polarized10p,
PixelFormat_Polarized12p,
PixelFormat_Polarized16,
PixelFormat_BayerRGPolarized8,
PixelFormat_BayerRGPolarized10p,
PixelFormat_BayerRGPolarized12p,
PixelFormat_BayerRGPolarized16,
PixelFormat_LLCMono8,
PixelFormat_LLCBayerRG8,
PixelFormat_JPEGMono8,
PixelFormat_JPEGColor8,
PixelFormat_Raw16,
PixelFormat_Raw8,
PixelFormat_R12_Jpeg,
PixelFormat_GR12_Jpeg,
PixelFormat_GB12_Jpeg,
PixelFormat_B12_Jpeg,
PixelFormat_GR12,
PixelFormat_GB12,
UNKNOWN_PIXELFORMAT,
NUM_PIXELFORMAT }

• enum DecimationVerticalModeEnums {
    DecimationVerticalMode_Discard,
    NUM_DECIMATIONVERTICALMODE }

• enum LineModeEnums {
    LineMode_Input,
    LineMode_Output,
    NUM_LINEMODE }

• enum LineSourceEnums {
    LineSource_Off,
    LineSource_Line0,
    LineSource_Line1,
    LineSource_Line2,
    LineSource_Line3,
    LineSource_UserOutput0,
    LineSource_UserOutput1,
    LineSource_UserOutput2,
    LineSource_UserOutput3,
    LineSource_Counter0Active,
    LineSource_Counter1Active,
    LineSource_LogicBlock0,
    LineSource_LogicBlock1,
    LineSource_ExposureActive,
    LineSource_FrameTriggerWait,
    LineSource_SerialPort0,
    LineSource_PPSSignal,
    LineSource_AllPixel,
    LineSource_AnyPixel,
    NUM_LINESOURCE }

• enum LineInputFilterSelectorEnums {
    LineInputFilterSelector_Degitch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }

• enum UserOutputSelectorEnums {
    UserOutputSelector_UserOutput0,
    UserOutputSelector_UserOutput1,
    UserOutputSelector_UserOutput2,
    UserOutputSelector_UserOutput3,

```

```

    NUM_USEROUTPUTSELECTOR }

    • enum LineFormatEnums {
        LineFormat_NoConnect,
        LineFormat_TriState,
        LineFormat_TTL,
        LineFormat_LVDS,
        LineFormat_RS422,
        LineFormat_OptoCoupled,
        LineFormat_OpenDrain,
        NUM_LINEFORMAT }

    • enum LineSelectorEnums {
        LineSelector_Line0,
        LineSelector_Line1,
        LineSelector_Line2,
        LineSelector_Line3,
        NUM_LINESELECTOR }

    • enum ExposureActiveModeEnums {
        ExposureActiveMode_Line1,
        ExposureActiveMode_AnyPixels,
        ExposureActiveMode_AllPixels,
        NUM_EXPOSUREACTIVEMODE }

    • enum CounterTriggerActivationEnums {
        CounterTriggerActivation_LevelLow,
        CounterTriggerActivation_LevelHigh,
        CounterTriggerActivation_FallingEdge,
        CounterTriggerActivation_RisingEdge,
        CounterTriggerActivation_AnyEdge,
        NUM_COUNTERTRIGGERACTIVATION }

    • enum CounterSelectorEnums {
        CounterSelector_Counter0,
        CounterSelector_Counter1,
        NUM_COUNTERSELECTOR }

    • enum CounterStatusEnums {
        CounterStatus_CounterIdle,
        CounterStatus_CounterTriggerWait,
        CounterStatus_CounterActive,
        CounterStatus_CounterCompleted,
        CounterStatus_CounterOverflow,
        NUM_COUNTERSTATUS }

    • enum CounterTriggerSourceEnums {
        CounterTriggerSource_Off,
        CounterTriggerSource_Line0,
        CounterTriggerSource_Line1,
        CounterTriggerSource_Line2,
        CounterTriggerSource_Line3,
        CounterTriggerSource_UserOutput0,
        CounterTriggerSource_UserOutput1,
        CounterTriggerSource_UserOutput2,
        CounterTriggerSource_UserOutput3,
        CounterTriggerSource_Counter0Start,
        CounterTriggerSource_Counter1Start,
        CounterTriggerSource_Counter0End,
        CounterTriggerSource_Counter1End,
        CounterTriggerSource_LogicBlock0,
        CounterTriggerSource_LogicBlock1,
        CounterTriggerSource_ExposureStart,
        CounterTriggerSource_ExposureEnd,

```



```

CounterTriggerSource_FrameTriggerWait,
NUM_COUNTERTRIGGERSOURCE }

• enum CounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,
CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }

• enum CounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,
CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }

• enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }

• enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }

• enum DeviceTypeEnums {

```

```

DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }

• enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }

• enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }

• enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }

• enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }

• enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }

• enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }

• enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }

• enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }

• enum SensorTapsEnums {
SensorTaps_One,
SensorTaps_Two,
SensorTaps_Three,
SensorTaps_Four,
SensorTaps_Eight,
SensorTaps_Ten,
NUM_SENSORTAPS }

• enum SensorDigitizationTapsEnums {
SensorDigitizationTaps_One,
SensorDigitizationTaps_Two,
SensorDigitizationTaps_Three,
SensorDigitizationTaps_Four,
SensorDigitizationTaps_Eight,

```

```
SensorDigitizationTaps_Ten,  
NUM_SENSORDIGITIZATIONTAPS }  
  
• enum RegionSelectorEnums {  
    RegionSelector_Region0,  
    RegionSelector_Region1,  
    RegionSelector_Region2,  
    RegionSelector_All,  
    NUM_REGIONSELECTOR }  
  
• enum RegionModeEnums {  
    RegionMode_Off,  
    RegionMode_On,  
    NUM_REGIONMODE }  
  
• enum RegionDestinationEnums {  
    RegionDestination_Stream0,  
    RegionDestination_Stream1,  
    RegionDestination_Stream2,  
    NUM_REGIONDESTINATION }  
  
• enum ImageComponentSelectorEnums {  
    ImageComponentSelector_Intensity,  
    ImageComponentSelector_Color,  
    ImageComponentSelector_Infrared,  
    ImageComponentSelector_Ultraviolet,  
    ImageComponentSelector_Range,  
    ImageComponentSelector_Disparity,  
    ImageComponentSelector_Confidence,  
    ImageComponentSelector_Scatter,  
    NUM_IMAGECOMPONENTSELECTOR }  
  
• enum PixelFormatInfoSelectorEnums {  
    PixelFormatInfoSelector_Mono1p,  
    PixelFormatInfoSelector_Mono2p,  
    PixelFormatInfoSelector_Mono4p,  
    PixelFormatInfoSelector_Mono8,  
    PixelFormatInfoSelector_Mono8s,  
    PixelFormatInfoSelector_Mono10,  
    PixelFormatInfoSelector_Mono10p,  
    PixelFormatInfoSelector_Mono12,  
    PixelFormatInfoSelector_Mono12p,  
    PixelFormatInfoSelector_Mono14,  
    PixelFormatInfoSelector_Mono16,  
    PixelFormatInfoSelector_Mono16s,  
    PixelFormatInfoSelector_Mono32f,  
    PixelFormatInfoSelector_BayerBG8,  
    PixelFormatInfoSelector_BayerBG10,  
    PixelFormatInfoSelector_BayerBG10p,  
    PixelFormatInfoSelector_BayerBG12,  
    PixelFormatInfoSelector_BayerBG12p,  
    PixelFormatInfoSelector_BayerBG16,  
    PixelFormatInfoSelector_BayerGB8,  
    PixelFormatInfoSelector_BayerGB10,  
    PixelFormatInfoSelector_BayerGB10p,  
    PixelFormatInfoSelector_BayerGB12,  
    PixelFormatInfoSelector_BayerGB12p,  
    PixelFormatInfoSelector_BayerGB16,  
    PixelFormatInfoSelector_BayerGR8,  
    PixelFormatInfoSelector_BayerGR10,  
    PixelFormatInfoSelector_BayerGR10p,  
    PixelFormatInfoSelector_BayerGR12,  
    PixelFormatInfoSelector_BayerGR12p,
```

PixelFormatInfoSelector_BayerGR16,
PixelFormatInfoSelector_BayerRG8,
PixelFormatInfoSelector_BayerRG10,
PixelFormatInfoSelector_BayerRG10p,
PixelFormatInfoSelector_BayerRG12,
PixelFormatInfoSelector_BayerRG12p,
PixelFormatInfoSelector_BayerRG16,
PixelFormatInfoSelector_RGBa8,
PixelFormatInfoSelector_RGBa10,
PixelFormatInfoSelector_RGBa10p,
PixelFormatInfoSelector_RGBa12,
PixelFormatInfoSelector_RGBa12p,
PixelFormatInfoSelector_RGBa14,
PixelFormatInfoSelector_RGBa16,
PixelFormatInfoSelector_RGB8,
PixelFormatInfoSelector_RGB8_Planar,
PixelFormatInfoSelector_RGB10,
PixelFormatInfoSelector_RGB10_Planar,
PixelFormatInfoSelector_RGB10p,
PixelFormatInfoSelector_RGB10p32,
PixelFormatInfoSelector_RGB12,
PixelFormatInfoSelector_RGB12_Planar,
PixelFormatInfoSelector_RGB12p,
PixelFormatInfoSelector_RGB14,
PixelFormatInfoSelector_RGB16,
PixelFormatInfoSelector_RGB16s,
PixelFormatInfoSelector_RGB32f,
PixelFormatInfoSelector_RGB16_Planar,
PixelFormatInfoSelector_RGB565p,
PixelFormatInfoSelector_BGRa8,
PixelFormatInfoSelector_BGRa10,
PixelFormatInfoSelector_BGRa10p,
PixelFormatInfoSelector_BGRa12,
PixelFormatInfoSelector_BGRa12p,
PixelFormatInfoSelector_BGRa14,
PixelFormatInfoSelector_BGRa16,
PixelFormatInfoSelector_RGBa32f,
PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,
PixelFormatInfoSelector_Coord3D_ABC8,

[PixelFormatInfoSelector_Coord3D_ABC8_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC10p](#),
[PixelFormatInfoSelector_Coord3D_ABC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC12p](#),
[PixelFormatInfoSelector_Coord3D_ABC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC16](#),
[PixelFormatInfoSelector_Coord3D_ABC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_ABC32f](#),
[PixelFormatInfoSelector_Coord3D_ABC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC8](#),
[PixelFormatInfoSelector_Coord3D_AC8_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC10p](#),
[PixelFormatInfoSelector_Coord3D_AC10p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC12p](#),
[PixelFormatInfoSelector_Coord3D_AC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC16](#),
[PixelFormatInfoSelector_Coord3D_AC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC32f](#),
[PixelFormatInfoSelector_Coord3D_AC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_A8](#),
[PixelFormatInfoSelector_Coord3D_A10p](#),
[PixelFormatInfoSelector_Coord3D_A12p](#),
[PixelFormatInfoSelector_Coord3D_A16](#),
[PixelFormatInfoSelector_Coord3D_A32f](#),
[PixelFormatInfoSelector_Coord3D_B8](#),
[PixelFormatInfoSelector_Coord3D_B10p](#),
[PixelFormatInfoSelector_Coord3D_B12p](#),
[PixelFormatInfoSelector_Coord3D_B16](#),
[PixelFormatInfoSelector_Coord3D_B32f](#),
[PixelFormatInfoSelector_Coord3D_C8](#),
[PixelFormatInfoSelector_Coord3D_C10p](#),
[PixelFormatInfoSelector_Coord3D_C12p](#),
[PixelFormatInfoSelector_Coord3D_C16](#),
[PixelFormatInfoSelector_Coord3D_C32f](#),
[PixelFormatInfoSelector_Confidence1](#),
[PixelFormatInfoSelector_Confidence1p](#),
[PixelFormatInfoSelector_Confidence8](#),
[PixelFormatInfoSelector_Confidence16](#),
[PixelFormatInfoSelector_Confidence32f](#),
[PixelFormatInfoSelector_BiColorBGRG8](#),
[PixelFormatInfoSelector_BiColorBGRG10](#),
[PixelFormatInfoSelector_BiColorBGRG10p](#),
[PixelFormatInfoSelector_BiColorBGRG12](#),
[PixelFormatInfoSelector_BiColorBGRG12p](#),
[PixelFormatInfoSelector_BiColorRGBG8](#),
[PixelFormatInfoSelector_BiColorRGBG10](#),
[PixelFormatInfoSelector_BiColorRGBG10p](#),
[PixelFormatInfoSelector_BiColorRGBG12](#),
[PixelFormatInfoSelector_BiColorRGBG12p](#),
[PixelFormatInfoSelector_SCF1WBWG8](#),
[PixelFormatInfoSelector_SCF1WBWG10](#),
[PixelFormatInfoSelector_SCF1WBWG10p](#),
[PixelFormatInfoSelector_SCF1WBWG12](#),
[PixelFormatInfoSelector_SCF1WBWG12p](#),
[PixelFormatInfoSelector_SCF1WBWG14](#),
[PixelFormatInfoSelector_SCF1WBWG16](#),
[PixelFormatInfoSelector_SCF1WGWB8](#),
[PixelFormatInfoSelector_SCF1WGWB10](#),

PixelFormatInfoSelector_SCF1WGWB10p,
PixelFormatInfoSelector_SCF1WGWB12,
PixelFormatInfoSelector_SCF1WGWB12p,
PixelFormatInfoSelector_SCF1WGWB14,
PixelFormatInfoSelector_SCF1WGWB16,
PixelFormatInfoSelector_SCF1WGWR8,
PixelFormatInfoSelector_SCF1WGWR10,
PixelFormatInfoSelector_SCF1WGWR10p,
PixelFormatInfoSelector_SCF1WGWR12,
PixelFormatInfoSelector_SCF1WGWR12p,
PixelFormatInfoSelector_SCF1WGWR14,
PixelFormatInfoSelector_SCF1WGWR16,
PixelFormatInfoSelector_SCF1WRWG8,
PixelFormatInfoSelector_SCF1WRWG10,
PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,

```

PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum DeinterlacingEnums {
    Deinterlacing_Off,
    Deinterlacing_LineDuplication,
    Deinterlacing_Weave,
    NUM_DEINTERLACING }

• enum ImageCompressionRateOptionEnums {
    ImageCompressionRateOption_FixBitrate,
    ImageCompressionRateOption_FixQuality,
    NUM_IMAGECOMPRESSIONRATEOPTION }

• enum ImageCompressionJPEGFormatOptionEnums {
    ImageCompressionJPEGFormatOption_Lossless,
    ImageCompressionJPEGFormatOption_BaselineStandard,
    ImageCompressionJPEGFormatOption_BaselineOptimized,
    ImageCompressionJPEGFormatOption_Progressive,
    NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

• enum AcquisitionStatusSelectorEnums {
    AcquisitionStatusSelector_AcquisitionTriggerWait,
    AcquisitionStatusSelector_AcquisitionActive,
    AcquisitionStatusSelector_AcquisitionTransfer,
    AcquisitionStatusSelector_FrameTriggerWait,
    AcquisitionStatusSelector_FrameActive,
    AcquisitionStatusSelector_ExposureActive,
    NUM_ACQUISITIONSTATUSSELECTOR }

• enum ExposureTimeModeEnums {
    ExposureTimeMode_Common,
    ExposureTimeMode_Individual,
    NUM_EXPOSURETIMEMODE }

• enum ExposureTimeSelectorEnums {
    ExposureTimeSelector_Common,
    ExposureTimeSelector_Red,

```

```

    ExposureTimeSelector_Green,
    ExposureTimeSelector_Blue,
    ExposureTimeSelector_Cyan,
    ExposureTimeSelector_Magenta,
    ExposureTimeSelector_Yellow,
    ExposureTimeSelector_Infrared,
    ExposureTimeSelector_Ultraviolet,
    ExposureTimeSelector_Stage1,
    ExposureTimeSelector_Stage2,
    NUM_EXPOSURETIMESELECTOR }

• enum GainAutoBalanceEnums {
    GainAutoBalance_Off,
    GainAutoBalance_Once,
    GainAutoBalance_Continuous,
    NUM_GAINAUTOBALANCE }

• enum BlackLevelAutoEnums {
    BlackLevelAuto_Off,
    BlackLevelAuto_Once,
    BlackLevelAuto_Continuous,
    NUM_BLACKLEVELAUTO }

• enum BlackLevelAutoBalanceEnums {
    BlackLevelAutoBalance_Off,
    BlackLevelAutoBalance_Once,
    BlackLevelAutoBalance_Continuous,
    NUM_BLACKLEVELAUTOBALANCE }

• enum WhiteClipSelectorEnums {
    WhiteClipSelector_All,
    WhiteClipSelector_Red,
    WhiteClipSelector_Green,
    WhiteClipSelector_Blue,
    WhiteClipSelector_Y,
    WhiteClipSelector_U,
    WhiteClipSelector_V,
    WhiteClipSelector_Tap1,
    WhiteClipSelector_Tap2,
    NUM_WHITECLIPSELECTOR }

• enum TimerSelectorEnums {
    TimerSelector_Timer0,
    TimerSelector_Timer1,
    TimerSelector_Timer2,
    NUM_TIMERSELECTOR }

• enum TimerStatusEnums {
    TimerStatus_TimerIdle,
    TimerStatus_TimerTriggerWait,
    TimerStatus_TimerActive,
    TimerStatus_TimerCompleted,
    NUM_TIMERSTATUS }

• enum TimerTriggerSourceEnums {
    TimerTriggerSource_Off,
    TimerTriggerSource_AcquisitionTrigger,
    TimerTriggerSource_AcquisitionStart,
    TimerTriggerSource_AcquisitionEnd,
    TimerTriggerSource_FrameTrigger,
    TimerTriggerSource_FrameStart,
    TimerTriggerSource_FrameEnd,
    TimerTriggerSource_FrameBurstStart,
    TimerTriggerSource_FrameBurstEnd,
    TimerTriggerSource_LineTrigger,

```



```

TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }

• enum TimerTriggerActivationEnums {
    TimerTriggerActivation_RisingEdge,
    TimerTriggerActivation_FallingEdge,
    TimerTriggerActivation_AnyEdge,
    TimerTriggerActivation_LevelHigh,
    TimerTriggerActivation_LevelLow,
    NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
    EncoderSelector_Encoder0,
    EncoderSelector_Encoder1,
    EncoderSelector_Encoder2,
    NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {
    EncoderSourceA_Off,
    EncoderSourceA_Line0,
    EncoderSourceA_Line1,
    EncoderSourceA_Line2,
    NUM_ENCODERSOURCEA }

• enum EncoderSourceBEnums {
    EncoderSourceB_Off,
    EncoderSourceB_Line0,
    EncoderSourceB_Line1,
    EncoderSourceB_Line2,

```

```

    NUM_ENCODERSOURCEB }
    • enum EncoderModeEnums {
        EncoderMode_FourPhase,
        EncoderMode_HighResolution,
        NUM_ENCODERMODE }
    • enum EncoderOutputModeEnums {
        EncoderOutputMode_Off,
        EncoderOutputMode_PositionUp,
        EncoderOutputMode_PositionDown,
        EncoderOutputMode_DirectionUp,
        EncoderOutputMode_DirectionDown,
        EncoderOutputMode_Motion,
        NUM_ENCODEROUTPUTMODE }
    • enum EncoderStatusEnums {
        EncoderStatus_EncoderUp,
        EncoderStatus_EncoderDown,
        EncoderStatus_EncoderIdle,
        EncoderStatus_EncoderStatic,
        NUM_ENCODERSTATUS }
    • enum EncoderResetSourceEnums {
        EncoderResetSource_Off,
        EncoderResetSource_AcquisitionTrigger,
        EncoderResetSource_AcquisitionStart,
        EncoderResetSource_AcquisitionEnd,
        EncoderResetSource_FrameTrigger,
        EncoderResetSource_FrameStart,
        EncoderResetSource_FrameEnd,
        EncoderResetSource_ExposureStart,
        EncoderResetSource_ExposureEnd,
        EncoderResetSource_Line0,
        EncoderResetSource_Line1,
        EncoderResetSource_Line2,
        EncoderResetSource_Counter0Start,
        EncoderResetSource_Counter1Start,
        EncoderResetSource_Counter2Start,
        EncoderResetSource_Counter0End,
        EncoderResetSource_Counter1End,
        EncoderResetSource_Counter2End,
        EncoderResetSource_Timer0Start,
        EncoderResetSource_Timer1Start,
        EncoderResetSource_Timer2Start,
        EncoderResetSource_Timer0End,
        EncoderResetSource_Timer1End,
        EncoderResetSource_Timer2End,
        EncoderResetSource_UserOutput0,
        EncoderResetSource_UserOutput1,
        EncoderResetSource_UserOutput2,
        EncoderResetSource_SoftwareSignal0,
        EncoderResetSource_SoftwareSignal1,
        EncoderResetSource_SoftwareSignal2,
        EncoderResetSource_Action0,
        EncoderResetSource_Action1,
        EncoderResetSource_Action2,
        EncoderResetSource_LinkTrigger0,
        EncoderResetSource_LinkTrigger1,
        EncoderResetSource_LinkTrigger2,
        NUM_ENCODERRESETSOURCE }
    • enum EncoderResetActivationEnums {

```

```

EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }

• enum SoftwareSignalSelectorEnums {
    SoftwareSignalSelector_SoftwareSignal0,
    SoftwareSignalSelector_SoftwareSignal1,
    SoftwareSignalSelector_SoftwareSignal2,
    NUM_SOFTWARESIGNALSELECTOR }

• enum ActionUnconditionalModeEnums {
    ActionUnconditionalMode_Off,
    ActionUnconditionalMode_On,
    NUM_ACTIONUNCONDITIONALMODE }

• enum SourceSelectorEnums {
    SourceSelector_Source0,
    SourceSelector_Source1,
    SourceSelector_Source2,
    SourceSelector_All,
    NUM_SOURCESELECTOR }

• enum TransferSelectorEnums {
    TransferSelector_Stream0,
    TransferSelector_Stream1,
    TransferSelector_Stream2,
    TransferSelector_All,
    NUM_TRANSFERSELECTOR }

• enum TransferTriggerSelectorEnums {
    TransferTriggerSelector_TransferStart,
    TransferTriggerSelector_TransferStop,
    TransferTriggerSelector_TransferAbort,
    TransferTriggerSelector_TransferPause,
    TransferTriggerSelector_TransferResume,
    TransferTriggerSelector_TransferActive,
    TransferTriggerSelector_TransferBurstStart,
    TransferTriggerSelector_TransferBurstStop,
    NUM_TRANSFERTRIGGERSELECTOR }

• enum TransferTriggerModeEnums {
    TransferTriggerMode_Off,
    TransferTriggerMode_On,
    NUM_TRANSFERTRIGGERMODE }

• enum TransferTriggerSourceEnums {
    TransferTriggerSource_Line0,
    TransferTriggerSource_Line1,
    TransferTriggerSource_Line2,
    TransferTriggerSource_Counter0Start,
    TransferTriggerSource_Counter1Start,
    TransferTriggerSource_Counter2Start,
    TransferTriggerSource_Counter0End,
    TransferTriggerSource_Counter1End,
    TransferTriggerSource_Counter2End,
    TransferTriggerSource_Timer0Start,
    TransferTriggerSource_Timer1Start,
    TransferTriggerSource_Timer2Start,
    TransferTriggerSource_Timer0End,
    TransferTriggerSource_Timer1End,
    TransferTriggerSource_Timer2End,
    TransferTriggerSource_SoftwareSignal0,

```

```

TransferTriggerSource_SoftwareSignal1,
TransferTriggerSource_SoftwareSignal2,
TransferTriggerSource_Action0,
TransferTriggerSource_Action1,
TransferTriggerSource_Action2,
NUM_TRANSFERTRIGGERSOURCE }

• enum TransferTriggerActivationEnums {
TransferTriggerActivation_RisingEdge,
TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }

• enum TransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }

• enum TransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,
TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }

• enum Scan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }

• enum Scan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }

• enum Scan3dOutputModeEnums {
Scan3dOutputMode_UncalibratedC,
Scan3dOutputMode_CalibratedABC_Grid,
Scan3dOutputMode_CalibratedABC_PointCloud,
Scan3dOutputMode_CalibratedAC,
Scan3dOutputMode_CalibratedAC_Linescan,
Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,
Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }

• enum Scan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }

• enum Scan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,
Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }

```

- enum Scan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,
Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }
- enum Scan3dCoordinateReferenceSelectorEnums {
Scan3dCoordinateReferenceSelector_RotationX,
Scan3dCoordinateReferenceSelector_RotationY,
Scan3dCoordinateReferenceSelector_RotationZ,
Scan3dCoordinateReferenceSelector_TranslationX,
Scan3dCoordinateReferenceSelector_TranslationY,
Scan3dCoordinateReferenceSelector_TranslationZ,
NUM_SCAN3DCOORDINATEREFERENCESELECTOR }
- enum ChunkImageComponentEnums {
ChunkImageComponent_Intensity,
ChunkImageComponent_Color,
ChunkImageComponent_Infrared,
ChunkImageComponent_Ultraviolet,
ChunkImageComponent_Range,
ChunkImageComponent_Disparity,
ChunkImageComponent_Confidence,
ChunkImageComponent_Scatter,
NUM_CHUNKIMAGECOMPONENT }
- enum ChunkCounterSelectorEnums {
ChunkCounterSelector_Counter0,
ChunkCounterSelector_Counter1,
ChunkCounterSelector_Counter2,
NUM_CHUNKCOUNTERSELECTOR }
- enum ChunkTimerSelectorEnums {
ChunkTimerSelector_Timer0,
ChunkTimerSelector_Timer1,
ChunkTimerSelector_Timer2,
NUM_CHUNKTIMERSELECTOR }
- enum ChunkEncoderSelectorEnums {
ChunkEncoderSelector_Encoder0,
ChunkEncoderSelector_Encoder1,
ChunkEncoderSelector_Encoder2,
NUM_CHUNKENCODERSELECTOR }
- enum ChunkEncoderStatusEnums {
ChunkEncoderStatus_EncoderUp,
ChunkEncoderStatus_EncoderDown,
ChunkEncoderStatus_EncoderIdle,
ChunkEncoderStatus_EncoderStatic,
NUM_CHUNKENCODERSTATUS }
- enum ChunkExposureTimeSelectorEnums {
ChunkExposureTimeSelector_Common,
ChunkExposureTimeSelector_Red,
ChunkExposureTimeSelector_Green,
ChunkExposureTimeSelector_Blue,
ChunkExposureTimeSelector_Cyan,
ChunkExposureTimeSelector_Magenta,
ChunkExposureTimeSelector_Yellow,
ChunkExposureTimeSelector_Infrared,
ChunkExposureTimeSelector_Ultraviolet,
ChunkExposureTimeSelector_Stage1,

```

    ChunkExposureTimeSelector_Stage2,
    NUM_CHUNKEXPOSURETIMESELECTOR }

• enum ChunkSourceIDEnums {
    ChunkSourceID_Source0,
    ChunkSourceID_Source1,
    ChunkSourceID_Source2,
    NUM_CHUNKSOURCEID }

• enum ChunkRegionIDEnums {
    ChunkRegionID_Region0,
    ChunkRegionID_Region1,
    ChunkRegionID_Region2,
    NUM_CHUNKREGIONID }

• enum ChunkTransferStreamIDEnums {
    ChunkTransferStreamID_Stream0,
    ChunkTransferStreamID_Stream1,
    ChunkTransferStreamID_Stream2,
    ChunkTransferStreamID_Stream3,
    NUM_CHUNKTRANSFERSTREAMID }

• enum ChunkScan3dDistanceUnitEnums {
    ChunkScan3dDistanceUnit_Millimeter,
    ChunkScan3dDistanceUnit_Inch,
    NUM_CHUNKSCAN3DDISTANCEUNIT }

• enum ChunkScan3dOutputModeEnums {
    ChunkScan3dOutputMode_UncalibratedC,
    ChunkScan3dOutputMode_CalibratedABC_Grid,
    ChunkScan3dOutputMode_CalibratedABC_PointCloud,
    ChunkScan3dOutputMode_CalibratedAC,
    ChunkScan3dOutputMode_CalibratedAC_Linescan,
    ChunkScan3dOutputMode_CalibratedC,
    ChunkScan3dOutputMode_CalibratedC_Linescan,
    ChunkScan3dOutputMode_RectifiedC,
    ChunkScan3dOutputMode_RectifiedC_Linescan,
    ChunkScan3dOutputMode_DisparityC,
    ChunkScan3dOutputMode_DisparityC_Linescan,
    NUM_CHUNKSCAN3DOUTPUTMODE }

• enum ChunkScan3dCoordinateSystemEnums {
    ChunkScan3dCoordinateSystem_Cartesian,
    ChunkScan3dCoordinateSystem_Spherical,
    ChunkScan3dCoordinateSystem_Cylindrical,
    NUM_CHUNKSCAN3DCOORDINATESYSTEM }

• enum ChunkScan3dCoordinateSystemReferenceEnums {
    ChunkScan3dCoordinateSystemReference_Anchor,
    ChunkScan3dCoordinateSystemReference_Transformed,
    NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }

• enum ChunkScan3dCoordinateSelectorEnums {
    ChunkScan3dCoordinateSelector_CoordinateA,
    ChunkScan3dCoordinateSelector_CoordinateB,
    ChunkScan3dCoordinateSelector_CoordinateC,
    NUM_CHUNKSCAN3DCOORDINATESELECTOR }

• enum ChunkScan3dCoordinateTransformSelectorEnums {
    ChunkScan3dCoordinateTransformSelector_RotationX,
    ChunkScan3dCoordinateTransformSelector_RotationY,
    ChunkScan3dCoordinateTransformSelector_RotationZ,
    ChunkScan3dCoordinateTransformSelector_TranslationX,
    ChunkScan3dCoordinateTransformSelector_TranslationY,
    ChunkScan3dCoordinateTransformSelector_TranslationZ,
    NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }

```

- `enum ChunkScan3dCoordinateReferenceSelectorEnums {`
`ChunkScan3dCoordinateReferenceSelector_RotationX,`
`ChunkScan3dCoordinateReferenceSelector_RotationY,`
`ChunkScan3dCoordinateReferenceSelector_RotationZ,`
`ChunkScan3dCoordinateReferenceSelector_TranslationX,`
`ChunkScan3dCoordinateReferenceSelector_TranslationY,`
`ChunkScan3dCoordinateReferenceSelector_TranslationZ,`
`NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }`
- `enum DeviceTapGeometryEnums {`
`DeviceTapGeometry_Geometry_1X_1Y,`
`DeviceTapGeometry_Geometry_1X2_1Y,`
`DeviceTapGeometry_Geometry_1X2_1Y2,`
`DeviceTapGeometry_Geometry_2X_1Y,`
`DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,`
`DeviceTapGeometry_Geometry_2XE_1Y2,`
`DeviceTapGeometry_Geometry_2XM_1Y,`
`DeviceTapGeometry_Geometry_2XM_1Y2,`
`DeviceTapGeometry_Geometry_1X_1Y2,`
`DeviceTapGeometry_Geometry_1X_2YE,`
`DeviceTapGeometry_Geometry_1X3_1Y,`
`DeviceTapGeometry_Geometry_3X_1Y,`
`DeviceTapGeometry_Geometry_1X,`
`DeviceTapGeometry_Geometry_1X2,`
`DeviceTapGeometry_Geometry_2X,`
`DeviceTapGeometry_Geometry_2XE,`
`DeviceTapGeometry_Geometry_2XM,`
`DeviceTapGeometry_Geometry_1X3,`
`DeviceTapGeometry_Geometry_3X,`
`DeviceTapGeometry_Geometry_1X4_1Y,`
`DeviceTapGeometry_Geometry_4X_1Y,`
`DeviceTapGeometry_Geometry_2X2_1Y,`
`DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,`
`DeviceTapGeometry_Geometry_1X2_2YE,`
`DeviceTapGeometry_Geometry_2X_2YE,`
`DeviceTapGeometry_Geometry_2XE_2YE,`
`DeviceTapGeometry_Geometry_2XM_2YE,`
`DeviceTapGeometry_Geometry_1X4,`
`DeviceTapGeometry_Geometry_4X,`
`DeviceTapGeometry_Geometry_2X2,`
`DeviceTapGeometry_Geometry_2X2E,`
`DeviceTapGeometry_Geometry_2X2M,`
`DeviceTapGeometry_Geometry_1X8_1Y,`
`DeviceTapGeometry_Geometry_8X_1Y,`
`DeviceTapGeometry_Geometry_4X2_1Y,`
`DeviceTapGeometry_Geometry_2X2E_2YE,`
`DeviceTapGeometry_Geometry_1X8,`
`DeviceTapGeometry_Geometry_8X,`
`DeviceTapGeometry_Geometry_4X2,`
`DeviceTapGeometry_Geometry_4X2E,`
`DeviceTapGeometry_Geometry_4X2E_1Y,`
`DeviceTapGeometry_Geometry_1X10_1Y,`
`DeviceTapGeometry_Geometry_10X_1Y,`
`DeviceTapGeometry_Geometry_1X10,`
`DeviceTapGeometry_Geometry_10X,`
`NUM_DEVICETAPGEOMETRY }`
- `enum GevPhysicalLinkConfigurationEnums {`
`GevPhysicalLinkConfiguration_SingleLink,`
`GevPhysicalLinkConfiguration_MultiLink,`

```

    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }
• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }
• enum GevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }
• enum GevGVCPEExtendedStatusCodesSelectorEnums {
    GevGVCPEExtendedStatusCodesSelector_Version1_1,
    GevGVCPEExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEEXTENDEDSTATUSCODESSELECTOR }
• enum GevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }
• enum ClConfigurationEnums {
    ClConfiguration_Base,
    ClConfiguration_Medium,
    ClConfiguration_Full,
    ClConfiguration_DualBase,
    ClConfiguration_EightyBit,
    NUM_CLCONFIGURATION }
• enum ClTimeSlotsCountEnums {
    ClTimeSlotsCount_One,
    ClTimeSlotsCount_Two,
    ClTimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }
• enum CxpLinkConfigurationStatusEnums {
    CxpLinkConfigurationStatus_None,
    CxpLinkConfigurationStatus_Pending,
    CxpLinkConfigurationStatus_CXP1_X1,
    CxpLinkConfigurationStatus_CXP2_X1,
    CxpLinkConfigurationStatus_CXP3_X1,
    CxpLinkConfigurationStatus_CXP5_X1,
    CxpLinkConfigurationStatus_CXP6_X1,
    CxpLinkConfigurationStatus_CXP1_X2,
    CxpLinkConfigurationStatus_CXP2_X2,
    CxpLinkConfigurationStatus_CXP3_X2,
    CxpLinkConfigurationStatus_CXP5_X2,
    CxpLinkConfigurationStatus_CXP6_X2,
    CxpLinkConfigurationStatus_CXP1_X3,
    CxpLinkConfigurationStatus_CXP2_X3,
    CxpLinkConfigurationStatus_CXP3_X3,
    CxpLinkConfigurationStatus_CXP5_X3,
    CxpLinkConfigurationStatus_CXP6_X3,
    CxpLinkConfigurationStatus_CXP1_X4,
    CxpLinkConfigurationStatus_CXP2_X4,
    CxpLinkConfigurationStatus_CXP3_X4,
    CxpLinkConfigurationStatus_CXP5_X4,

```



```

CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

```

- **enum** CxpLinkConfigurationPreferredEnums {


```

CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,
CxpLinkConfigurationPreferred_CXP5_X6,
CxpLinkConfigurationPreferred_CXP6_X6,
NUM_CXPLINKCONFIGURATIONPREFERRED }

```
- **enum** CxpLinkConfigurationEnums {


```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,
CxpLinkConfiguration_CXP1_X3,
CxpLinkConfiguration_CXP2_X3,

```

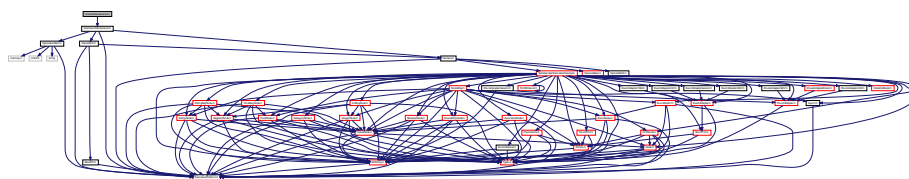
```

CxpLinkConfiguration_CXP3_X3,
CxpLinkConfiguration_CXP5_X3,
CxpLinkConfiguration_CXP6_X3,
CxpLinkConfiguration_CXP1_X4,
CxpLinkConfiguration_CXP2_X4,
CxpLinkConfiguration_CXP3_X4,
CxpLinkConfiguration_CXP5_X4,
CxpLinkConfiguration_CXP6_X4,
CxpLinkConfiguration_CXP1_X5,
CxpLinkConfiguration_CXP2_X5,
CxpLinkConfiguration_CXP3_X5,
CxpLinkConfiguration_CXP5_X5,
CxpLinkConfiguration_CXP6_X5,
CxpLinkConfiguration_CXP1_X6,
CxpLinkConfiguration_CXP2_X6,
CxpLinkConfiguration_CXP3_X6,
CxpLinkConfiguration_CXP5_X6,
CxpLinkConfiguration_CXP6_X6,
NUM_CXPLINKCONFIGURATION }
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXPCONNECTIONTESTMODE }
• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

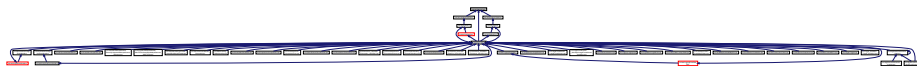
```

16.13 include/CameraList.h File Reference

Include dependency graph for CameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

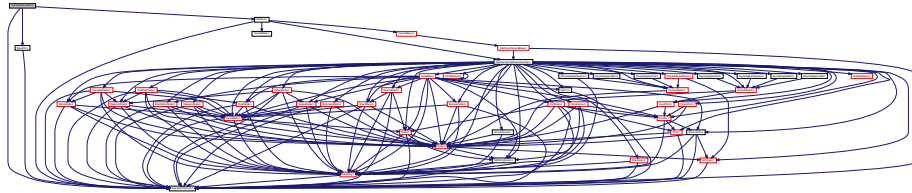
- class [CameraList](#)
Used to hold a list of camera objects.

Namespaces

- [Spinnaker](#)

16.14 include/CameraPtr.h File Reference

Include dependency graph for CameraPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

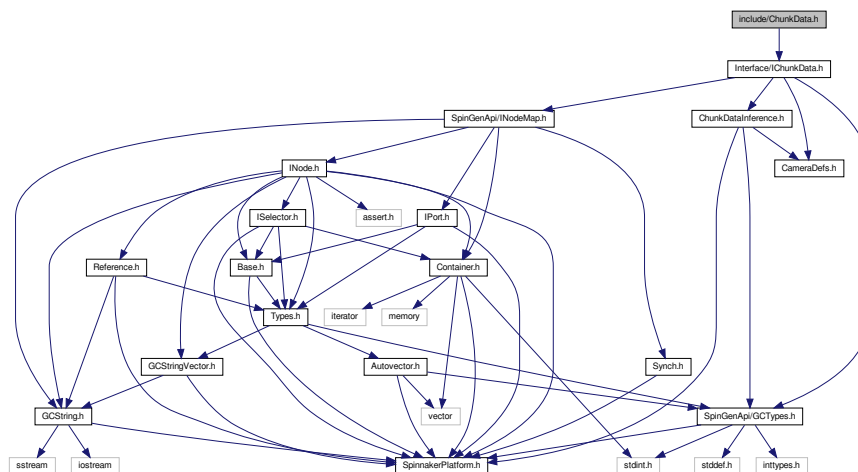
- class [CameraPtr](#)
A reference tracked pointer to a camera object.

Namespaces

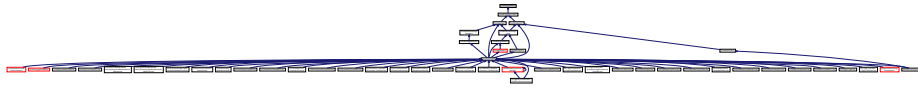
- [Spinnaker](#)

16.15 include/ChunkData.h File Reference

Include dependency graph for ChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ChunkData](#)

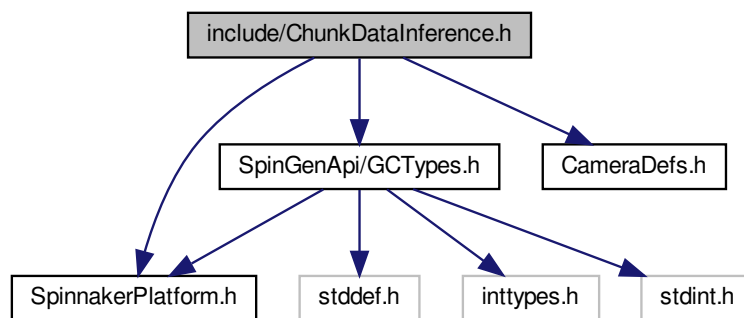
The chunk data which contains additional information about an image.

Namespaces

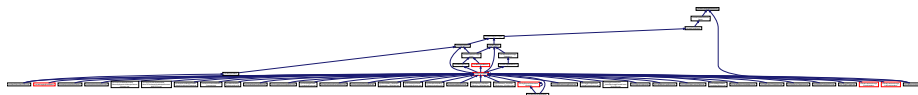
- [Spinnaker](#)

16.16 include/ChunkDataInference.h File Reference

Include dependency graph for ChunkDataInference.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [InferenceBoxRect](#)

Inference Bounding Box Type Data Structures.

- struct [InferenceBoxCircle](#)
- struct [InferenceBoxRotatedRect](#)
- struct [InferenceBoundingBox](#)

Inference Bounding Boxes data structure.

- class [InferenceBoundingBoxResult](#)

An inference bounding boxes object which holds information about the detected bounding boxes.

Namespaces

- [Spinnaker](#)

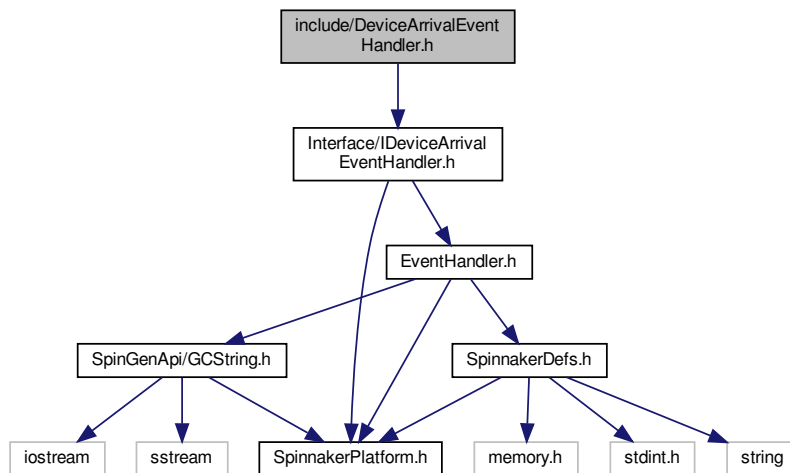
Enumerations

- enum [InferenceBoxType](#) {
[INFERENCE_BOX_TYPE_RECTANGLE](#) = 0,
[INFERENCE_BOX_TYPE_CIRCLE](#) = 1,
[INFERENCE_BOX_TYPE_ROTATED_RECTANGLE](#) = 2 }

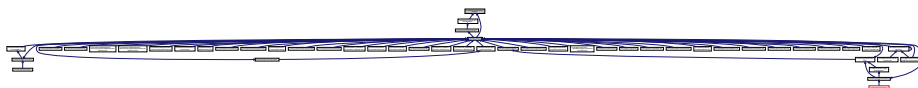
Inference Bounding Box Type.

16.17 include/DeviceArrivalEventHandler.h File Reference

Include dependency graph for DeviceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

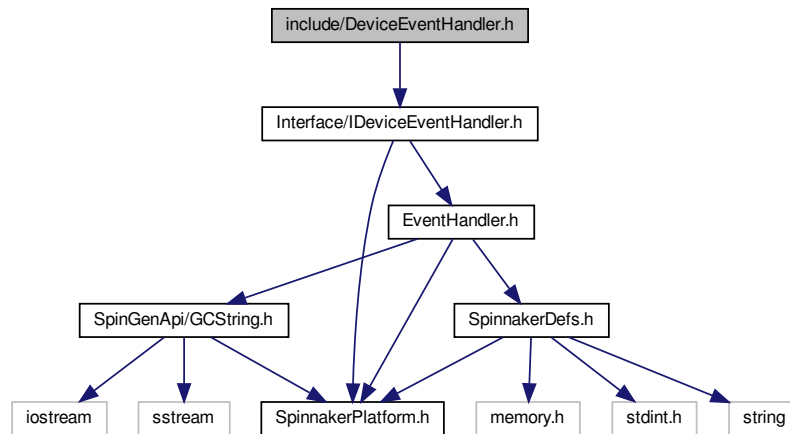
- class [DeviceArrivalEventHandler](#)
An event handler for capturing the device arrival event.

Namespaces

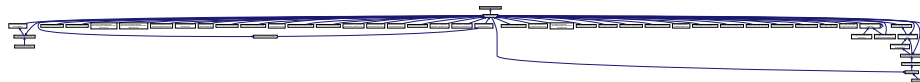
- [Spinnaker](#)

16.18 include/DeviceEventHandler.h File Reference

Include dependency graph for DeviceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DeviceEventHandler](#)

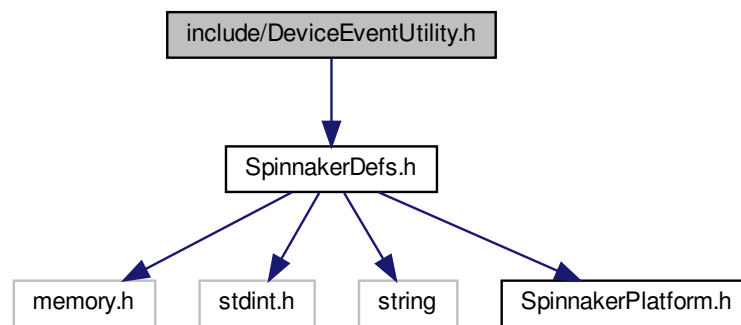
A handler to device events.

Namespaces

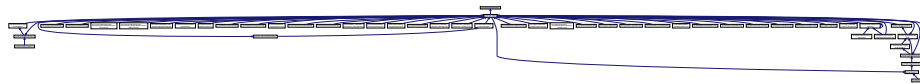
- [Spinnaker](#)

16.19 include/DeviceEventUtility.h File Reference

Include dependency graph for DeviceEventUtility.h:



This graph shows which files directly or indirectly include this file:



Classes

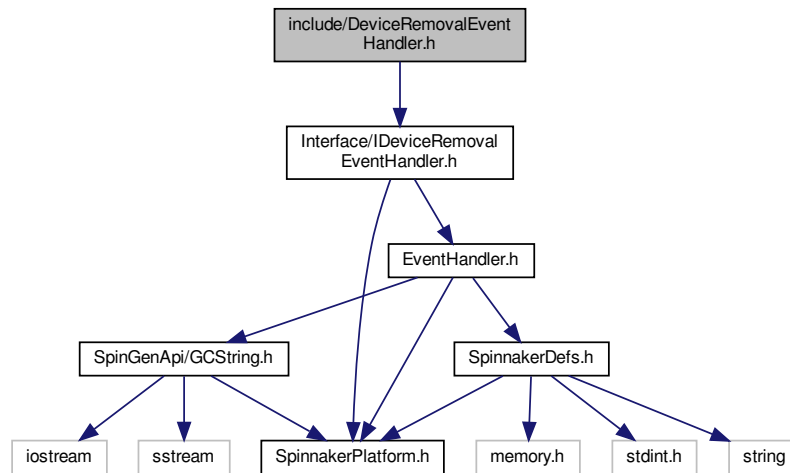
- class [DeviceEventUtility](#)

Namespaces

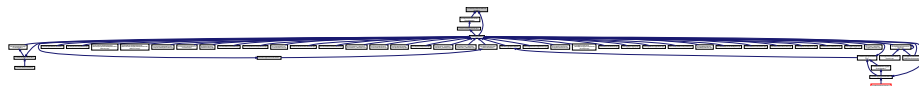
- [Spinnaker](#)

16.20 include/DeviceRemovalEventHandler.h File Reference

Include dependency graph for DeviceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DeviceRemovalEventHandler](#)

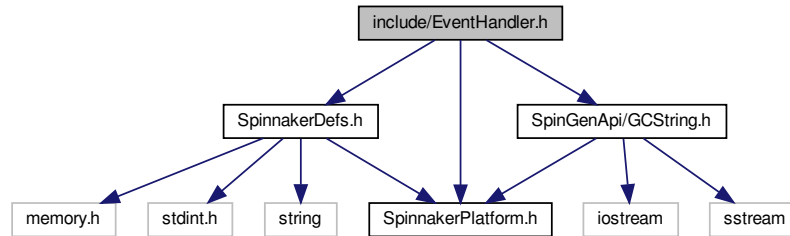
An event handler for capturing the device removal event.

Namespaces

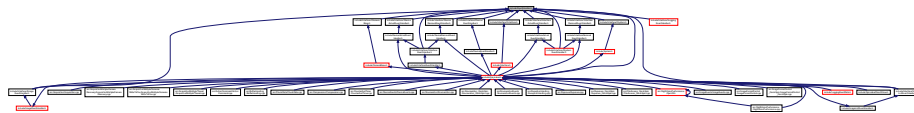
- [Spinnaker](#)

16.21 include/EventHandler.h File Reference

Include dependency graph for EventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

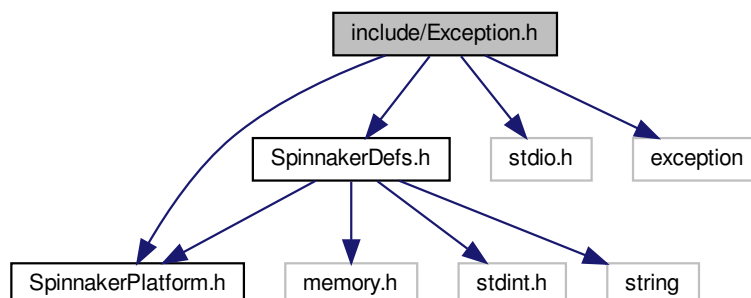
- class [EventHandler](#)
The base class for all event handler types.

Namespaces

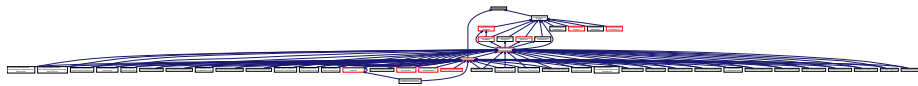
- [Spinnaker](#)

16.22 include/Exception.h File Reference

Include dependency graph for Exception.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Exception](#)

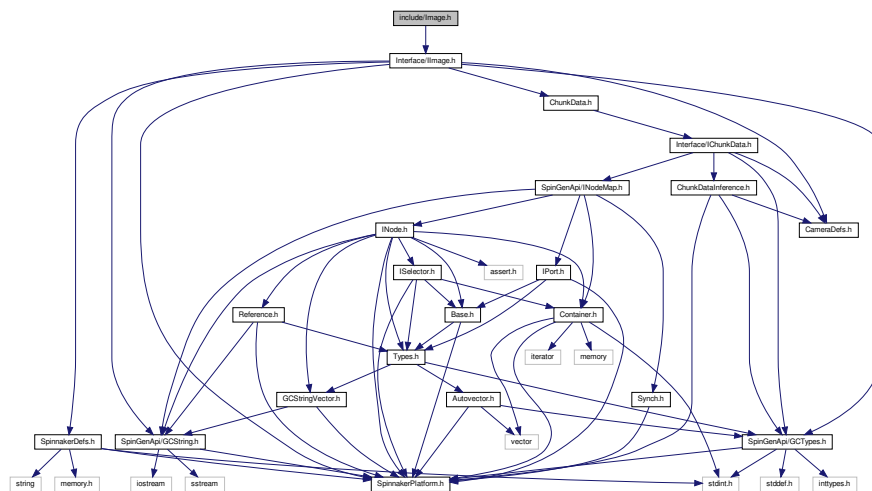
The [Exception](#) object represents an error that is returned from the library.

Namespaces

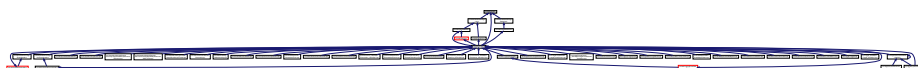
- [Spinnaker](#)

16.23 include/Image.h File Reference

Include dependency graph for Image.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Image](#)

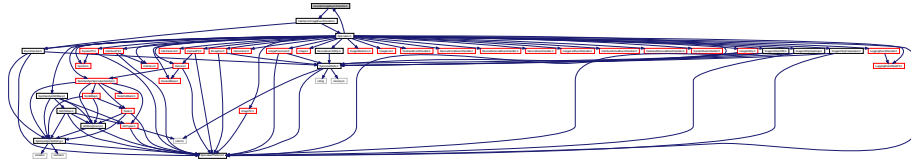
The [Image](#) object class.

Namespaces

- [Spinnaker](#)

16.24 include/ImageEventHandler.h File Reference

Include dependency graph for ImageEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

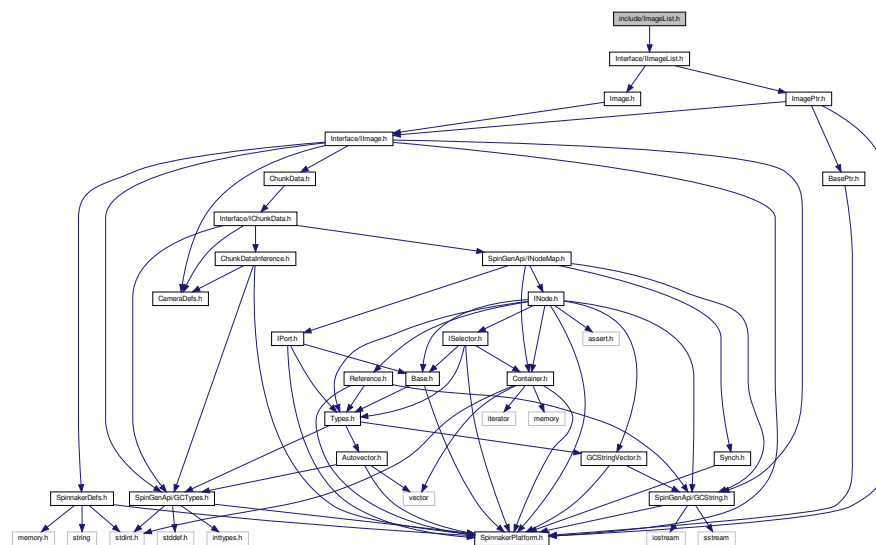
- class [ImageEventHandler](#)
A handler for capturing image arrival events.

Namespaces

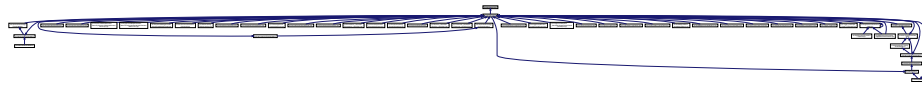
- [Spinnaker](#)

16.25 include/ImageList.h File Reference

Include dependency graph for ImageList.h:



This graph shows which files directly or indirectly include this file:



Classes

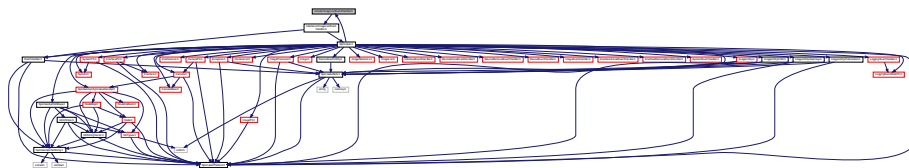
- class [ImageList](#)

Namespaces

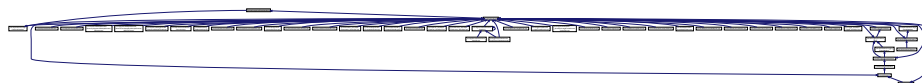
- [Spinnaker](#)

16.26 include/ImageListEventHandler.h File Reference

Include dependency graph for ImageListEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

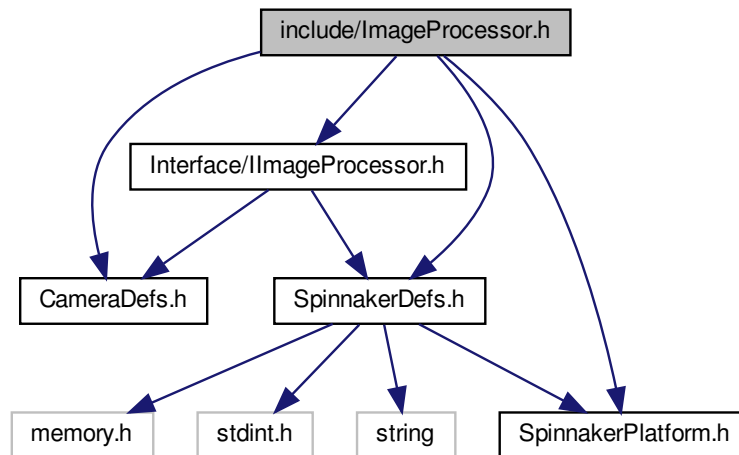
- class [ImageListEventHandler](#)
A handler for capturing image list arrival events.

Namespaces

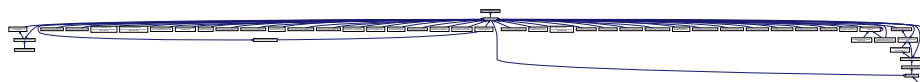
- [Spinnaker](#)

16.27 include/ImageProcessor.h File Reference

Include dependency graph for ImageProcessor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageProcessor](#)

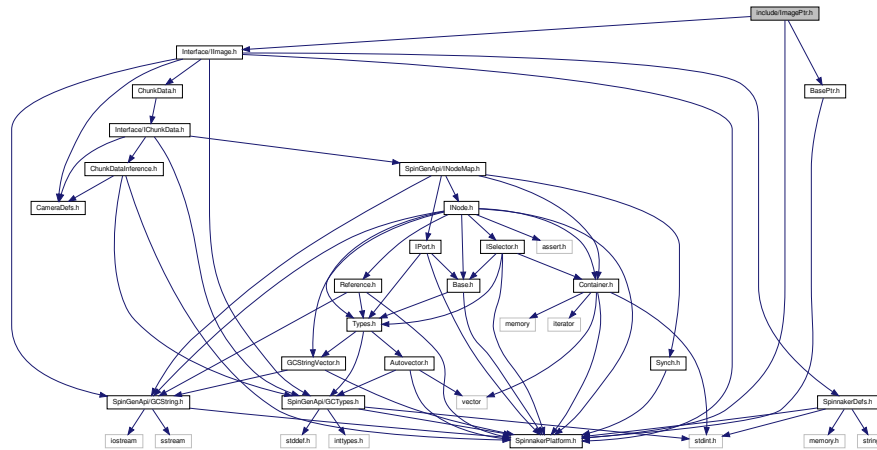
Image post processing class for converting a source image to another pixel format.

Namespaces

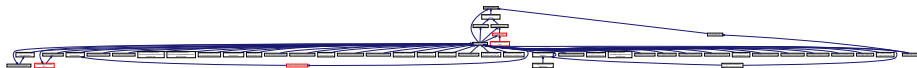
- [Spinnaker](#)

16.28 include/ImagePtr.h File Reference

Include dependency graph for ImagePtr.h:



This graph shows which files directly or indirectly include this file:



Classes

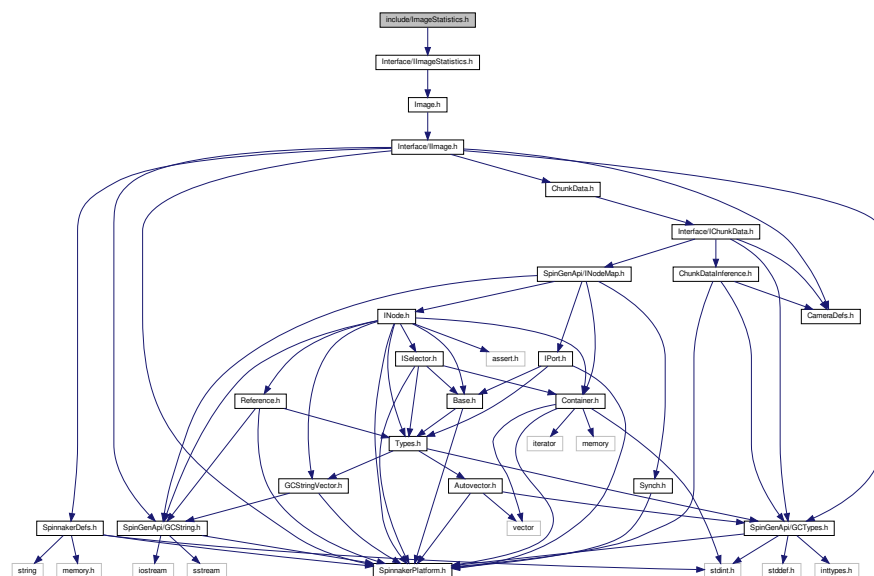
- class [ImagePtr](#)

A reference tracked pointer to an image object.

Namespaces

- [Spinnaker](#)

Include dependency graph for ImageStatistics.h:



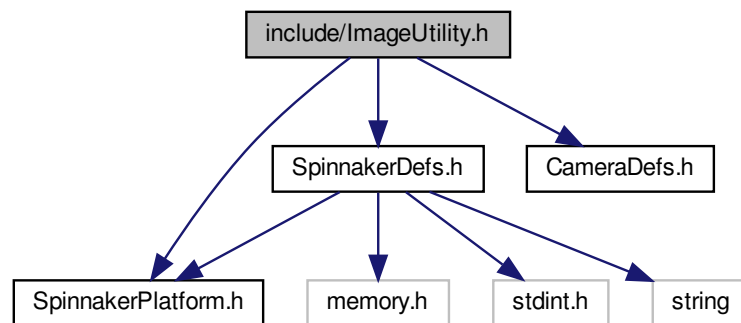
- class ImageStatistics

Namespaces

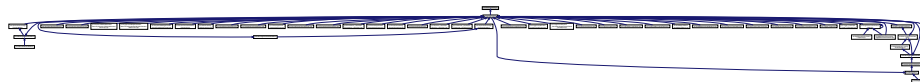
- Spinnaker

16.30 include/ImageUtility.h File Reference

Include dependency graph for ImageUtility.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtility](#)

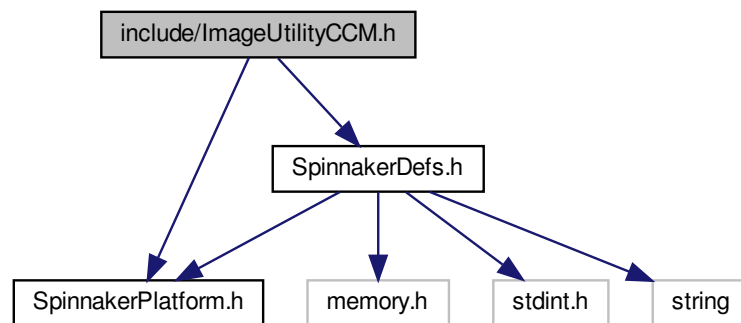
Static helper functions for the image object class.

Namespaces

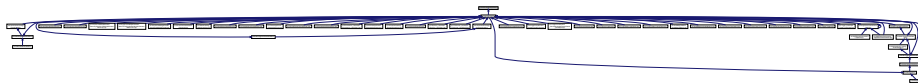
- [Spinnaker](#)

16.31 include/ImageUtilityCCM.h File Reference

Include dependency graph for ImageUtilityCCM.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtilityCCM](#)

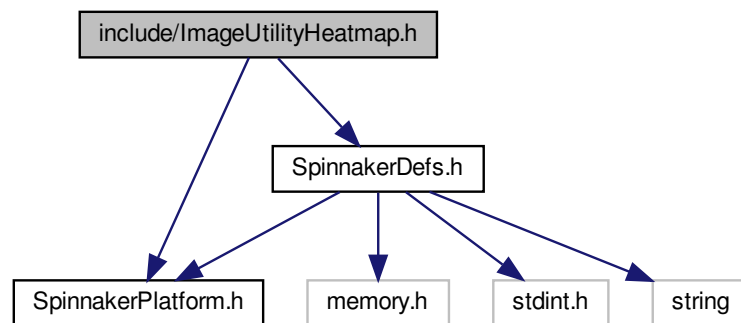
Static function to create color corrected images from an image object.

Namespaces

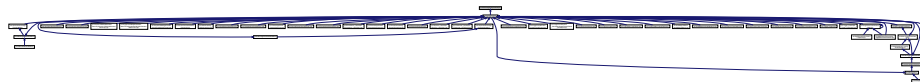
- [Spinnaker](#)

16.32 include/ImageUtilityHeatmap.h File Reference

Include dependency graph for ImageUtilityHeatmap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtilityHeatmap](#)

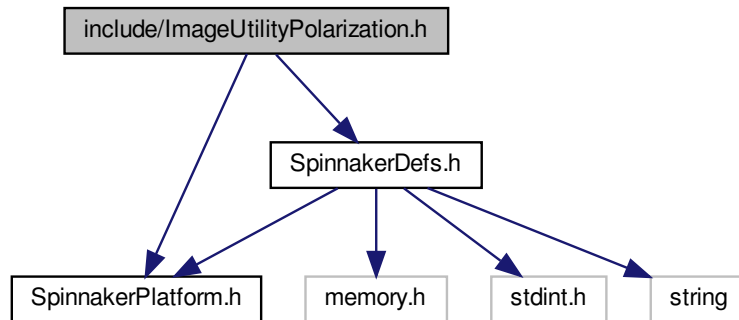
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

Namespaces

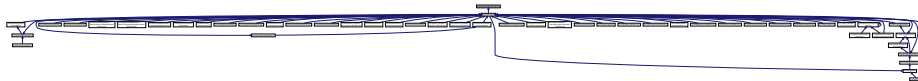
- [Spinnaker](#)

16.33 include/ImageUtilityPolarization.h File Reference

Include dependency graph for ImageUtilityPolarization.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageUtilityPolarization](#)

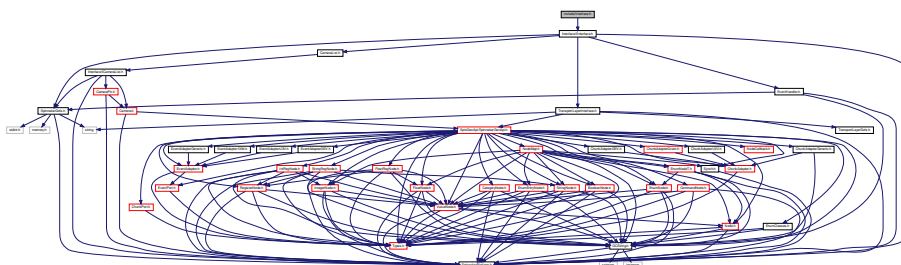
Static functions to create polarization images from image objects of pixel format `Polarized8` and `BayerRGPolarized8`.

Namespaces

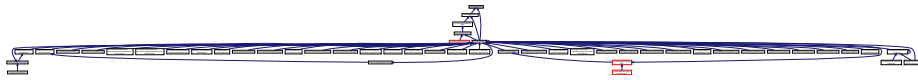
- [Spinnaker](#)

16.34 include/Interface.h File Reference

Include dependency graph for Interface.h:



This graph shows which files directly or indirectly include this file:



Classes

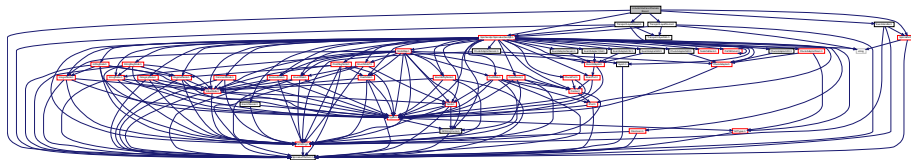
- class [Interface](#)
An interface object which holds a list of cameras.

Namespaces

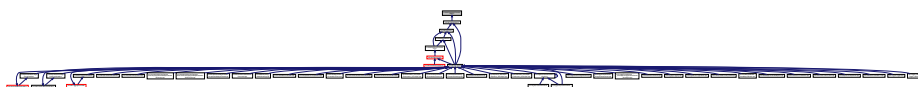
- [Spinnaker](#)

16.35 include/Interface/ICameraBase.h File Reference

Include dependency graph for ICameraBase.h:



This graph shows which files directly or indirectly include this file:



Classes

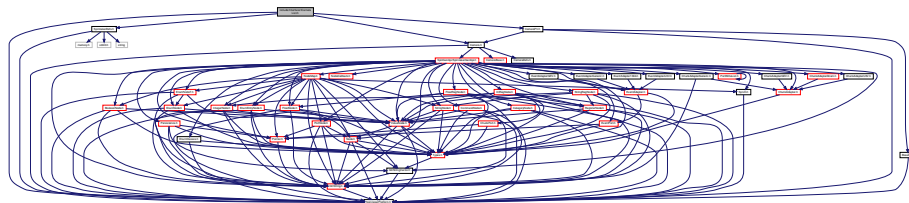
- class [ICameraBase](#)
The interface file for base class for the camera object.

Namespaces

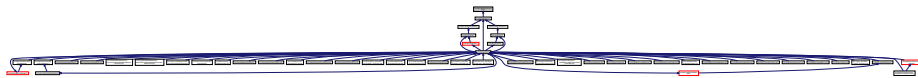
- [Spinnaker](#)

16.36 include/Interface/ICameraList.h File Reference

Include dependency graph for ICameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

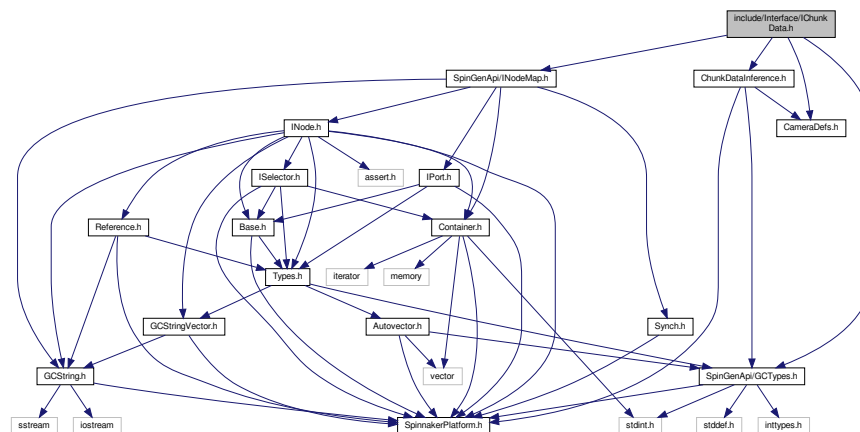
- class [ICameraList](#)
Used to hold a list of camera objects.

Namespaces

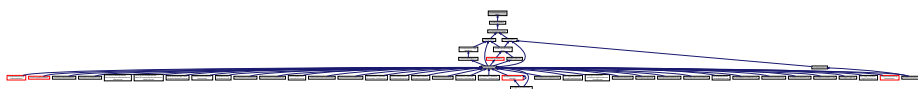
- [Spinnaker](#)

16.37 include/Interface/IChunkData.h File Reference

Include dependency graph for IChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

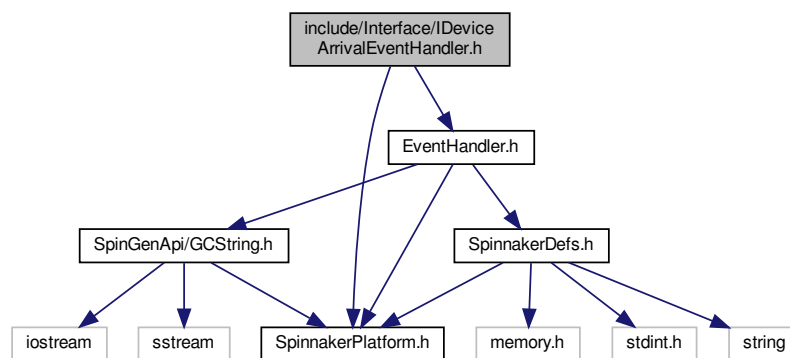
- class [IChunkData](#)
The [Interface](#) file for [ChunkData](#).

Namespaces

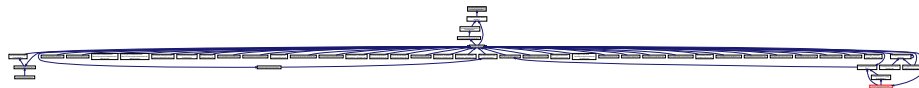
- [Spinnaker](#)

16.38 include/Interface/IDeviceArrivalEventHandler.h File Reference

Include dependency graph for IDeviceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

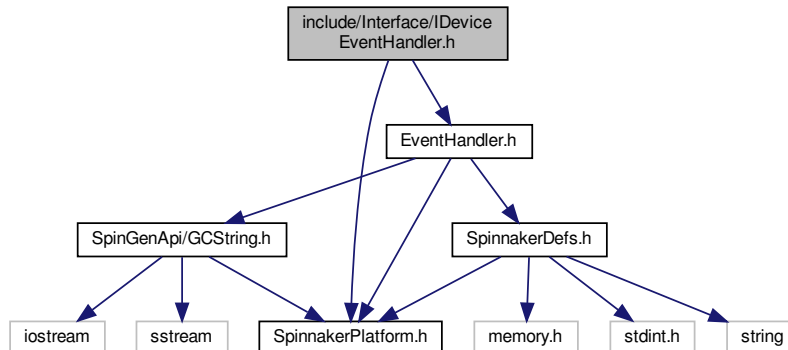
- class [IDeviceArrivalEventHandler](#)

Namespaces

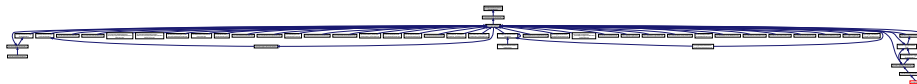
- [Spinnaker](#)

16.39 include/Interface/IDeviceEventHandler.h File Reference

Include dependency graph for IDeviceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

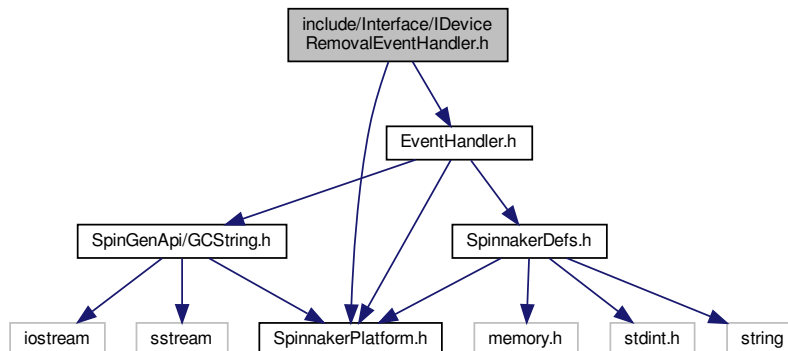
- class [IDeviceEventHandler](#)

Namespaces

- [Spinnaker](#)

16.40 include/Interface/IDeviceRemovalEventHandler.h File Reference

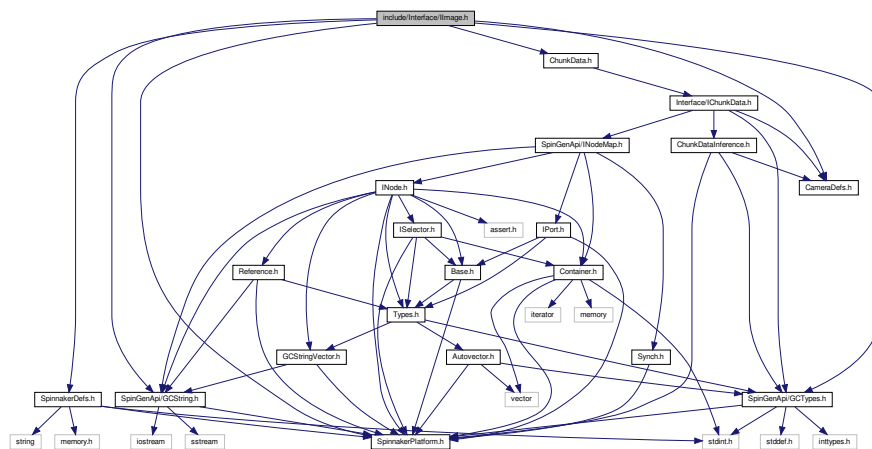
Include dependency graph for IDeviceRemovalEventHandler.h:



- class `IDeviceRemovalEventHandler`

- Spinnaker

Include dependency graph for Image.h:



This diagram illustrates the internal structure of a ship's hull, showing the keel, ribs, and various internal components like the engine and fuel tank. The hull is shown in a cross-section, revealing the internal structure and the placement of various components. The keel is the central structural member, and the ribs are the transverse members that support the hull plating. The diagram also shows the location of the engine, fuel tank, and other internal systems.

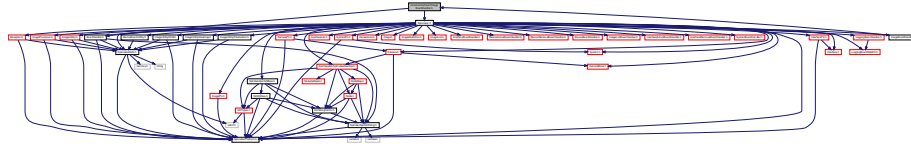
- class `Image`
The interface file for `Image`.

Namespaces

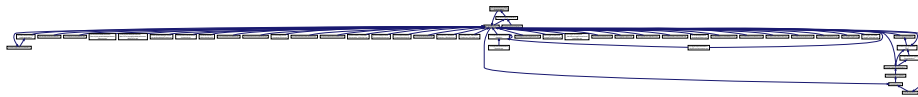
- [Spinnaker](#)

16.42 include/Interface/IImageEventHandler.h File Reference

Include dependency graph for IImageEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

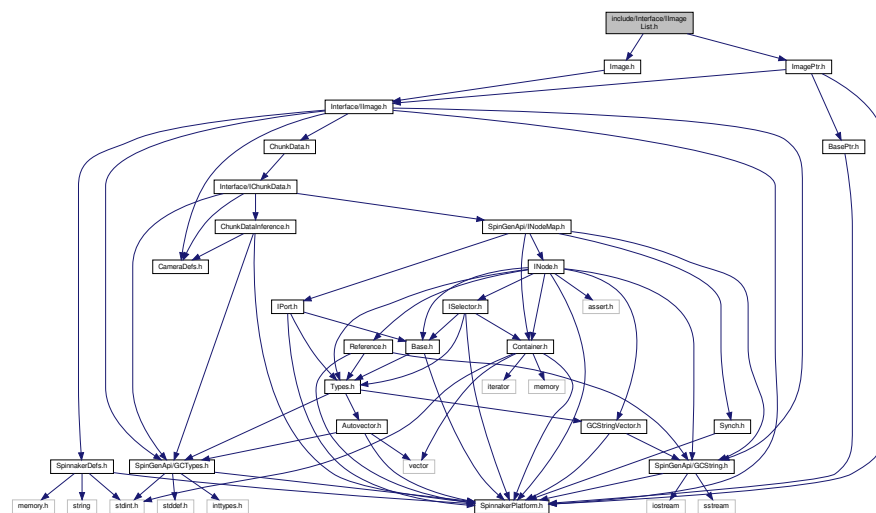
- class [IImageEventHandler](#)

Namespaces

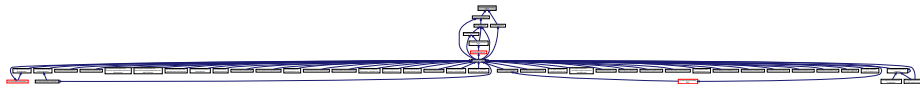
- [Spinnaker](#)

16.43 include/Interface/IImageList.h File Reference

Include dependency graph for IImageList.h:



This graph shows which files directly or indirectly include this file:



Classes

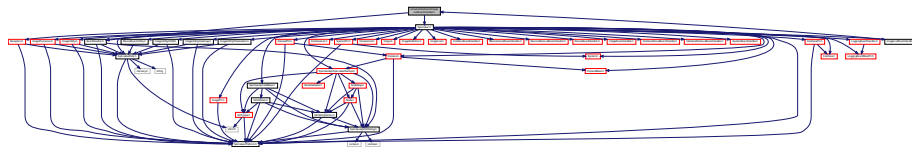
- class [IImageList](#)
Used to hold a list of image objects.

Namespaces

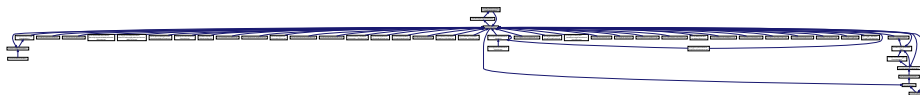
- [Spinnaker](#)

16.44 include/Interface/IImageListEventHandler.h File Reference

Include dependency graph for IImageListEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

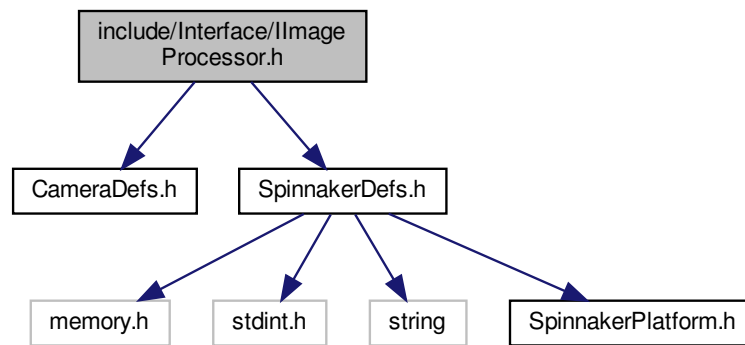
- class [IImageListEventHandler](#)

Namespaces

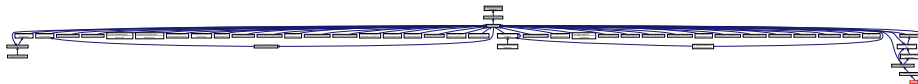
- [Spinnaker](#)

16.45 include/Interface/IImageProcessor.h File Reference

Include dependency graph for IImageProcessor.h:



This graph shows which files directly or indirectly include this file:



Classes

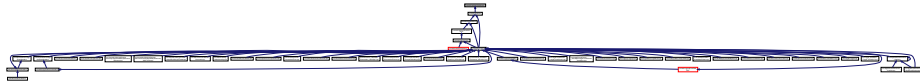
- class [IImageProcessor](#)

Used to hold a list of image objects.

Namespaces

- [Spinnaker](#)

This graph shows which files directly or indirectly include this file:



Classes

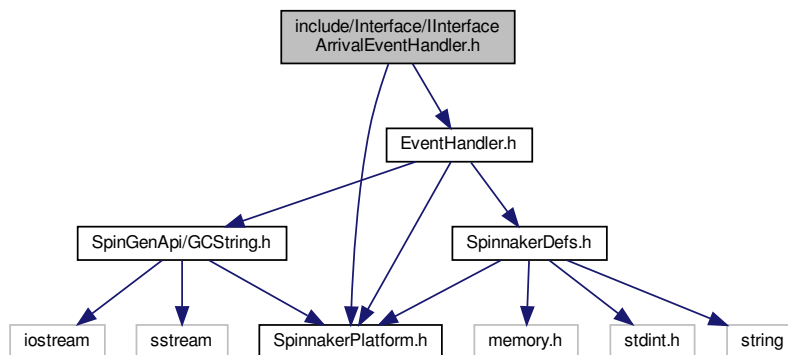
- class [IInterface](#)
The interface file for [IInterface](#).

Namespaces

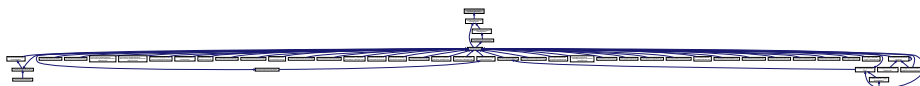
- [Spinnaker](#)

16.48 include/Interface/IInterfaceArrivalEventHandler.h File Reference

Include dependency graph for IInterfaceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

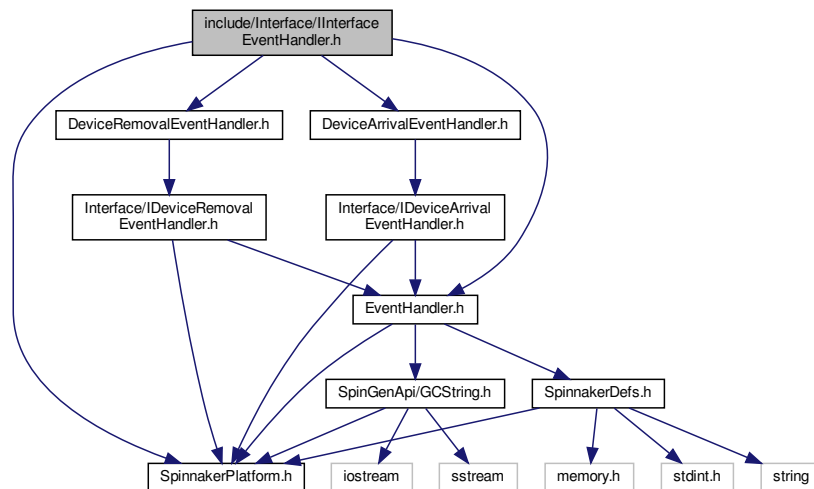
- class [IInterfaceArrivalEventHandler](#)

Namespaces

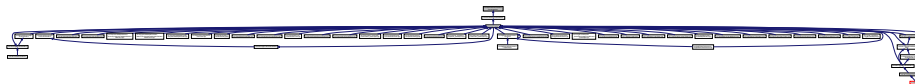
- [Spinnaker](#)

16.49 include/Interface/IInterfaceEventHandler.h File Reference

Include dependency graph for IInterfaceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

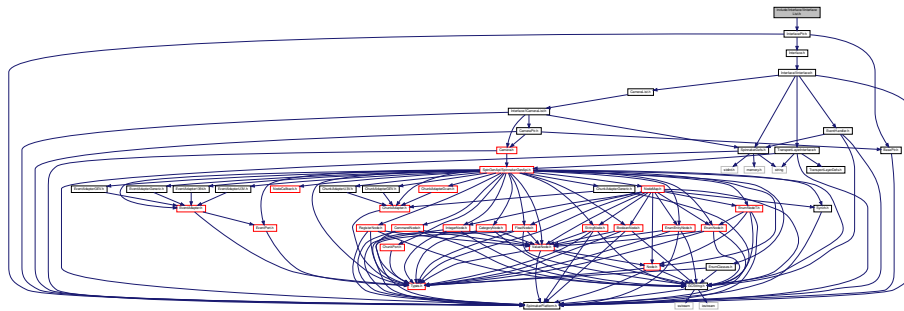
- class [IInterfaceEventHandler](#)

Namespaces

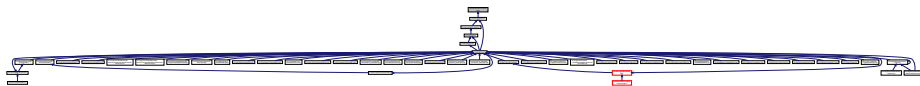
- [Spinnaker](#)

16.50 include/Interface/IInterfaceList.h File Reference

Include dependency graph for IInterfaceList.h:



This graph shows which files directly or indirectly include this file:



Classes

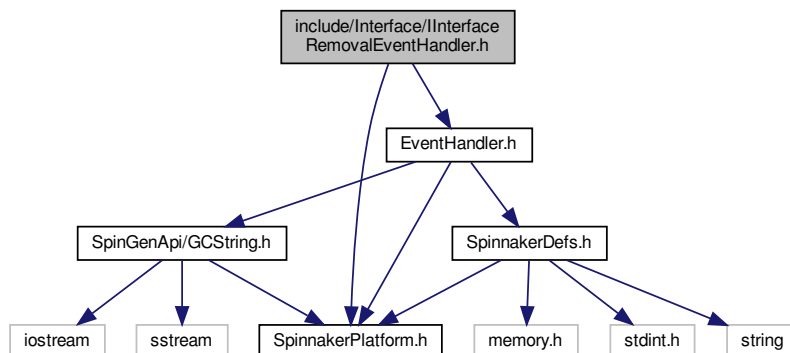
- class [IInterfaceList](#)
The interface file for [IInterfaceList](#) class.

Namespaces

- [Spinnaker](#)

16.51 include/Interface/IInterfaceRemovalEventHandler.h File Reference

Include dependency graph for IInterfaceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

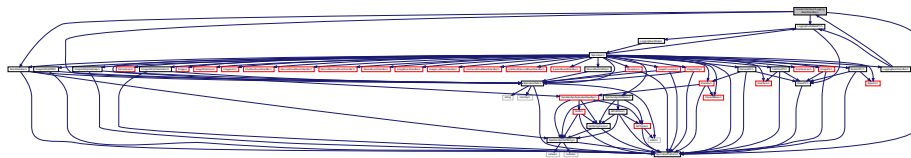
- class [InterfaceRemovalEventHandler](#)

Namespaces

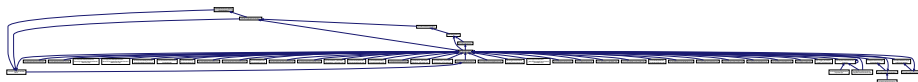
- [Spinnaker](#)

16.52 include/Interface/ILoggingEventHandler.h File Reference

Include dependency graph for ILoggingEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

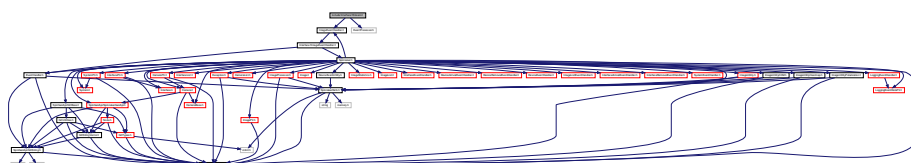
- class [ILoggingEventHandler](#)

Namespaces

- [Spinnaker](#)

16.53 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:



Classes

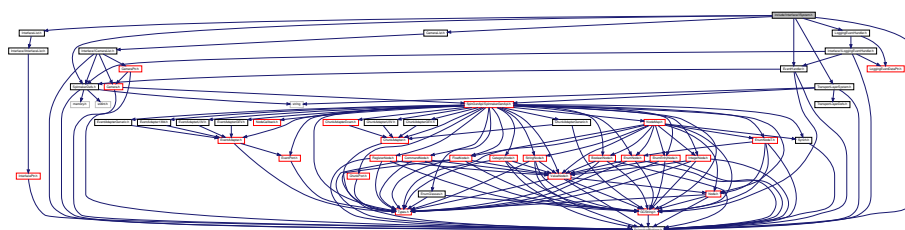
- class [IDataStream](#)

Namespaces

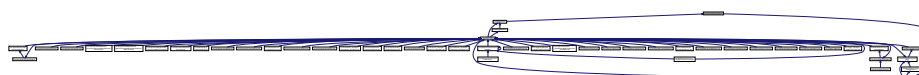
- [Spinnaker](#)

16.54 include/Interface/ISystem.h File Reference

Include dependency graph for ISystem.h:



This graph shows which files directly or indirectly include this file:



Classes

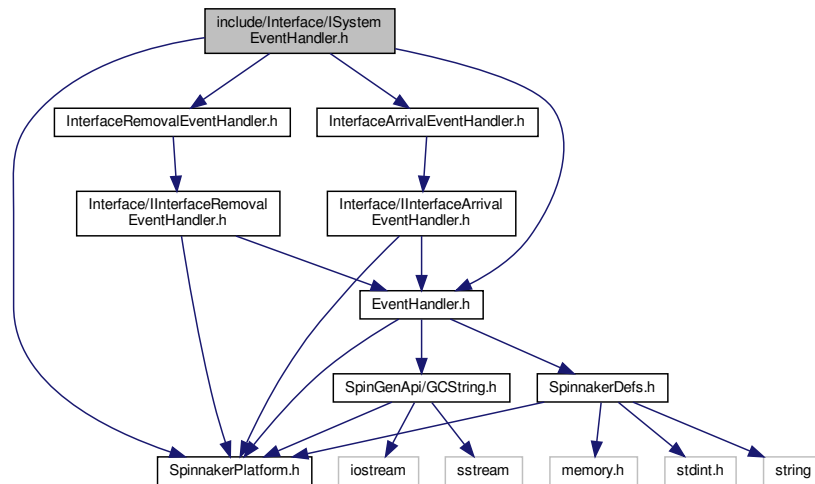
- class [ISystem](#)
The interface file for [System](#).

Namespaces

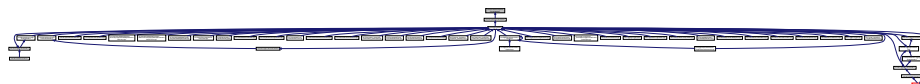
- [Spinnaker](#)

16.55 include/Interface/ISystemEventHandler.h File Reference

Include dependency graph for ISystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

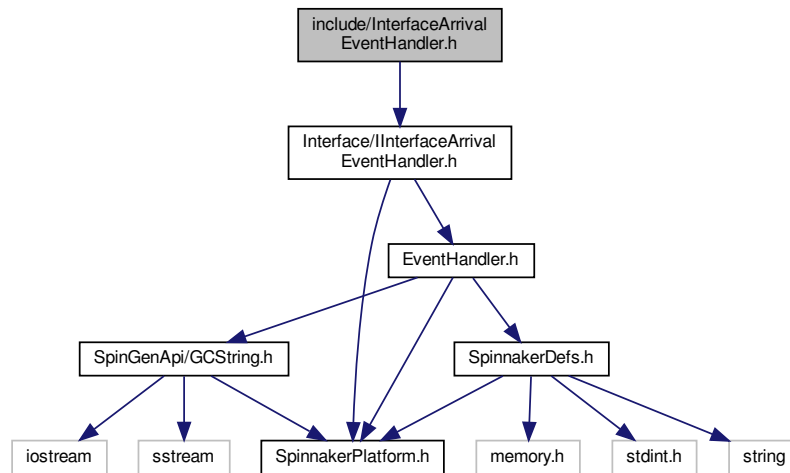
- class [ISystemEventHandler](#)

Namespaces

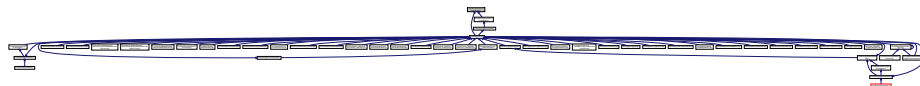
- [Spinnaker](#)

16.56 include/InterfaceArrivalEventHandler.h File Reference

Include dependency graph for InterfaceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [InterfaceArrivalEventHandler](#)

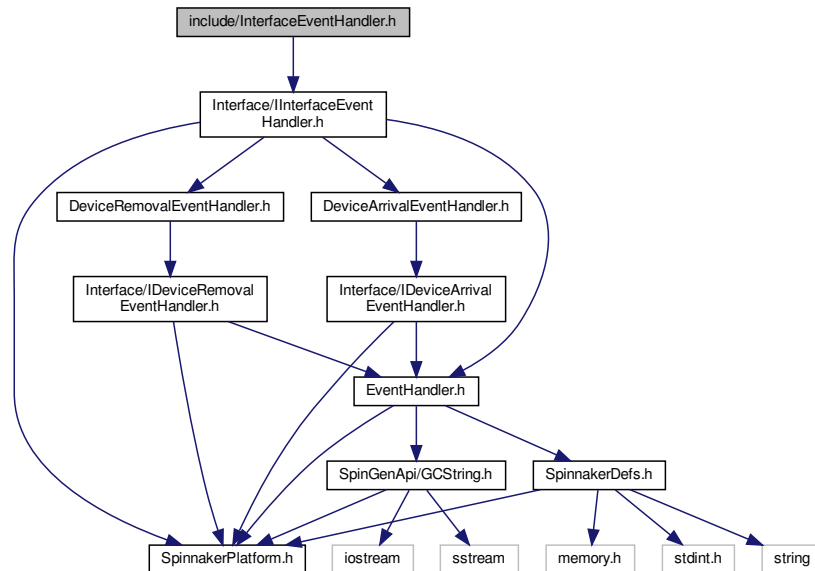
An event handler for capturing the interface arrival event.

Namespaces

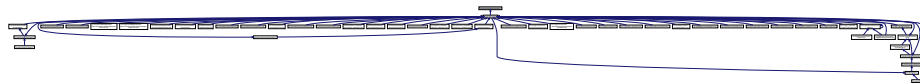
- [Spinnaker](#)

16.57 include/InterfaceEventHandler.h File Reference

Include dependency graph for InterfaceEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [InterfaceEventHandler](#)

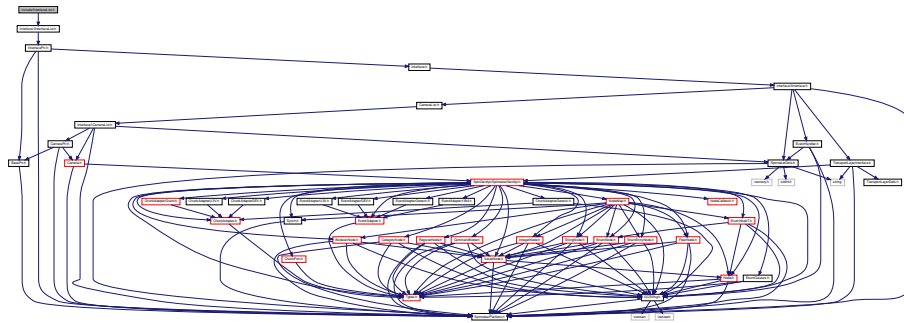
A handler to device arrival and removal events on all interfaces.

Namespaces

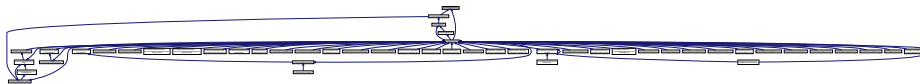
- [Spinnaker](#)

16.58 include/InterfaceList.h File Reference

Include dependency graph for InterfaceList.h:



This graph shows which files directly or indirectly include this file:



Classes

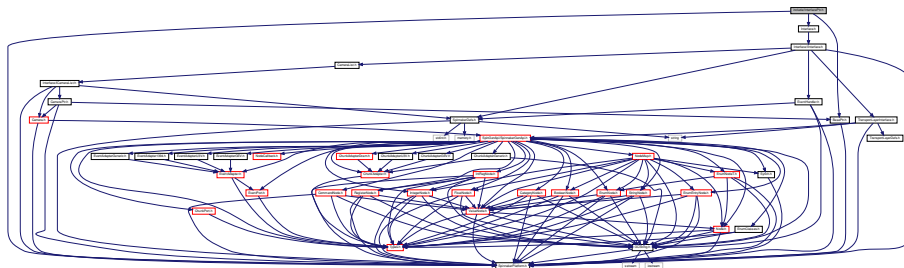
- class [InterfaceList](#)
A list of the available interfaces on the system.

Namespaces

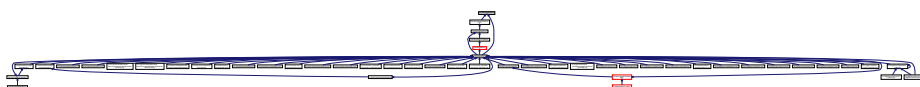
- [Spinnaker](#)

16.59 include/InterfacePtr.h File Reference

Include dependency graph for InterfacePtr.h:



This graph shows which files directly or indirectly include this file:



Classes

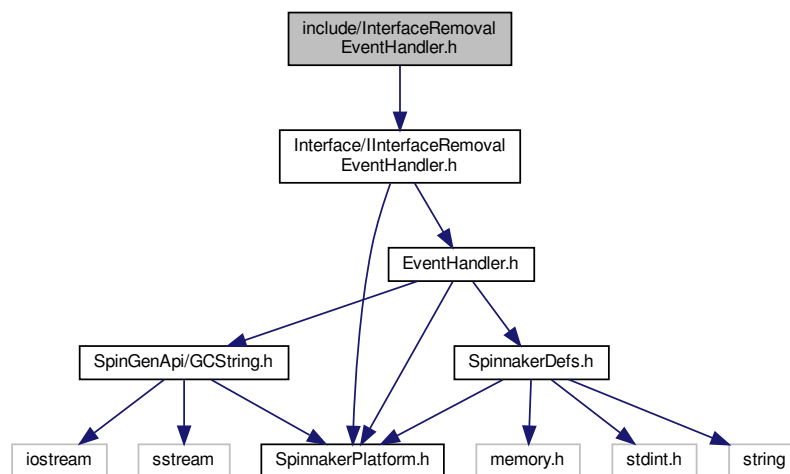
- class [InterfacePtr](#)
A reference tracked pointer to the interface object.

Namespaces

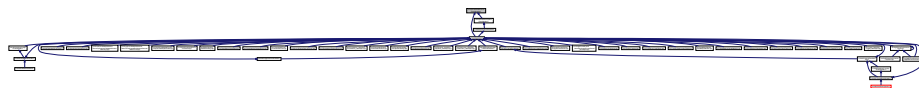
- [Spinnaker](#)

16.60 include/InterfaceRemovalEventHandler.h File Reference

Include dependency graph for InterfaceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

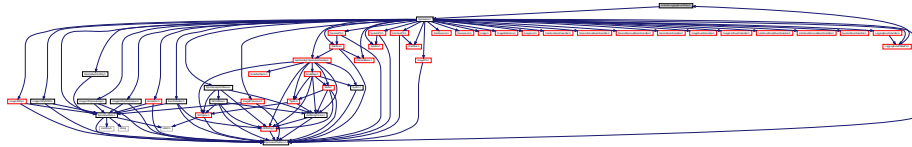
- class [InterfaceRemovalEventHandler](#)
An event handler for capturing the interface removal event.

Namespaces

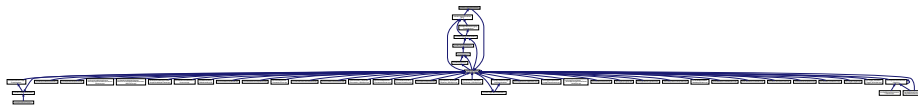
- [Spinnaker](#)

16.61 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:



This graph shows which files directly or indirectly include this file:



Classes

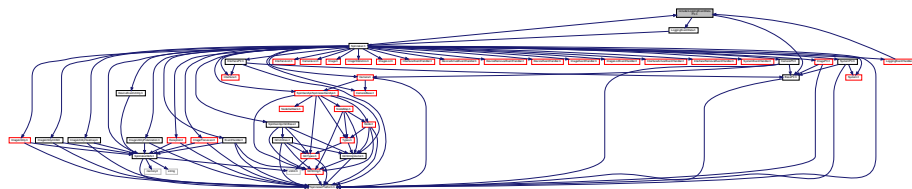
- class [LoggingEventData](#)
The *LoggingEventData* object.

Namespaces

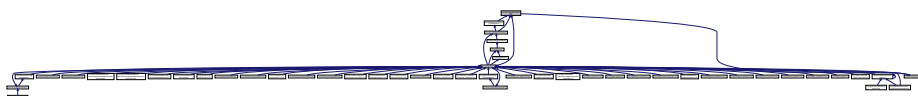
- [Spinnaker](#)

16.62 include/LoggingEventDataPtr.h File Reference

Include dependency graph for LoggingEventDataPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [LoggingEventDataPtr](#)

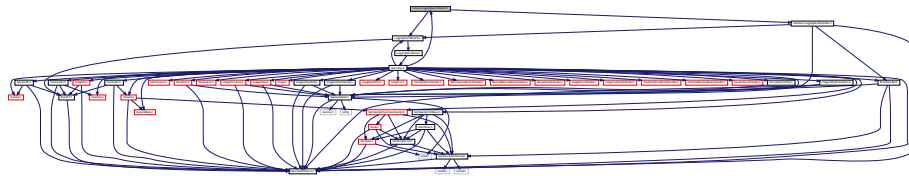
A reference tracked pointer to the `LoggingEvent` object.

Namespaces

- [Spinnaker](#)

16.63 include/LoggingEventHandler.h File Reference

Include dependency graph for `LoggingEventHandler.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [LoggingEventHandler](#)

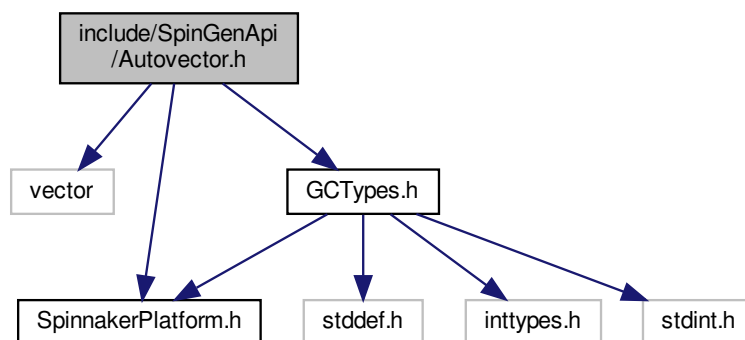
An event handler for capturing the device logging event.

Namespaces

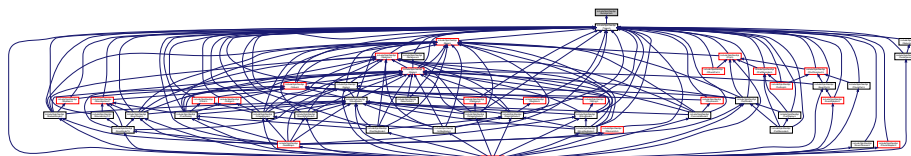
- [Spinnaker](#)

16.64 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



Classes

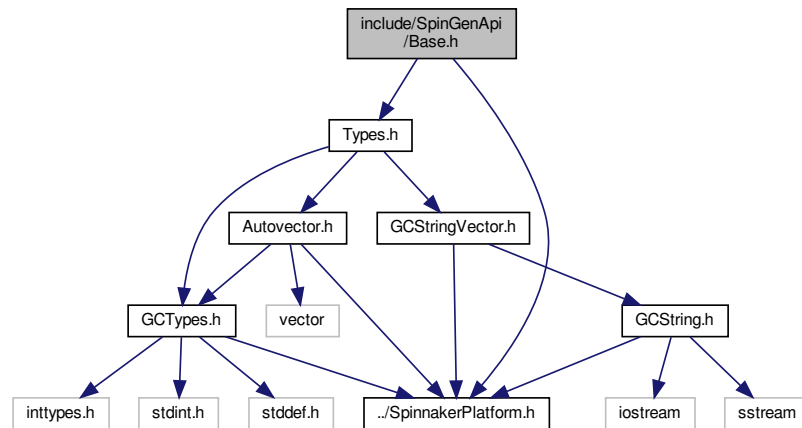
- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [double_autovector_t](#)
Vector of doubles with reference counting.

Namespaces

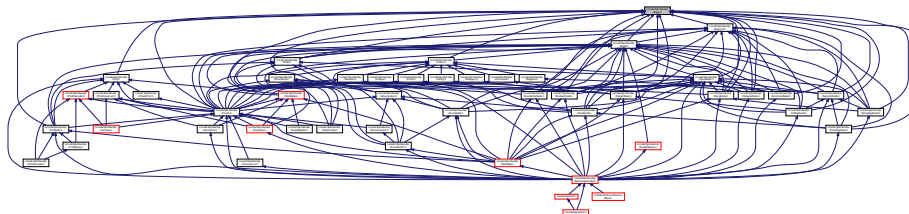
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.65 include/SpinGenApi/Base.h File Reference

Include dependency graph for Base.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

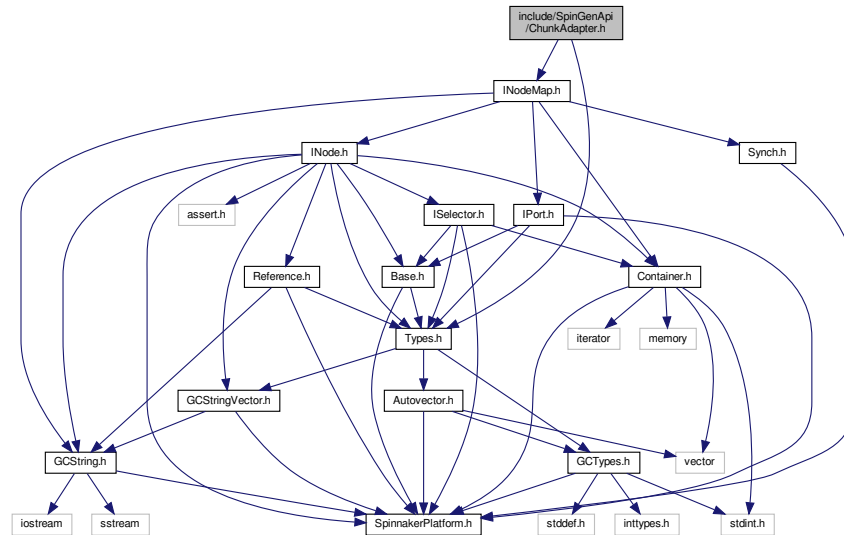
Variables

- [interface SPINNAKER_API_ABSTRACT IBase](#)

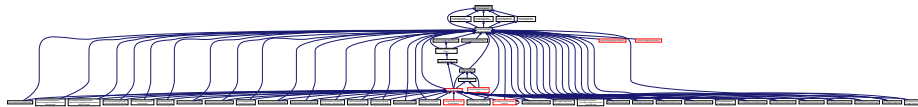
Base interface common to all nodes.

16.68 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



Classes

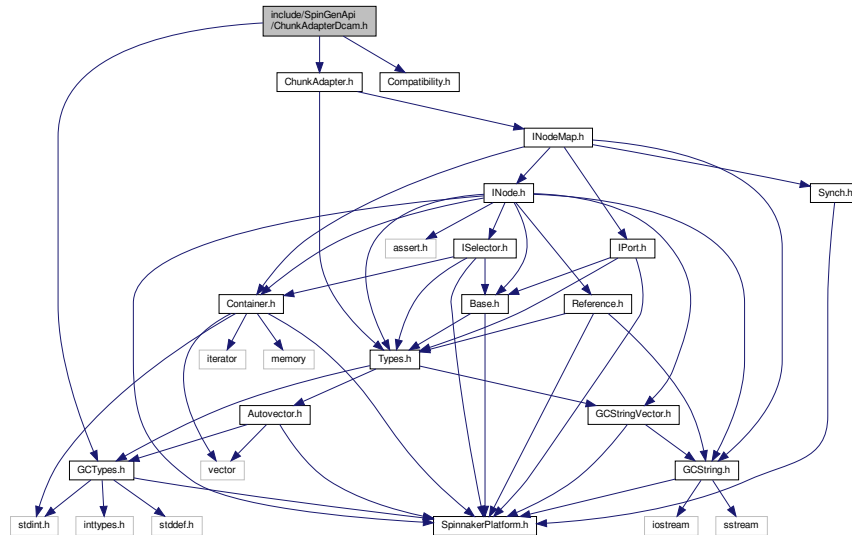
- struct [AttachStatistics_t](#)
Delivers information about the attached chunks and nodes.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.

Namespaces

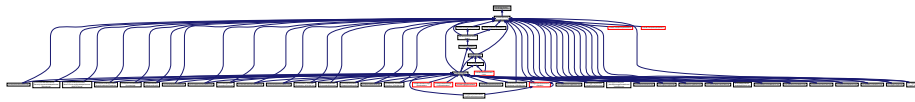
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.69 include/SpinGenApi/ChunkAdapterDcam.h File Reference

Include dependency graph for ChunkAdapterDcam.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [DCAM_CHUNK_TRAILER](#)
- struct [DCAM_CHECKSUM](#)
- class [CChunkAdapterDcam](#)

Connects a chunked DCAM buffer to a node map.

Namespaces

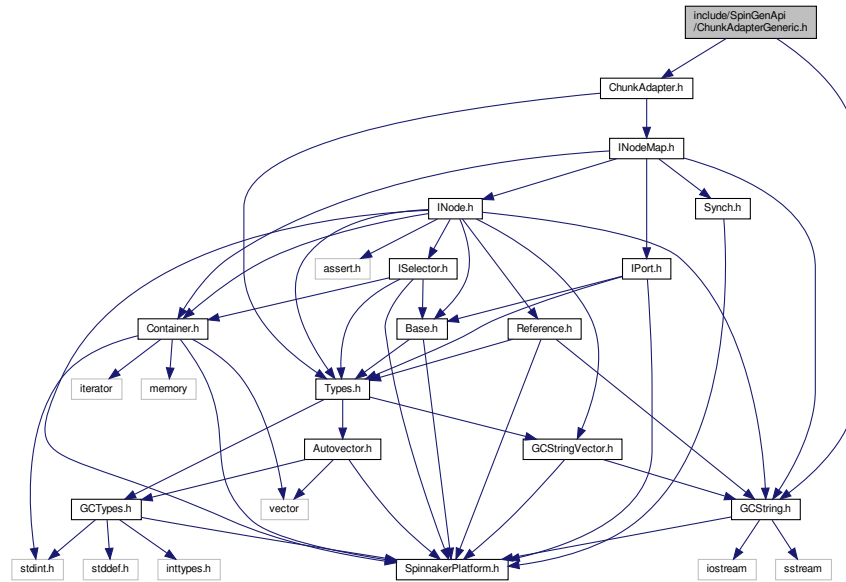
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

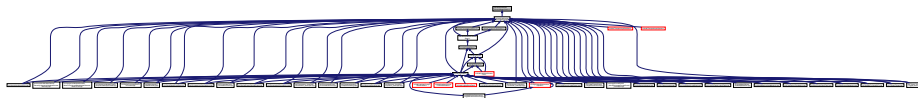
- void [SPINNAKER_API SET_GUID](#) (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)

16.70 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



Classes

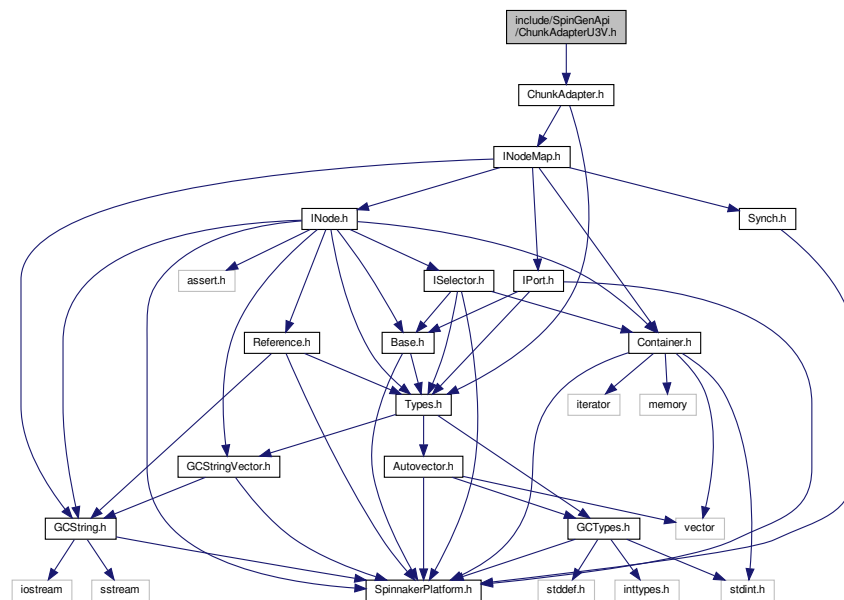
- struct [SingleChunkData_t](#)
- struct [SingleChunkDataStr_t](#)
- class [CChunkAdapterGeneric](#)

Namespaces

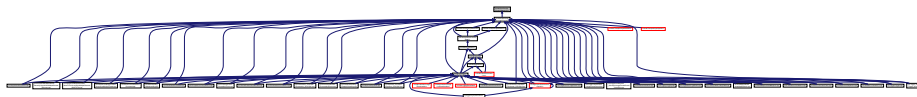
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.72 include/SpinGenApi/ChunkAdapterU3V.h File Reference

Include dependency graph for ChunkAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



Classes

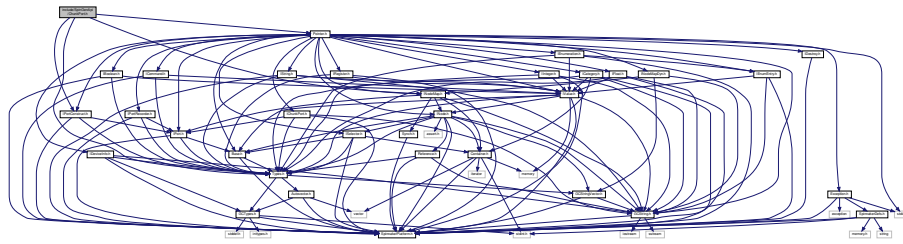
- struct [U3V_CHUNK_TRAILER](#)
header of a GVCP request packet
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.

Namespaces

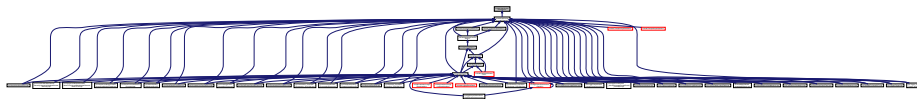
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.73 include/SpinGenApi/ChunkPort.h File Reference

Include dependency graph for ChunkPort.h:



This graph shows which files directly or indirectly include this file:



Classes

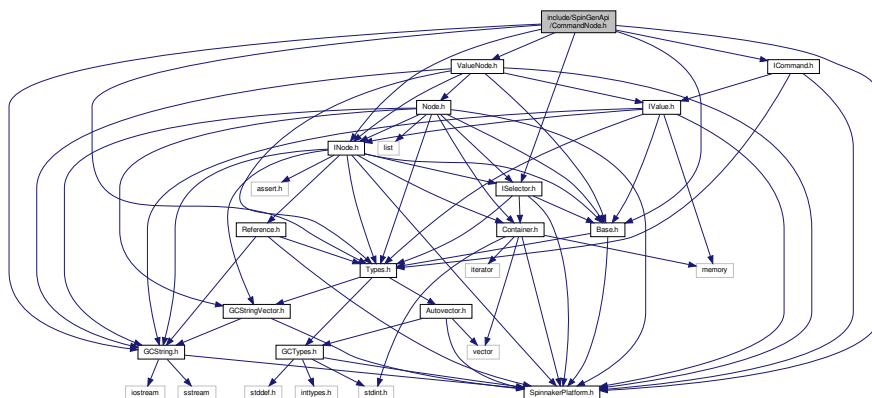
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.

Namespaces

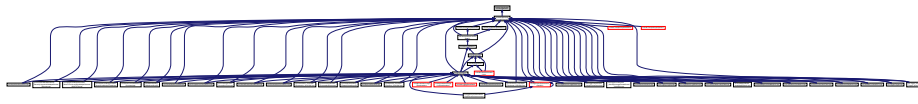
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.74 include/SpinGenApi/CommandNode.h File Reference

Include dependency graph for CommandNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CommandNode](#)
Interface for string properties.

Namespaces

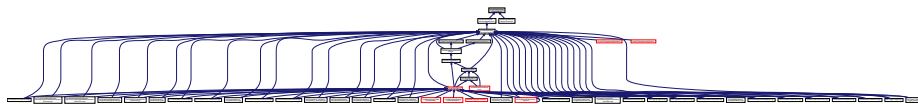
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef [CommandNode](#) [CCommandRef](#)

16.75 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [FMT_I64](#) "ll"

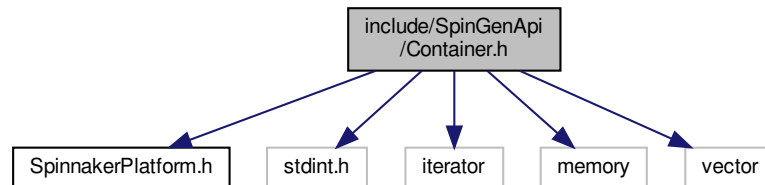
16.75.1 Macro Definition Documentation

16.75.1.1 FMT_I64

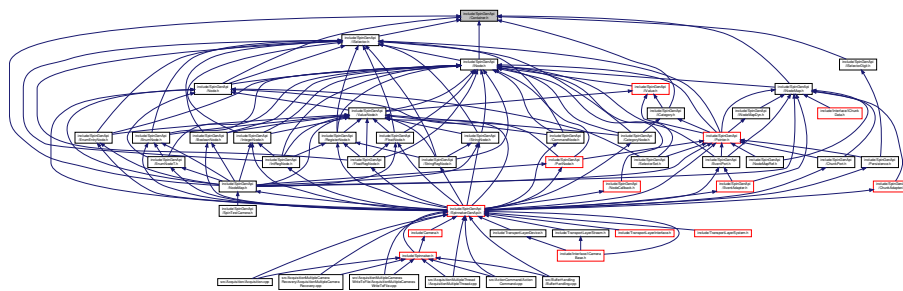
```
#define FMT_I64 "ll"
```

16.76 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:



This graph shows which files directly or indirectly include this file:



16.77 include/SpinGenApi/Counter.h File Reference

Classes

- class [Counter](#)

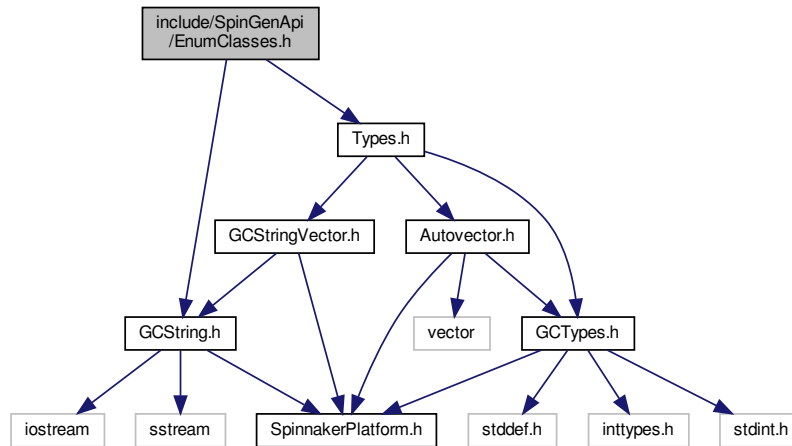
Definition of a simple [Counter](#) class.

Namespaces

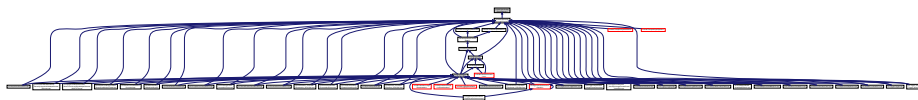
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.78 include/SpinGenApi/EnumClasses.h File Reference

Include dependency graph for EnumClasses.h:



This graph shows which files directly or indirectly include this file:



Classes

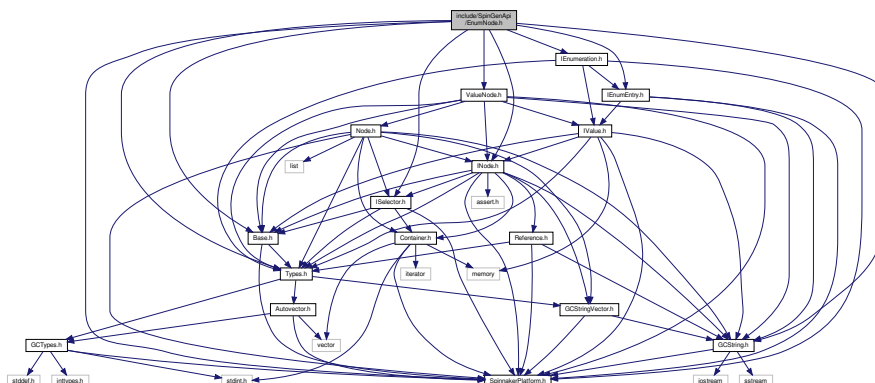
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [EEndiannessClass](#)
Holds conversion methods for the endianness enumeration.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [ESlopeClass](#)

Typedefs

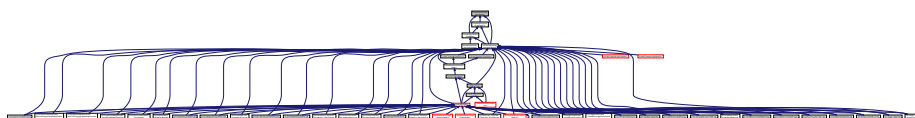
- typedef EnumEntryNode CEnumEntryRef

16.80 include/SpinGenApi/EnumNode.h File Reference

Include dependency graph for EnumNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class EnumNode
 - Interface* for string properties.

Namespaces

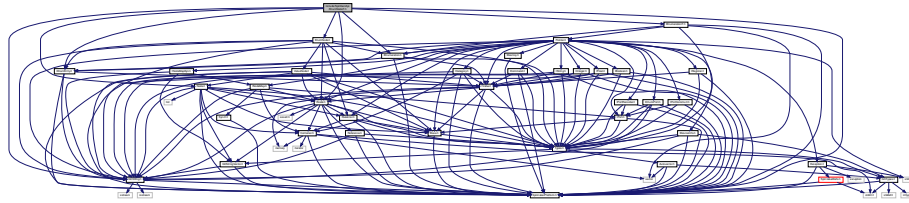
- Spinnaker
- Spinnaker::GenApi

Typedefs

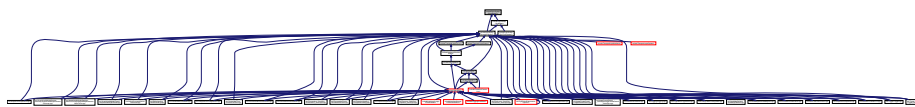
- typedef EnumNode CEnumerationRef

16.81 include/SpinGenApi/EnumNodeT.h File Reference

Include dependency graph for EnumNodeT.h:



This graph shows which files directly or indirectly include this file:



Classes

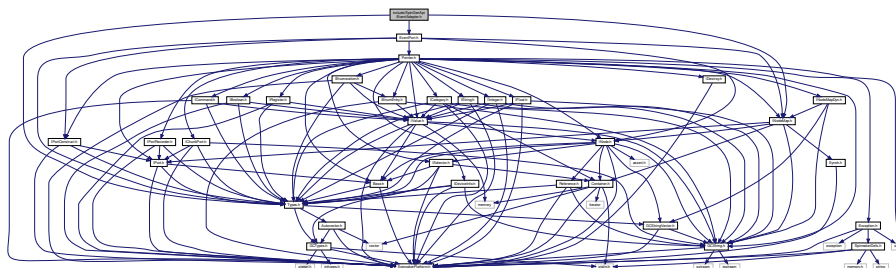
- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.

Namespaces

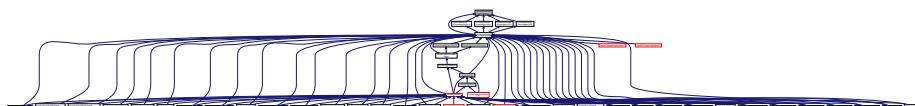
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.82 include/SpinGenApi/EventAdapter.h File Reference

Include dependency graph for EventAdapter.h:

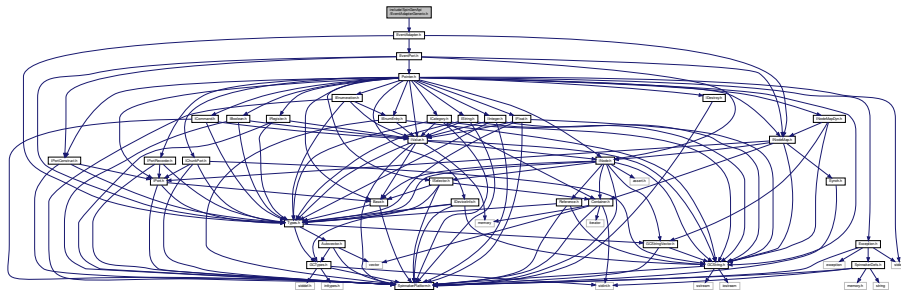


This graph shows which files directly or indirectly include this file:

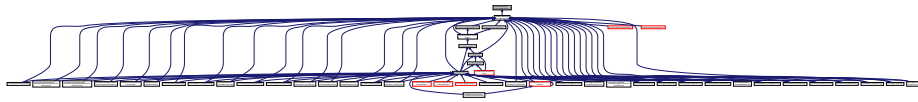


16.84 include/SpinGenApi/EventAdapterGeneric.h File Reference

Include dependency graph for EventAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



Classes

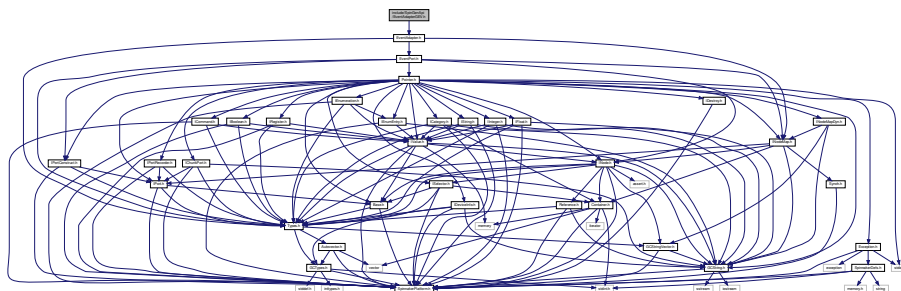
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.

Namespaces

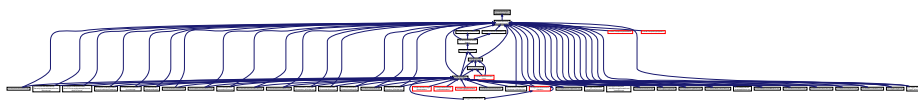
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.85 include/SpinGenApi/EventAdapterGEV.h File Reference

Include dependency graph for EventAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [GVCP_REQUEST_HEADER](#)
header of a GVCP request packet
- struct [GVCP_EVENT_ITEM_BASIC](#)
layout of a GVCP event item (common to all types)
- struct [GVCP_EVENT_ITEM](#)
layout of a GVCP event item (Extended ID flag not set)
- struct [GVCP_EVENT_REQUEST](#)
Layout of a GVCP event request packet (Extended ID flag not set)
- struct [GVCP_EVENTDATA_REQUEST](#)
Layout of a GVCP event data request packet (Extended ID flag not set)
- struct [GVCP_EVENT_ITEM_EXTENDED_ID](#)
layout of a GVCP event item (Extended ID flag set)
- struct [GVCP_EVENT_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event request packet (Extended ID flag set)
- struct [GVCP_EVENTDATA_REQUEST_EXTENDED_ID](#)
Layout of a GVCP event data request packet (Extended ID flag set)
- class [CEventAdapterGEV](#)
Connects a GigE Event to a node map.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

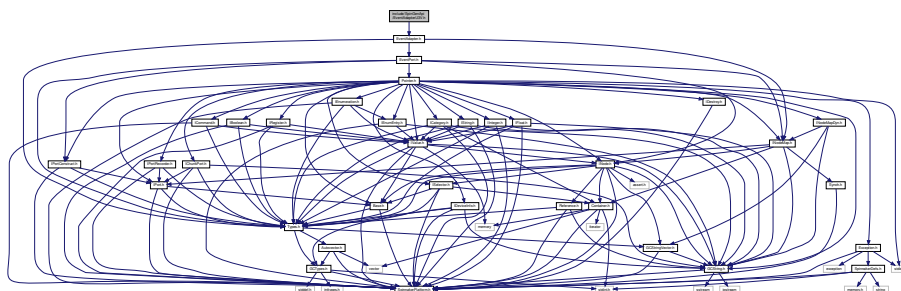
- enum [GVCP_MESSAGE_TAGS](#) {
 [TAG_EVENT_CMD](#) = 0xc0,
 [TAG_EVENTDATA_CMD](#) = 0xc2 }

Variables

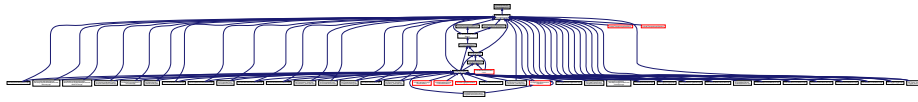
- const uint8_t [COMMAND_MAGIC](#) = 0x42

16.86 include/SpinGenApi/EventAdapterU3V.h File Reference

Include dependency graph for EventAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [U3V_COMMAND_HEADER](#)
U3V/GenCP command header.
- struct [U3V_EVENT_DATA](#)
U3V/GenCP EVENT_CMD specific command data.
- struct [U3V_EVENT_MESSAGE](#)
Entire event data message (without the variable-sized data field)
- class [CEventAdapterU3V](#)
Connects a U3V Event to a node map.

Namespaces

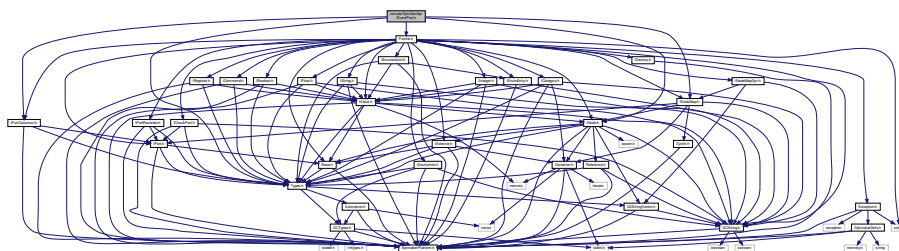
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Variables

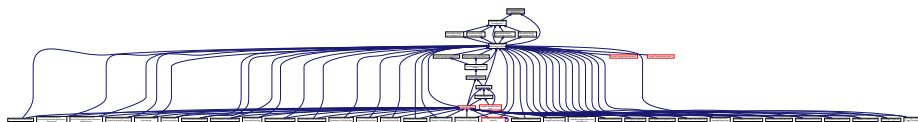
- const uint32_t [U3V_EVENT_PREFIX](#) = 0x45563355
- const uint16_t [GENCP_EVENT_CMD_ID](#) = 0x0C00
- const size_t [GENCP_COMMAND_HEADER_SIZE](#) = sizeof(U3V_COMMAND_HEADER)
- const size_t [GENCP_EVENT_BASIC_SIZE](#) = sizeof(U3V_EVENT_MESSAGE)

16.87 include/SpinGenApi/EventPort.h File Reference

Include dependency graph for EventPort.h:



This graph shows which files directly or indirectly include this file:



Classes

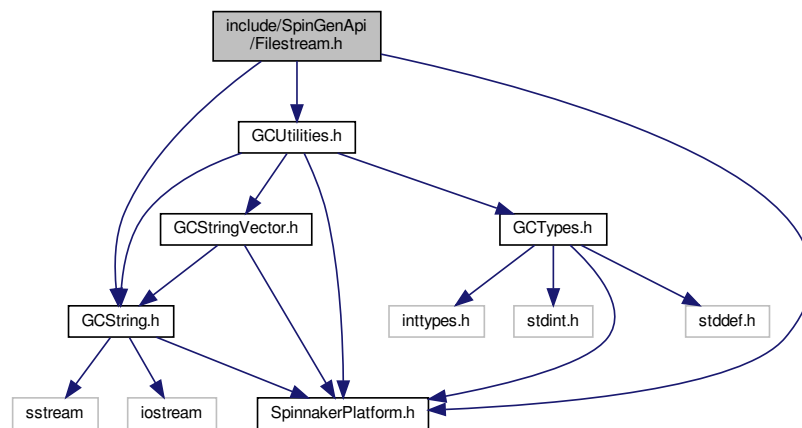
- class [CEventPort](#)
Port attachable to an event.

Namespaces

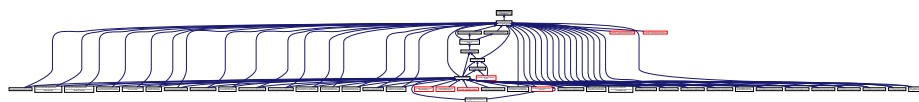
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.88 include/SpinGenApi/Filestream.h File Reference

Include dependency graph for FileStream.h:



This graph shows which files directly or indirectly include this file:

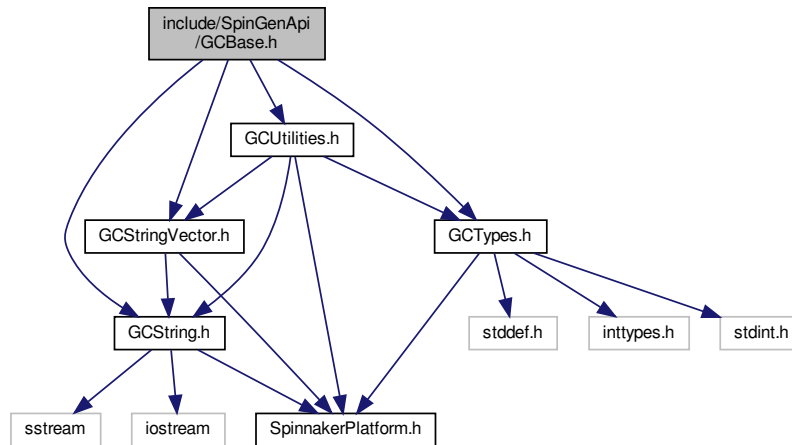


Classes

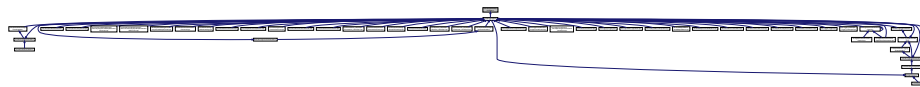
- class [FileProtocolAdapter](#)
Adapter between the `std::iostreambuf` and the SFNC Features representing the device file system.
- class [IDevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBuf< CharType, Traits >](#)
- class [ODevFileStreamBase< CharType, Traits >](#)
- class [IDevFileStreamBase< CharType, Traits >](#)

16.91 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

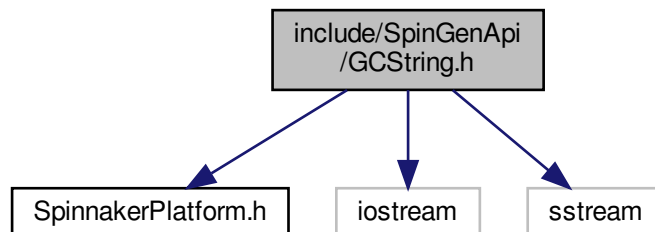


This graph shows which files directly or indirectly include this file:

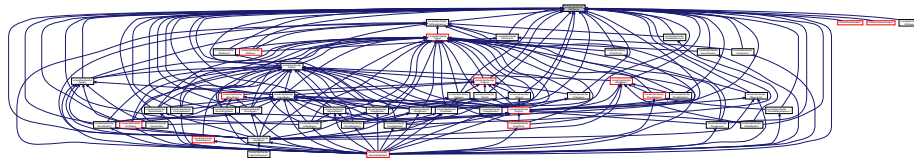


16.92 include/SpinGenApi/GCString.h File Reference

Include dependency graph for GCString.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gcstring](#)

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenlCam](#)

Macros

- `#define GCSTRING\_NPOS size_t(-1)`

Functions

- [SPINNAKER_API](#) void [ThrowBadAlloc](#) ()
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenlCam::gcstring](#) &str)
STL getline.
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenlCam::gcstring](#) &str, char delim)
STL getline.
- std::ostream & [operator<<](#) (std::ostream &ostr, const [Spinnaker::GenlCam::gcstring](#) &str)
STL operator out.
- std::istream & [operator>>](#) (std::istream &istr, [Spinnaker::GenlCam::gcstring](#) &str)
STL operator in.

16.92.1 Macro Definition Documentation

16.92.1.1 GCSTRING_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

16.92.2 Function Documentation

16.92.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & ostr,
    const Spinnaker::GenICam::gcstring & str ) [inline]
```

STL operator out.

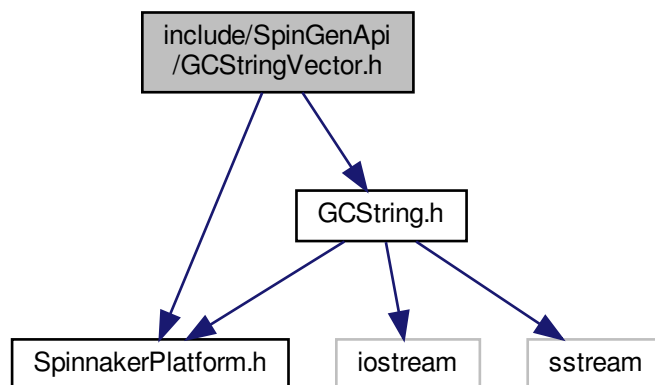
16.92.2.2 operator>>()

```
std::istream& operator>> (
    std::istream & istr,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

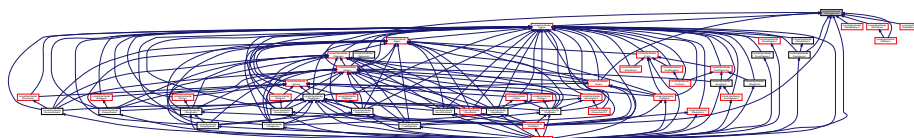
STL operator in.

16.93 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

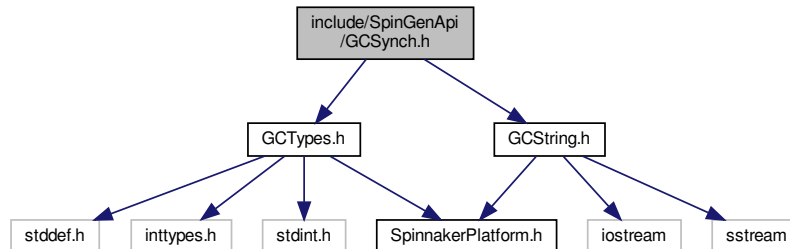


This graph shows which files directly or indirectly include this file:



16.94 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



Classes

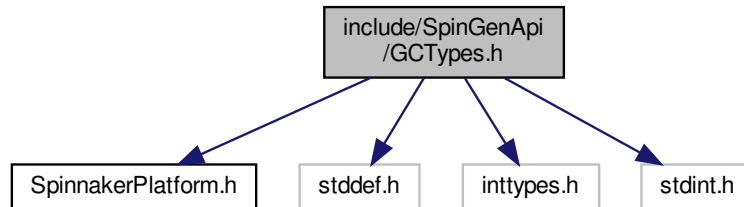
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)
- class [LockableObject< Object >](#)
Instance-Lock for an object.
- class [LockableObject< Object >::Lock](#)
A scopelevel [Lock](#) class.
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.

Namespaces

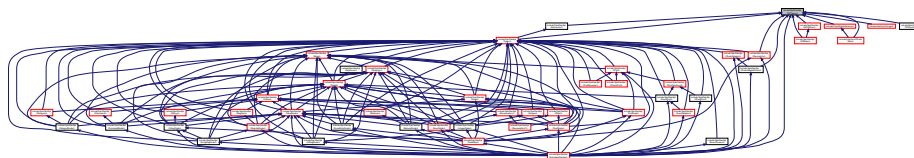
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

16.95 include/SpinGenApi/GCTypes.h File Reference

Include dependency graph for GCTypes.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Version_t](#)
Version.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

Macros

- `#define __STDC_LIMIT_MACROS`
- `#define __STDC_CONSTANT_MACROS`
- `#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */`
- `#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */`
- `#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */`
- `#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */`
- `#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */`
- `#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */`
- `#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */`
- `#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffff80LL) /* minimum signed int8 value */`
- `#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */`

Typedefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

16.95.1 Macro Definition Documentation

16.95.1.1 __STDC_CONSTANT_MACROS

```
#define __STDC_CONSTANT_MACROS
```

16.95.1.2 __STDC_LIMIT_MACROS

```
#define __STDC_LIMIT_MACROS
```

16.95.1.3 GC_INT32_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */
```

16.95.1.4 GC_INT32_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */
```

16.95.1.5 GC_INT64_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */
```

16.95.1.6 GC_INT64_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */
```

16.95.1.7 GC_INT8_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x00000000000007fLL) /* maximum signed int8 value */
```

16.95.1.8 GC_INT8_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */
```

16.95.1.9 GC_UINT32_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */
```

16.95.1.10 GC_UINT64_MAX

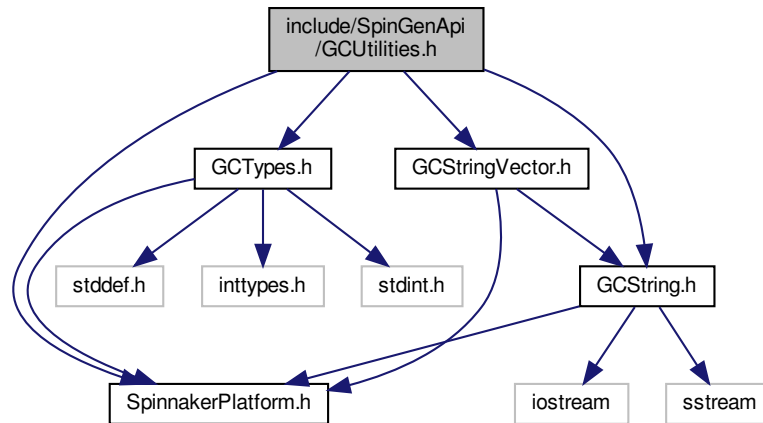
```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */
```

16.95.1.11 GC_UINT8_MAX

```
#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */
```

16.96 include/SpinGenApi/GCUtilities.h File Reference

Include dependency graph for GCUtilities.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

Macros

- `#define USE_TEMP_CACHE_FILE 1`
- `#define USE_TEMP_CACHE_FILE 1`
- `#define GC_COUNTOF(arr) (sizeof(arr) / sizeof(arr)[0])`
- `#define GENICAM_UNUSED(unused_var) ((void)(unused_var))`
- `#define GENICAM_DEPRECATED(FUNCTION) FUNCTION`
- `#define _TO_STRING(__stN) #__stN`
- `#define EXPAND_TO_STRINGISE(__stN) _TO_STRING(__stN)`
- `#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)`
- `#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"`
- `#define __OUTPUT_FORMATER__(__type) __LOCATION__ ": " __type " : "`
- `#define __WARN__ __OUTPUT_FORMATER__("WARNING")`
- `#define __ERR__ __OUTPUT_FORMATER__("ERROR")`
- `#define __TODO__ __OUTPUT_FORMATER__("TBD")`

Functions

- `template<typename Td , typename Ts >`
`Td INTEGRAL_CAST2 (Ts s)`
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- `template<typename T >`
`T INTEGRAL_CAST (int64_t ll)`
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- `SPINNAKER_API bool DoesEnvironmentVariableExist (const Spinnaker::GenICam::gcstring &VariableName)`
Returns true if an environment variable exists.
- `SPINNAKER_API gcstring GetValueOfEnvironmentVariable (const gcstring &VariableName)`
Retrieve the value of an environment variable.
- `SPINNAKER_API bool GetValueOfEnvironmentVariable (const gcstring &VariableName, gcstring &VariableContent)`
Retrieve the value of an environment variable.
- `SPINNAKER_API gcstring UriEncode (const gcstring &Input)`
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- `SPINNAKER_API gcstring UriDecode (const gcstring &Input)`
Replaces xx escapes by their char equivalent.
- `SPINNAKER_API void ReplaceEnvironmentVariables (gcstring &Buffer, bool ReplaceBlankBy20=false)`
Replaces in a string and replace ' ' with %20.
- `SPINNAKER_API gcstring GetGenICamCacheFolder (void)`
Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling SetGenICamCacheFolder().
- `SPINNAKER_API gcstring GetGenICamLogConfig (void)`
Retrieve the path of the GenICam logging properties file.
- `SPINNAKER_API gcstring GetGenICamCLProtocolFolder (void)`
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling SetGenICamCLProtocolFolder().
- `SPINNAKER_API void SetGenICamCacheFolder (const gcstring &path)`
Stores the path of the GenICam cache folder.
- `SPINNAKER_API void SetGenICamLogConfig (const gcstring &path)`
Stores the path of the GenICam logging properties file.
- `SPINNAKER_API void SetGenICamCLProtocolFolder (const gcstring &path)`
Stores the path of the CLProtocol folder.
- `SPINNAKER_API void Tokenize (const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")`
splits str input string into a list of tokens using the delimiter
- `SPINNAKER_API void GetFiles (const gcstring &FileTemplate, gcstring_vector &FileNames, const bool DirectoriesOnly=false)`
Gets a list of files or directories matching a given FileTemplate.
- `SPINNAKER_API gcstring GetModulePathFromFunction (void *pFunction)`
Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

16.96.1 Macro Definition Documentation

16.96.1.1 __ERR__

```
#define __ERR__ __OUTPUT_FORMATER__("ERROR")
```

16.96.1.2 __LINE_STR__

```
#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)
```

16.96.1.3 __LOCATION__

```
#define __LOCATION__ __FILE__ "(" __LINE_STR__ ")"
```

16.96.1.4 __OUTPUT_FORMATER__

```
#define __OUTPUT_FORMATER__(  
    _type ) __LOCATION__ " : " _type " : "
```

16.96.1.5 __TODO__

```
#define __TODO__ __OUTPUT_FORMATER__("TBD")
```

16.96.1.6 __WARN__

```
#define __WARN__ __OUTPUT_FORMATER__("WARNING")
```

16.96.1.7 _TO_STRING

```
#define _TO_STRING(  
    __stN ) #__stN
```

16.96.1.8 EXPAND_TO_STRINGISE

```
#define EXPAND_TO_STRINGISE(  
    __stN ) _TO_STRING(__stN)
```

16.96.1.9 GC_COUNTOF

```
#define GC_COUNTOF(  
    arr ) (sizeof(arr) / sizeof(arr)[0])
```

16.96.1.10 GENICAM_DEPRECATED

```
#define GENICAM_DEPRECATED(  
    FUNCTION ) FUNCTION
```

16.96.1.11 GENICAM_UNUSED

```
#define GENICAM_UNUSED(  
    unused_var ) ((void)(unused_var))
```

16.96.1.12 USE_TEMP_CACHE_FILE [1/2]

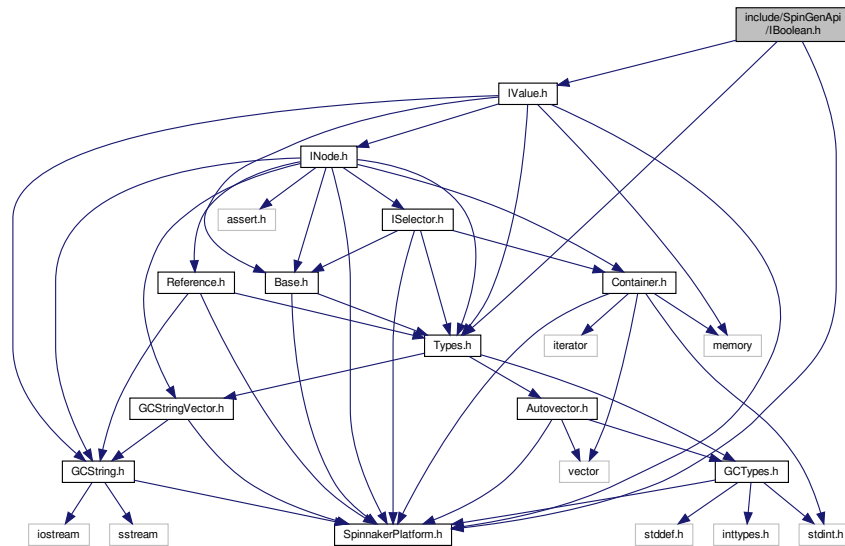
```
#define USE_TEMP_CACHE_FILE 1
```

16.96.1.13 USE_TEMP_CACHE_FILE [2/2]

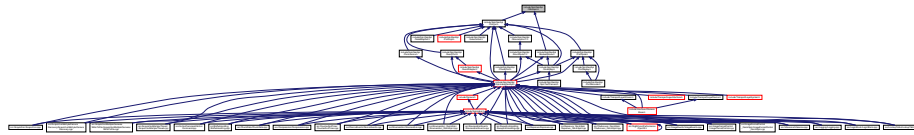
```
#define USE_TEMP_CACHE_FILE 1
```

16.97 include/SpinGenApi/IBoolean.h File Reference

Include dependency graph for IBoolean.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

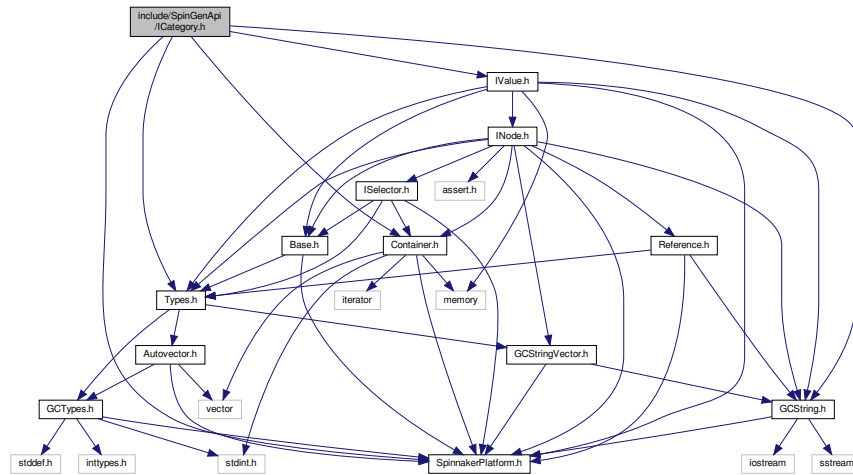
- virtual void [operator=](#) (bool Value)
Set node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.

Variables

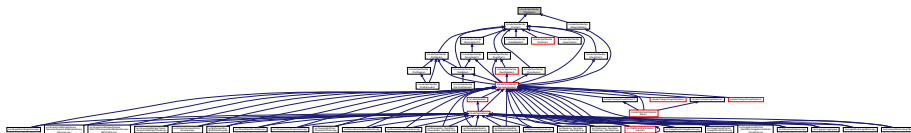
- [interface SPINNAKER_API_ABSTRACT IBoolean](#)
Interface for Boolean properties.
- [interface SPINNAKER_API_ABSTRACT](#) bool [Verify](#) = true) = 0

16.98 include/SpinGenApi/ICategory.h File Reference

Include dependency graph for ICategory.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

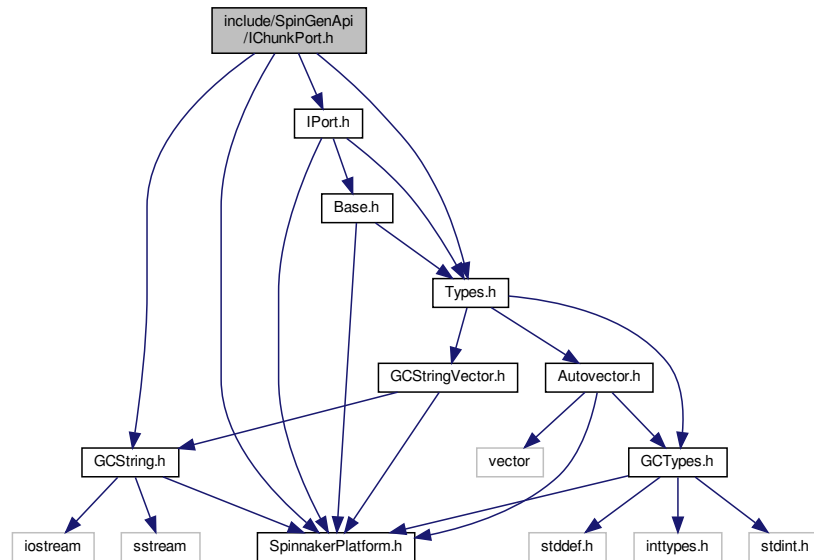
Variables

- [interface SPINNAKER_API_ABSTRACT ICategory](#)

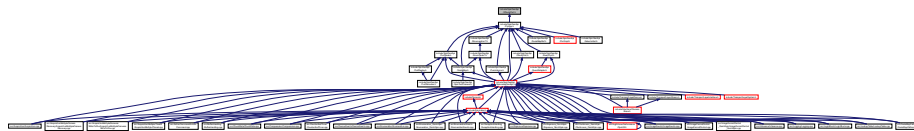
Gives access to a category node.

16.99 include/SpinGenApi/IChunkPort.h File Reference

Include dependency graph for IChunkPort.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Macros

- `#define` [CHUNK_BASE_ADDRESS_REGISTER](#) [GC_INT64_MAX](#)
Address of a `int64_t` pseudo register containing the base address of the chunk (`MAX_INT64`)
- `#define` [CHUNK_BASE_ADDRESS_REGISTER_LEN](#) 8
Length of the `CHUNK_BASE_ADDRESS_REGISTER` pseudo register.
- `#define` [CHUNK_LENGTH_REGISTER](#) ([GC_INT64_MAX](#) - 15)
Address of a `int64_t` pseudo register containing the length of the chunk.
- `#define` [CHUNK_LENGTH_REGISTER_LEN](#) 8
Length of the `CHUNK_LENGTH_REGISTER` pseudo register.

Functions

- virtual EYesNo [CacheChunkData](#) () const =0
Indicates if the chunk a adapter must hold a cached version of the chunk data.

Variables

- [interface SPINNAKER_API_ABSTRACT IChunkPort](#)
Interface for ports attached to a chunk.

16.99.1 Macro Definition Documentation

16.99.1.1 CHUNK_BASE_ADDRESS_REGISTER

```
#define CHUNK_BASE_ADDRESS_REGISTER GC_INT64_MAX
```

Address of a int64_t pseudo register containing the base address of the chunk (MAX_INT64)

16.99.1.2 CHUNK_BASE_ADDRESS_REGISTER_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Lenght of the CHUNK_BASE_ADDRESS_REGISTER pseudo register.

16.99.1.3 CHUNK_LENGTH_REGISTER

```
#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX - 15)
```

Address of a int64_t pseudo register containing the length of the chunk.

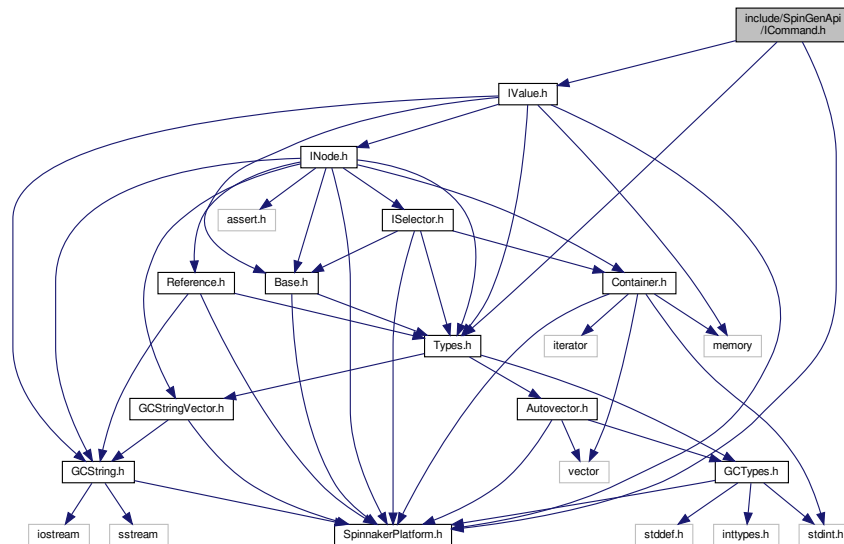
16.99.1.4 CHUNK_LENGTH_REGISTER_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

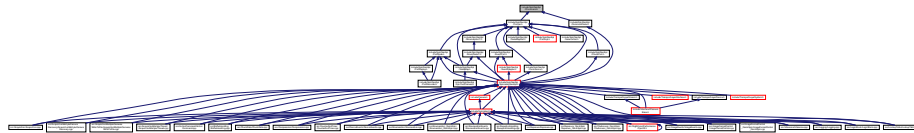
Lenght of the CHUNK_LENGTH_REGISTER pseudo register.

16.100 include/SpinGenApi/ICommand.h File Reference

Include dependency graph for ICommand.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

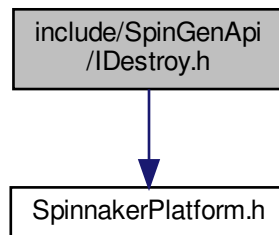
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual bool [IsDone](#) (bool Verify=true)=0
Query whether the command is executed.

Variables

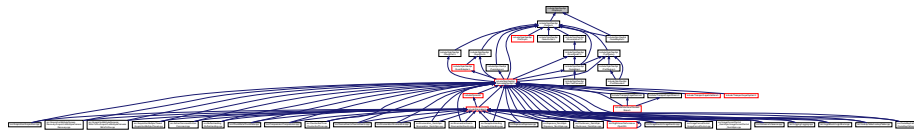
- [interface SPINNAKER_API_ABSTRACT ICommand](#)
Interface for command like properties.

16.101 include/SpinGenApi/IDestroy.h File Reference

Include dependency graph for IDestroy.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

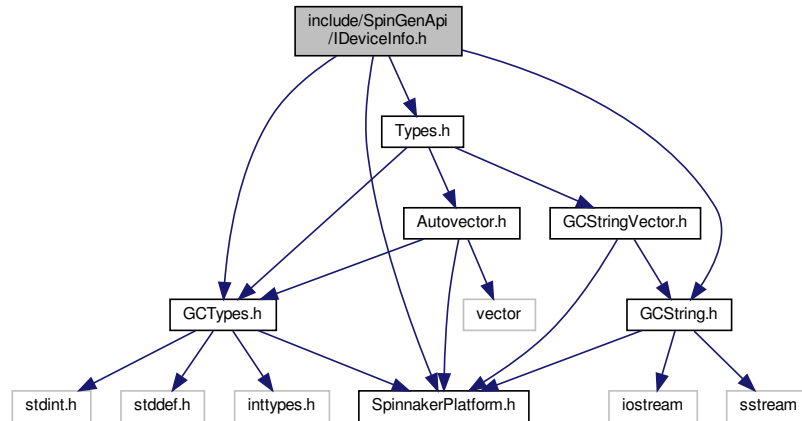
Variables

- [interface SPINNAKER_API_ABSTRACT IDestroy](#)

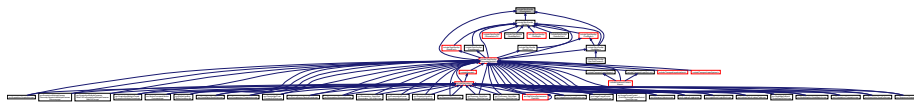
Interface to destroy an object.

16.102 include/SpinGenApi/IDeviceInfo.h File Reference

Include dependency graph for IDeviceInfo.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

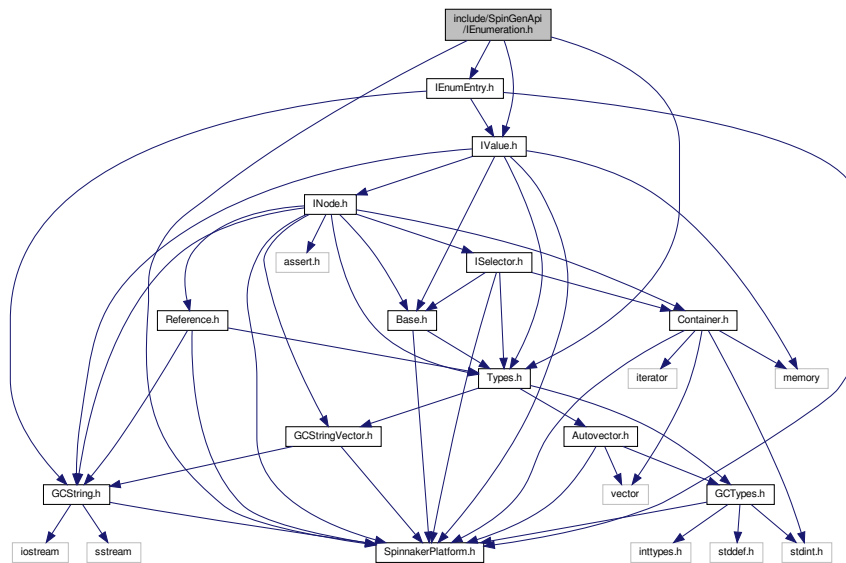
- virtual `GenICam::gcstring` [GetVendorName](#) ()=0
Get the vendor name.
- virtual `GenICam::gcstring` [GetToolTip](#) ()=0
Get tool tip.
- virtual `GenICam::gcstring` [GetStandardNameSpace](#) ()=0
Get the standard name space.
- virtual void [GetGenApiVersion](#) (GenICam::Version_t &Version, uint16_t &Build)=0
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) (GenICam::Version_t &Version)=0
Get the schema version number.
- virtual void [GetDeviceVersion](#) (GenICam::Version_t &Version)=0
Get the version of the device description file.
- virtual `GenICam::gcstring` [GetProductGuid](#) ()=0
Get the Guid describing the product.
- virtual `GenICam::gcstring` [GetVersionGuid](#) ()=0
Get the Guid describing the product version.

Variables

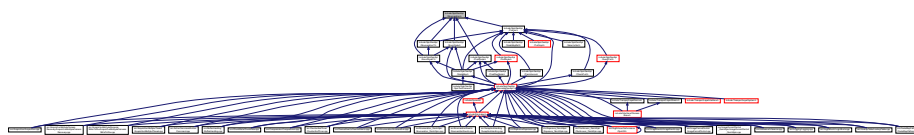
- [interface SPINNAKER_API_ABSTRACT IEnumEntry](#)
Interface of single enum value.

16.104 include/SpinGenApi/IEnumeration.h File Reference

Include dependency graph for IEnumeration.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

- virtual void [GetEntries](#) (NodeList_t &Entries)=0
Get list of entry nodes.
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool Verify=true)=0

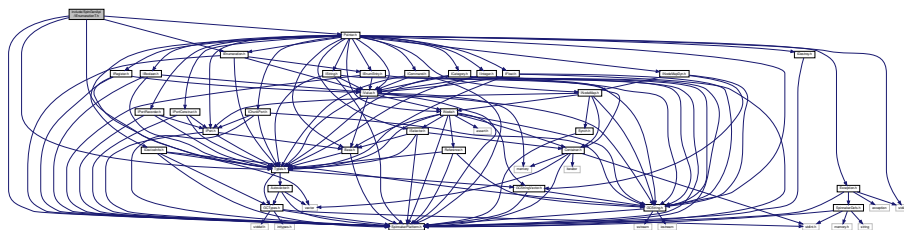
- Set integer node value.*
- virtual GenICam::gcstring [operator*](#) ()=0
- Get string node value.*
- virtual int64_t [GetIntValue](#) (bool Verify=false, bool IgnoreCache=false)=0
- Get integer node value.*
- virtual IEnumEntry * [GetEntryByName](#) (const GenICam::gcstring &Symbolic)=0
- Get an entry node by name.*
- virtual IEnumEntry * [GetEntry](#) (const int64_t IntValue)=0
- Get an entry node by its IntValue.*
- virtual IEnumEntry * [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0
- Get the current entry.*

Variables

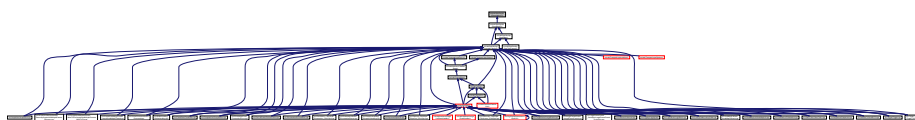
- [interface SPINNAKER_API_ABSTRACT IEnumeration](#)
Interface for enumeration properties.

16.105 include/SpinGenApi/IEnumerationT.h File Reference

Include dependency graph for IEnumerationT.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

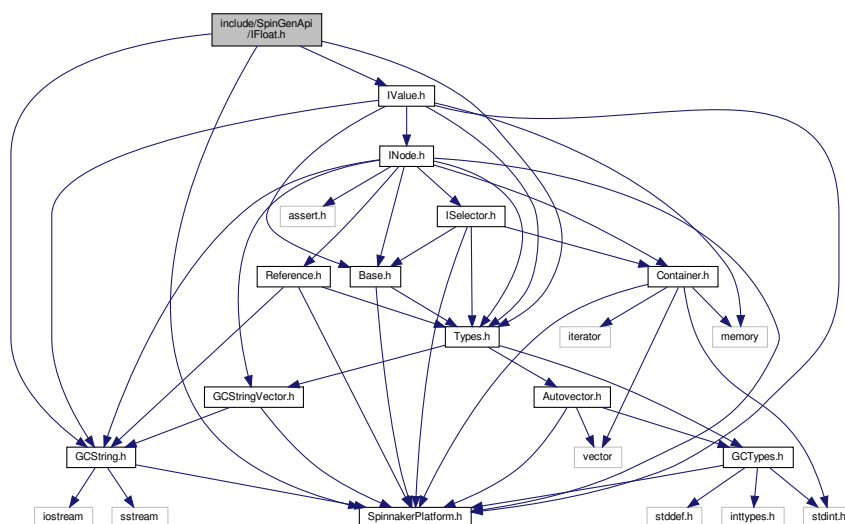
- virtual IEnumeration & **operator=** (EnumT Value)=0
Set node value.
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool **operator()** () const
Get node value.
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual IEnumEntry * **GetEntry** (const int64_t IntValue)=0
Get an entry node by its IntValue.
- virtual IEnumEntry * **GetEntry** (const EnumT Value)=0
returns the EnumEntry object belonging to the Value
- virtual IEnumEntry * **GetCurrentEntry** (bool Verify=false, bool IgnoreCache=false)=0
Get the current entry.

Variables

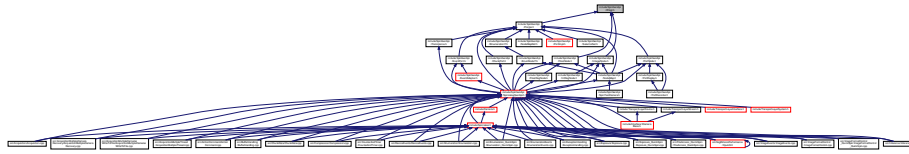
- template<typename EnumT >
interface SPINNAKER_API_ABSTRACT IEnumerationT
Interface for enumeration properties.
- interface SPINNAKER_API_ABSTRACT** virtual public **IEnumReference**
Interface to construct an enum reference.

16.106 include/SpinGenApi/IFloat.h File Reference

Include dependency graph for IFloat.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

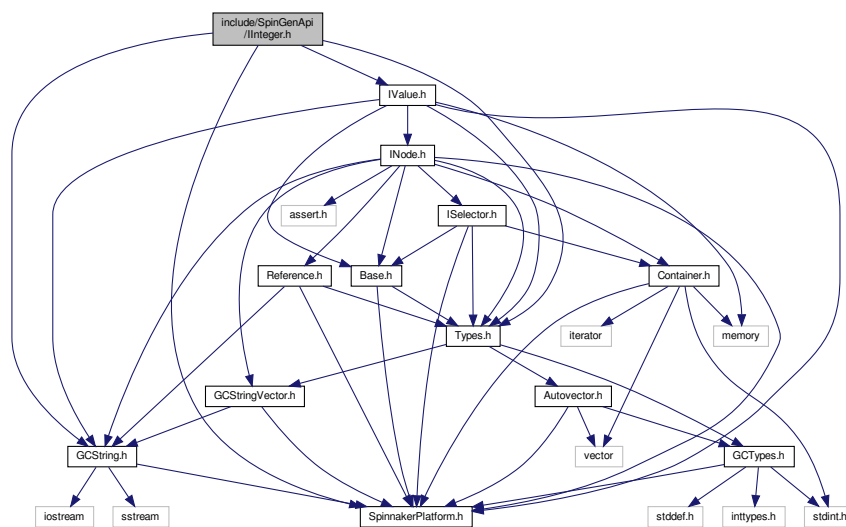
- virtual IFloat & [operator=](#) (double Value)=0
Set node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual GenICam::gcstring [operator*](#) ()=0
Get string node value.
- virtual double [GetMin](#) ()=0
Get minimum value allowed.
- virtual double [GetMax](#) ()=0
Get maximum value allowed.
- virtual bool [HasInc](#) ()=0
True if the float has a constant increment.
- virtual EIncMode [GetIncMode](#) ()=0
Get increment mode.
- virtual double [GetInc](#) ()=0
Get the constant increment if there is any.
- virtual double*_autovector_t [GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual ERepresentation [GetRepresentation](#) ()=0
Get recommended representation.
- virtual GenICam::gcstring [GetUnit](#) () const =0
Get the physical unit name.
- virtual EDisplayNotation [GetDisplayNotation](#) () const =0
Get the way the float should be converted to a string.
- virtual int64_t [GetDisplayPrecision](#) () const =0
Get the precision to be used when converting the float to a string.
- virtual void [ImposeMin](#) (double Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)=0
Restrict maximum value.

Variables

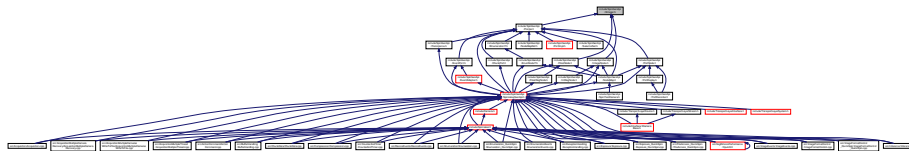
- [interface SPINNAKER_API_ABSTRACT IFloat](#)
Interface for float properties.

16.107 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinaker](#)
- [Spinaker::GenApi](#)

Functions

- virtual `IInteger` & [operator=](#) (int64_t Value)=0
Set node value.
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const

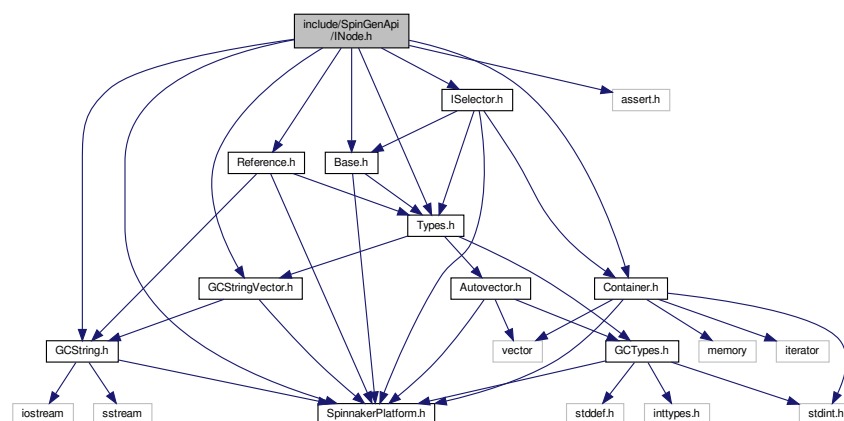
- Get node value.*
- virtual GenICam::gcstring **operator*** ()=0
- Get string node value.*
- virtual double **GetMin** ()=0
- Get minimum value allowed.*
- virtual double **GetMax** ()=0
- Get maximum value allowed.*
- virtual EIncMode **GetIncMode** ()=0
- Get increment mode.*
- virtual double **GetInc** ()=0
- Get the constant increment if there is any.*
- virtual double_autovector_t **GetListOfValidValues** (bool bounded=true)=0
- Get list of valid value.*
- virtual ERepresentation **GetRepresentation** ()=0
- Get recommended representation.*
- virtual GenICam::gcstring **GetUnit** () const =0
- Get the physical unit name.*
- virtual void **ImposeMin** (int64_t Value)=0
- Restrict minimum value.*
- virtual void **ImposeMax** (int64_t Value)=0
- Restrict maximum value.*

Variables

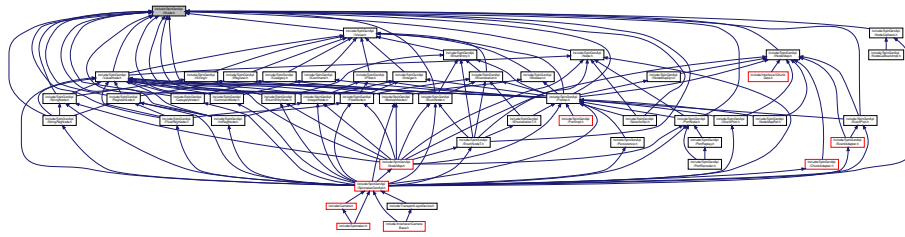
- **interface SPINNAKER_API_ABSTRACT Integer**
Interface for integer properties.

16.108 include/SpinGenApi/INode.h File Reference

Include dependency graph for INode.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef node_vector [NodeList_t](#)
a list of node references
- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes

Functions

- virtual GenApi::ENamespace [GetNameSpace](#) () const =0
Get name space.
- virtual EVisibility [GetVisibility](#) () const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()=0
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const =0
Is the node value cacheable.
- virtual EYesNo [IsAccessModeCacheable](#) () const =0
True if the AccessMode can be cached.
- virtual ECachingMode [GetCachingMode](#) () const =0
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const =0
recommended polling time (for non-cacheable nodes)
- virtual GenICam::gcstring [GetToolTip](#) ()=0
Get tool tip.
- virtual GenICam::gcstring [GetDescription](#) () const =0
Get a long description of the node.
- virtual GenICam::gcstring [GetDisplayName](#) () const =0
Get a name string for display.
- virtual GenICam::gcstring [GetDeviceName](#) () const =0
Get a name of the device.
- virtual void [GetChildren](#) (GenApi::NodeList_t &Children, ELinkType LinkType=ctReadingChildren) const =0
Get all nodes this node directly depends on.

- virtual void [GetParents](#) (GenApi::NodeList_t &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual CallbackHandleType [RegisterCallback](#) (CNodeCallback *pCallback)=0
Register change callback Takes ownership of the [CNodeCallback](#) object.
- virtual bool [DeregisterCallback](#) (CallbackHandleType hCallback)=0
De register change callback Destroys [CNodeCallback](#) object.
- virtual INodeMap * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual GenICam::gcstring [GetEventID](#) () const =0
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const =0
True if the node is streamable.
- virtual void [GetPropertyNames](#) (GenICam::gcstring_vector &PropertyNames) const =0
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)=0
Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)=0
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)=0
Imposes a visibility to the natural visibility of the node.
- virtual INode * [GetAlias](#) () const =0
Retrieves the a node which describes the same feature in a different way.
- virtual INode * [GetCastAlias](#) () const =0
Retrieves the a node which describes the same feature so that it can be casted.
- virtual GenICam::gcstring [GetDocuURL](#) () const =0
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const =0
True if the node should not be used any more.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const =0
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const =0
True if the node can be reached via category nodes from a category node named "Root".
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) (EAccessMode AccessMode)
Tests if readable.
- bool [IsReadable](#) (const IBase *p)
Checks if a node is readable.
- bool [IsReadable](#) (const IBase &r)
Checks if a node is readable.
- bool [IsWritable](#) (EAccessMode AccessMode)
Tests if writable.
- bool [IsWritable](#) (const IBase *p)
Checks if a node is writable.
- bool [IsWritable](#) (const IBase &r)
Checks if a node is writable.
- bool [IsImplemented](#) (EAccessMode AccessMode)
Tests if implemented.
- bool [IsImplemented](#) (const IBase *p)

- Checks if a node is implemented.*

 - bool [IsImplemented](#) (const IBase &r)
- Checks if a node is implemented.*

 - bool [IsAvailable](#) (EAccessMode AccessMode)
- Tests if available.*

 - bool [IsAvailable](#) (const IBase *p)
- Checks if a node is available.*

 - bool [IsAvailable](#) (const IBase &r)
- Checks if a node is available.*

 - EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)
- Computes which access mode the two guards allow together.*

 - bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)
- Tests Visibility CAVE : this relies on the EVisibility enum's coding.*

 - EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)
- Computes which visibility the two guards allow together.*

 - bool [IsCacheable](#) (ECachingMode CachingMode)
- Tests Cacheability.*

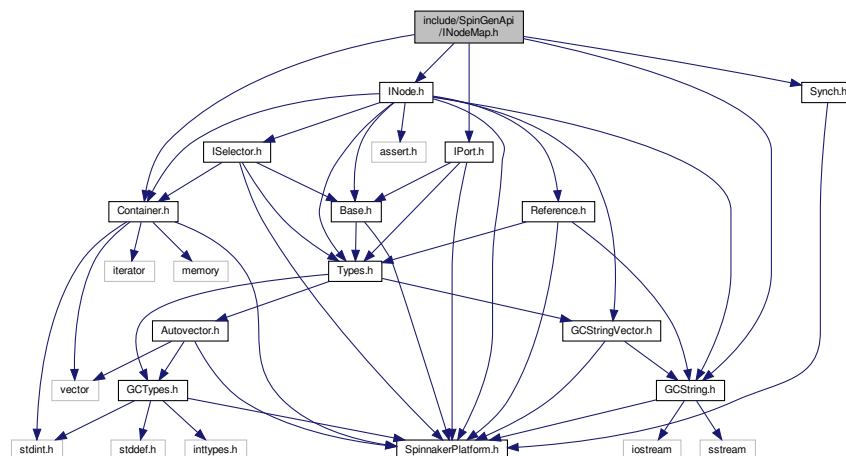
 - ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)
- Computes which CachingMode results from a combination.*

Variables

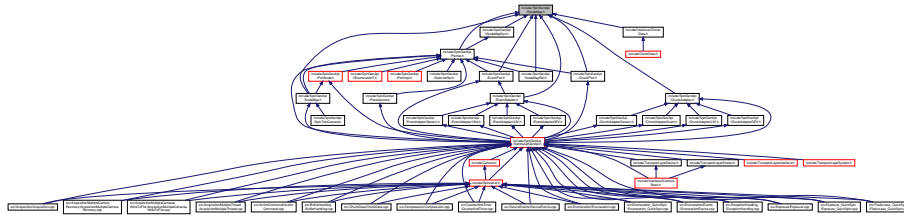
- [interface SPINNAKER_API_ABSTRACT INode](#)
Interface common to all nodes.
- [interface SPINNAKER_API_ABSTRACT](#) virtual public [IReference](#)
Interface to construct a reference.

16.109 include/SpinGenApi/INodeMap.h File Reference

Include dependency graph for INodeMap.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

- virtual `INode *` [GetNode](#) (`const GenICam::gcstring &Name`) `const =0`
Retrieves the node from the central map by Name.
- virtual `void` [InvalidateNodes](#) () `const =0`
Invalidates all nodes.
- virtual `bool` [Connect](#) (`IPort *pPort, const GenICam::gcstring &PortName`) `const =0`
Connects a port to a port node with given name.
- virtual `bool` [Connect](#) (`IPort *pPort`) `const =0`
Connects a port to the standard port "Device".
- virtual `GenICam::gcstring` [GetDeviceName](#) () `const =0`
Get a name of the device.
- virtual `void` [Poll](#) (`int64_t ElapsedTime`)`=0`
Fires nodes which have a polling time.
- virtual `CLock &` [GetLock](#) () `const =0`
Returns the lock which guards the node map.
- virtual `uint64_t` [GetNumNodes](#) () `const =0`
Get the number of nodes in the map.

Variables

- [interface SPINNAKER_API_ABSTRACT INodeMap](#)
Interface to access the node map.

Classes

- class `IntegerNode`
Interface for string properties.

Namespaces

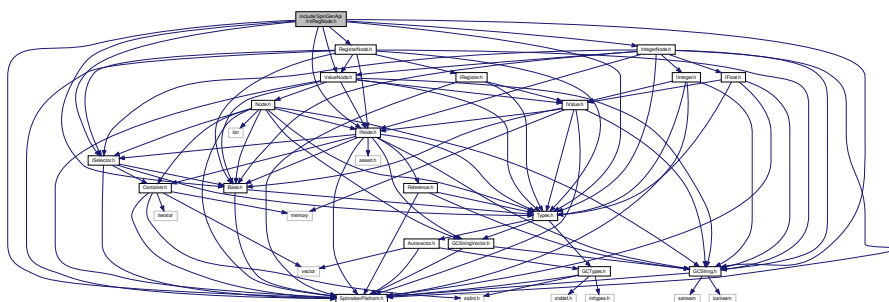
- Spinnaker
- Spinnaker::GenApi

Typedefs

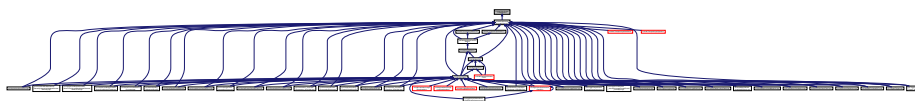
- typedef IntegerNode CIntegerRef

16.112 include/SpinGenApi/IntRegNode.h File Reference

Include dependency graph for IntRegNode.h:



This graph shows which files directly or indirectly include this file:



Classes

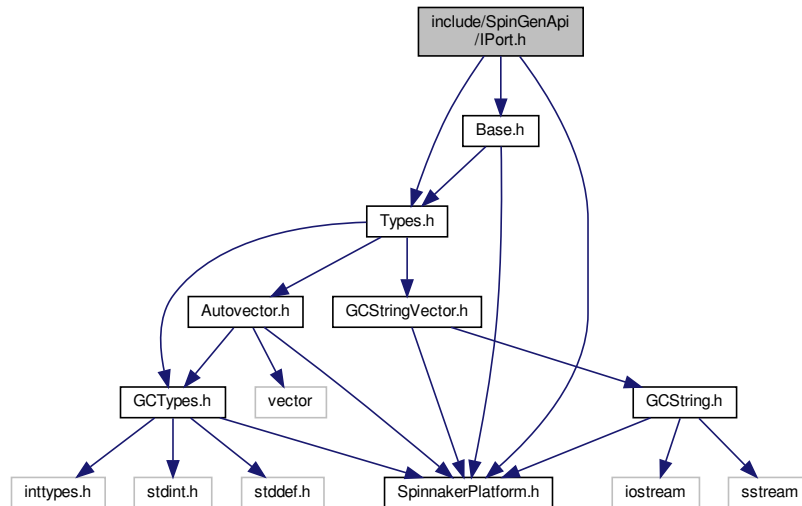
- class `IntRegNode`
Interface for string properties.

Namespaces

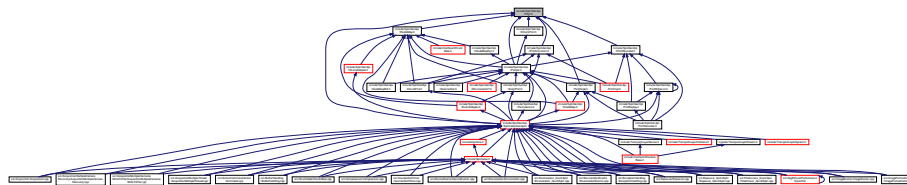
- Spinnaker
- Spinnaker::GenApi

16.113 include/SpinGenApi/IPort.h File Reference

Include dependency graph for IPort.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

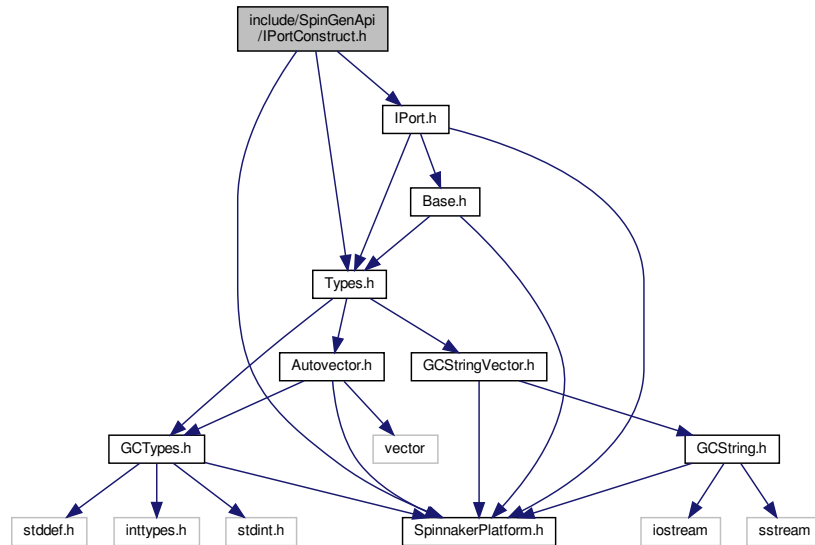
- virtual void [Write](#) (const void *pBuffer, int64_t Address, int64_t Length)=0
Writes a chunk of bytes to the port.

Variables

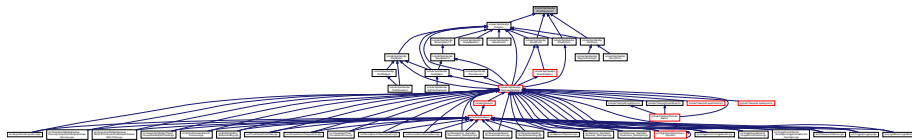
- [interface SPINNAKER_API_ABSTRACT IPort](#)
Interface for ports.
- [interface SPINNAKER_API_ABSTRACT int64_t Address](#)
- [interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0](#)

16.114 include/SpinGenApi/IPortConstruct.h File Reference

Include dependency graph for IPortConstruct.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

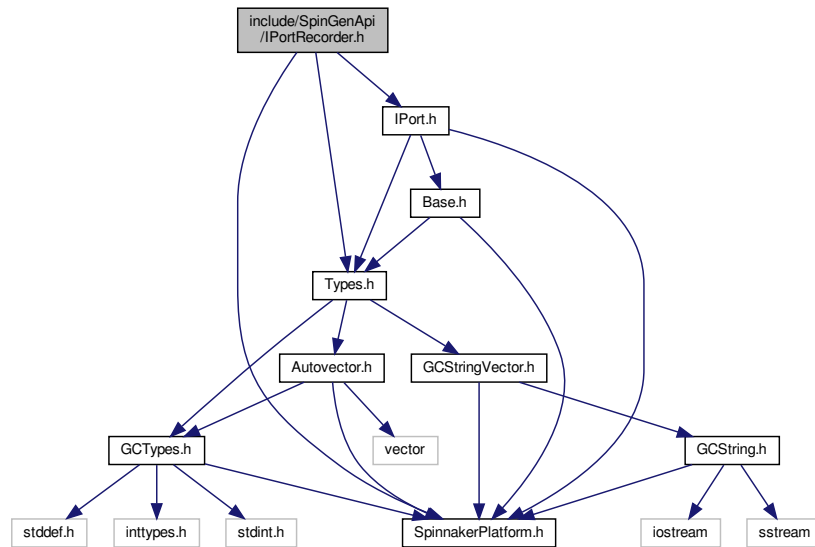
- virtual EYesNo [GetSwapEndianness](#) ()=0
Determines if the port adapter must perform an endianness swap.

Variables

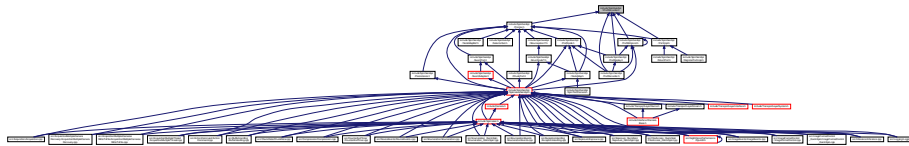
- [interface SPINNAKER_API IPortConstruct](#)
Interface for ports.

16.115 include/SpinGenApi/IPortRecorder.h File Reference

Include dependency graph for IPortRecorder.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

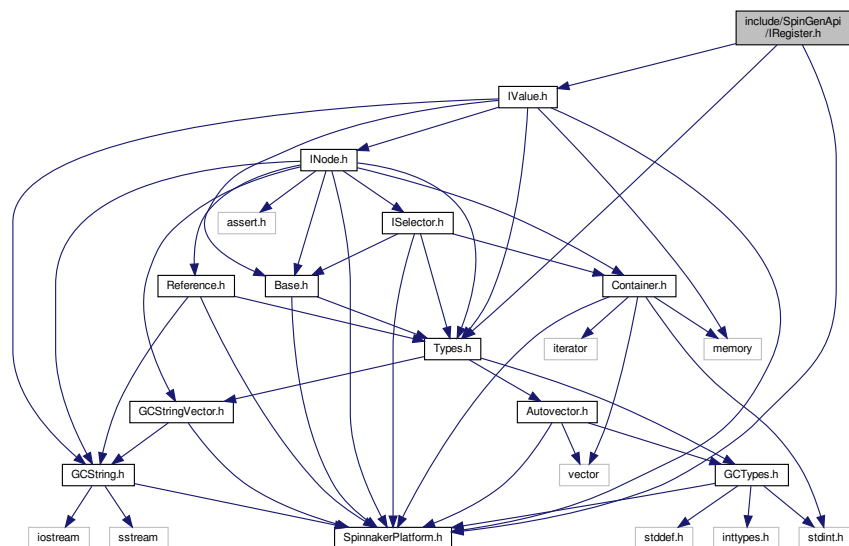
- virtual void [Replay](#) (IPort *pPort)=0
Replays the write command to the given port interface.
- virtual void [SetCookie](#) (const int64_t Value)=0
Sets a cookie in case the port implementation want to cache a command list.
- virtual int64_t [GetCookie](#) ()=0
Gets the cookie a port implementation may have set for caching a command list.
- virtual void [StopRecording](#) ()=0
Stops recording.

Variables

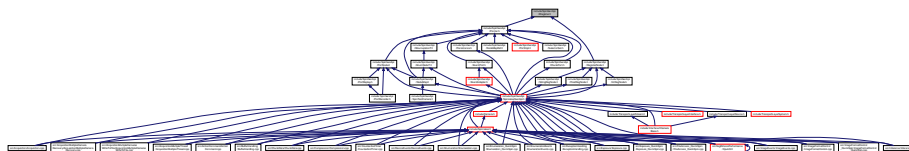
- [interface SPINNAKER_API_ABSTRACT IPortWriteList](#)
- [interface SPINNAKER_API_ABSTRACT IPortReplay](#)
Interface for replaying write commands on a port.
- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)
Interface for recording write commands on a port.

16.116 include/SpinGenApi/IRegister.h File Reference

Include dependency graph for IRegister.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

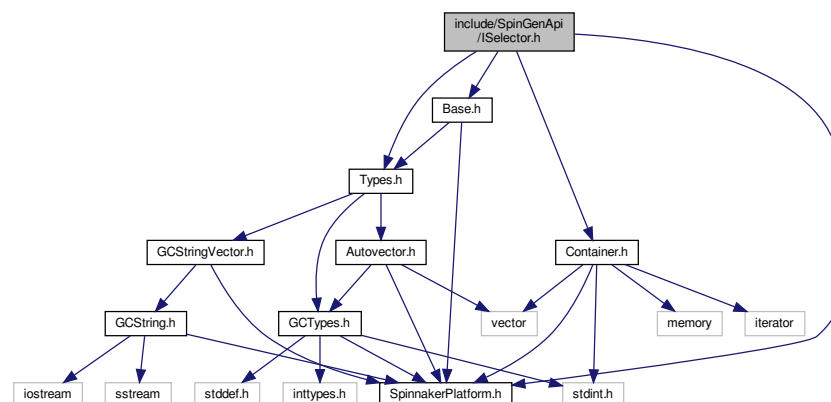
- virtual void [Get](#) (uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)=0
Fills a buffer with the register's contents.
- virtual int64_t [GetLength](#) ()=0
Retrieves the Length of the register [Bytes].
- virtual int64_t [GetAddress](#) ()=0
Retrieves the Address of the register.

Variables

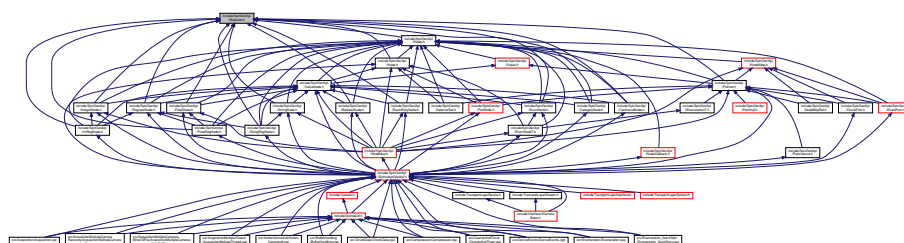
- [interface SPINNAKER_API_ABSTRACT IRegister](#)
Interface for registers.

16.117 include/SpinGenApi/ISelector.h File Reference

Include dependency graph for ISelector.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

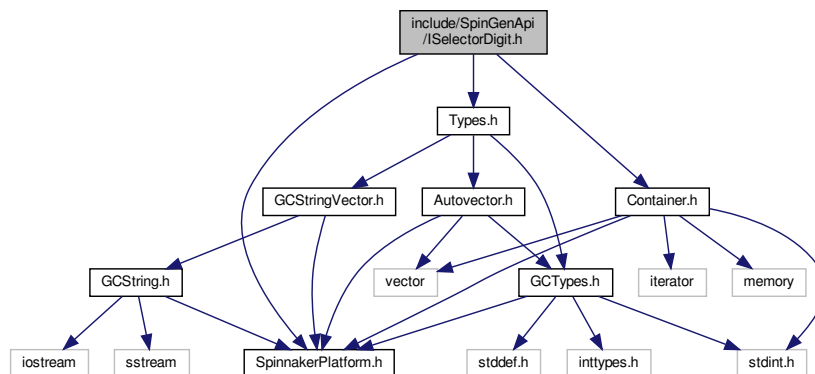
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node

Variables

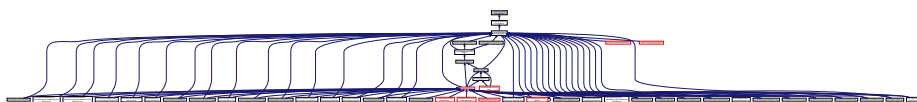
- [interface SPINNAKER_API_ABSTRACT ISelector](#)
Interface for groups of features selected by a single one.

16.118 include/SpinGenApi/ISelectorDigit.h File Reference

Include dependency graph for ISelectorDigit.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

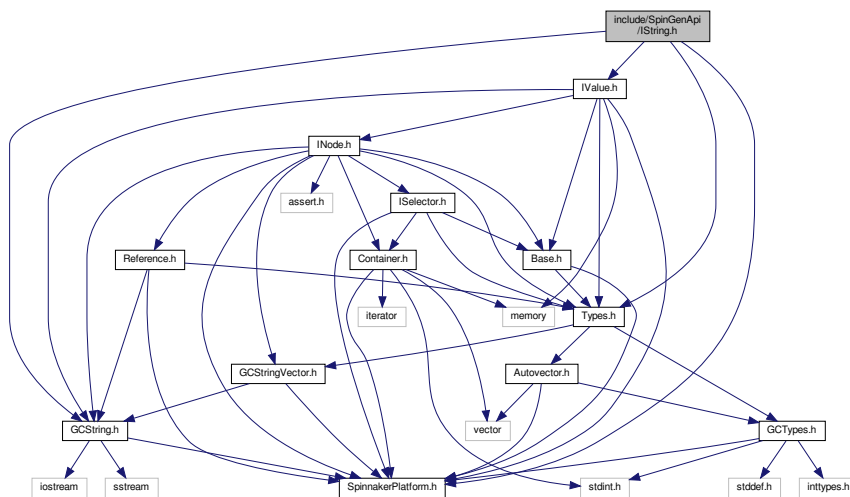
- virtual bool [SetNext](#) (bool Tick=true)=0
Sets digit to next value.
- virtual void [Restore](#) ()=0
Restores the selectors' values found at creation.
- virtual GenICam::gcstring [ToString](#) ()=0
Returns a string representation of the digit.
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)=0
Retrieves an ordered list of selectors.

Variables

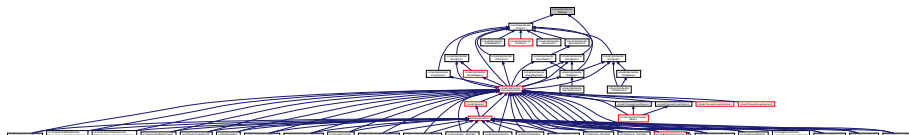
- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)
Interface of a "digit" of the "counter" formed by the selector set.

16.119 include/SpinGenApi/IString.h File Reference

Include dependency graph for IString.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinaker](#)
- [Spinaker::GenApi](#)

Functions

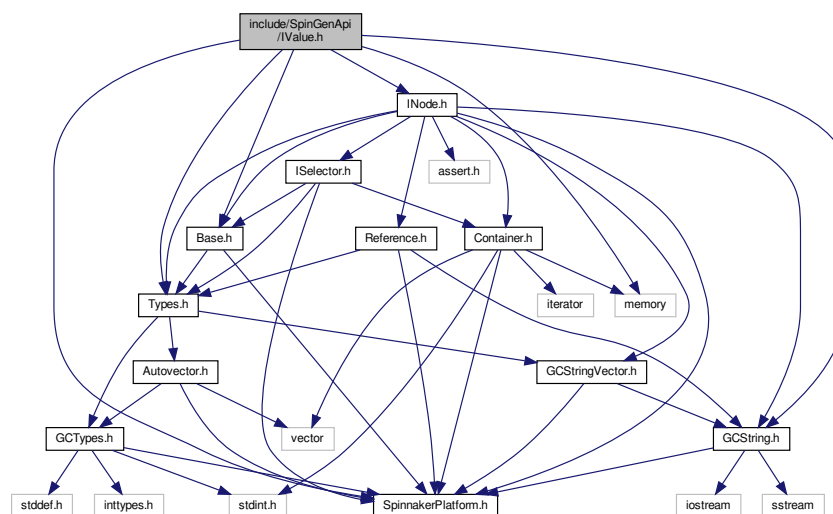
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool **operator()** () const
Get node value.
- virtual GenICam::gcstring **operator*** ()=0
Get string node value.
- virtual int64_t **GetMaxLength** ()=0
Retrieves the maximum length of the string in bytes.

Variables

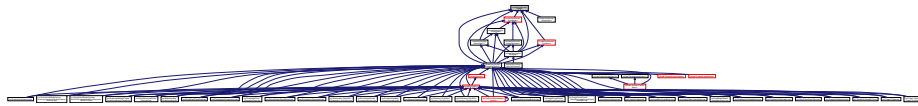
- interface SPINNAKER_API_ABSTRACT IString**
Interface for string properties.

16.120 include/SpinGenApi/IValue.h File Reference

Include dependency graph for IValue.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Node](#)
class common to all nodes

Namespaces

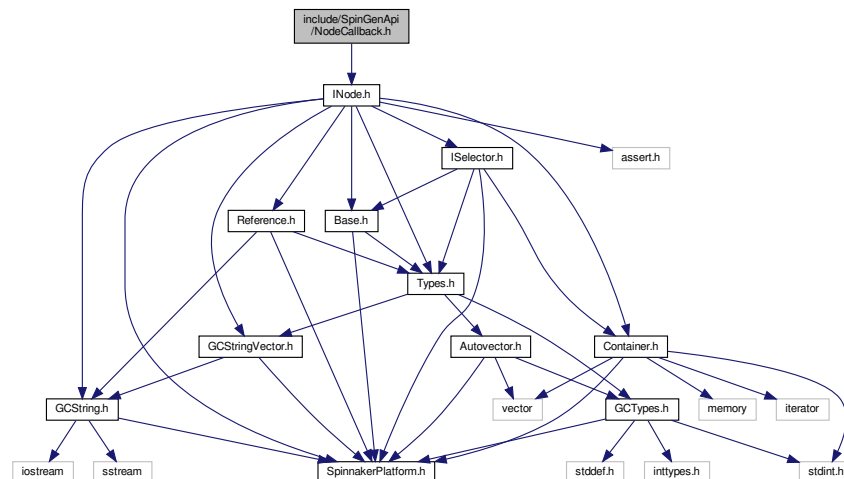
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

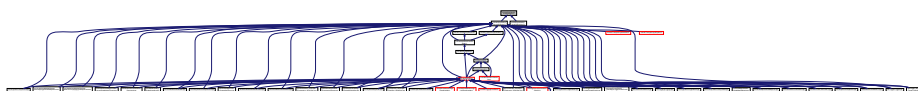
- typedef Node [CNodeRef](#)
- typedef Node [CSelectorRef](#)

16.122 include/SpinGenApi/NodeCallback.h File Reference

Include dependency graph for NodeCallback.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CNodeCallback](#)
callback body instance for INode pointers
- class [Function_NodeCallback](#)< [Function](#) >
Container for a function pointer.
- class [Member_NodeCallback](#)< [Client](#), [Member](#) >
Container for a member function pointer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

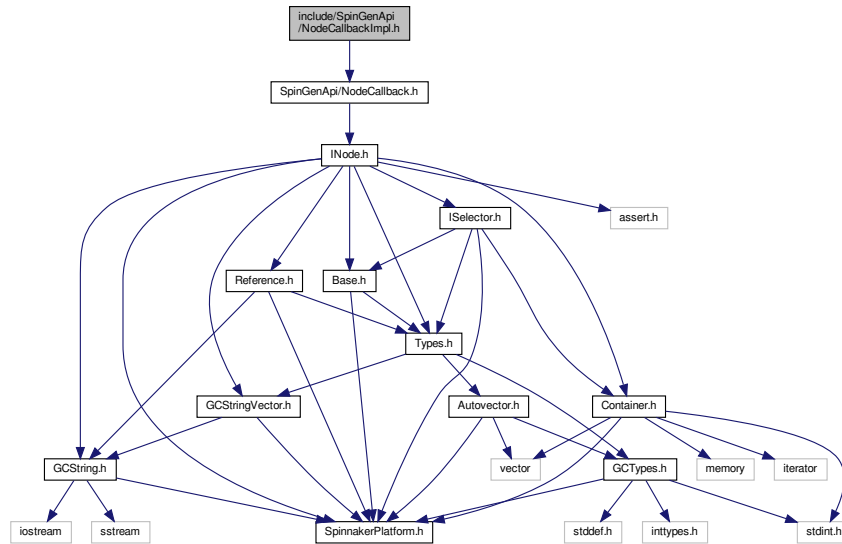
- enum [_ECallbackType](#) {
 [cbPostInsideLock](#) = 1,
 [cbPostOutsideLock](#) = 2 }
the type of callback

Functions

- template<class [Function](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Function](#) function, [ECallbackType](#) CallbackType)
make a new callback object for C functions
- template<class [Function](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Function](#) f, [ECallbackType](#) CallbackType=cbPostInsideLock)
Register a C-function as a callback.
- template<class [Client](#) , class [Member](#) >
 [CNodeCallback](#) * [make_NodeCallback](#) ([INode](#) *pNode, [Client](#) &client, [Member](#) member, [ECallbackType](#) CallbackType)
make a new callback object for member functions
- template<class [Client](#) , class [Member](#) >
 intptr_t [Register](#) ([INode](#) *pNode, [Client](#) &c, [Member](#) m, [ECallbackType](#) CallbackType=cbPostInsideLock)
Register a C++-member function a callback.
- [SPINNAKER_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)
Unregistering callback by handle.

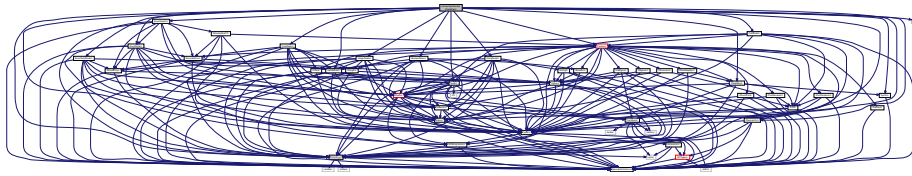
16.123 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

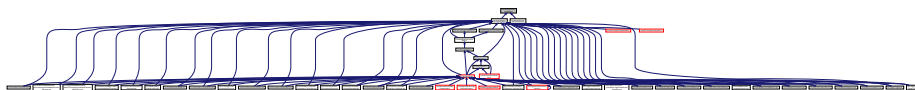


16.124 include/SpinGenApi/NodeMap.h File Reference

Include dependency graph for NodeMap.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [NodeMap](#)
Smart pointer template for NodeMaps with create function.

Namespaces

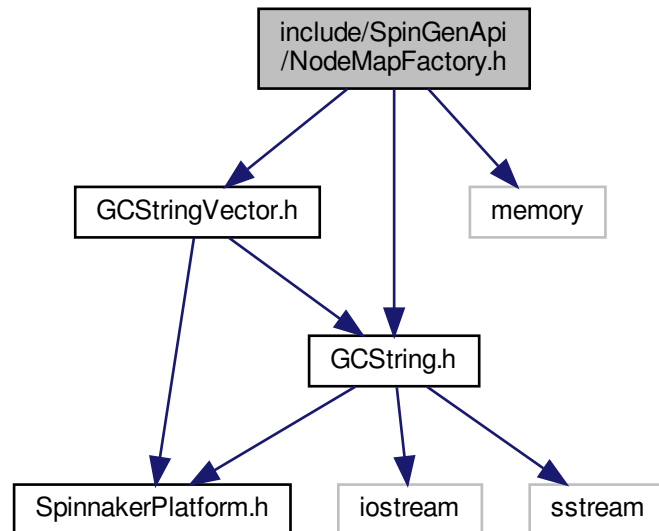
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

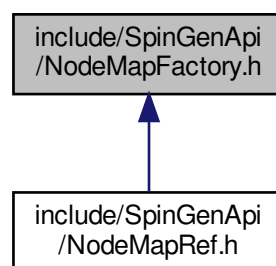
- typedef NodeMap [CNodeMapRef](#)

16.125 include/SpinGenApi/NodeMapFactory.h File Reference

Include dependency graph for NodeMapFactory.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CNodeMapFactory](#)
The node map factory is used for creating node maps from camera description files.
- struct [CNodeMapFactory::NodeStatistics_t](#)

Namespaces

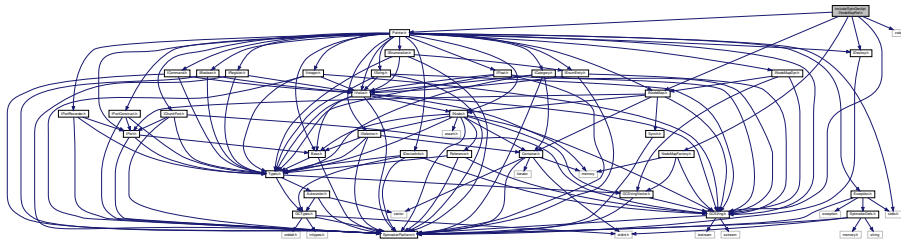
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Enumerations

- enum [ECacheUsage_t](#) {
[CacheUsage_Automatic](#),
[CacheUsage_ForceWrite](#),
[CacheUsage_ForceRead](#),
[CacheUsage_Ignore](#) }
Lists the cache usage strategies.
- enum [EContentType_t](#) {
[ContentType_Xml](#),
[ContentType_ZippedXml](#) }
Lists the processable file types.

16.126 include/SpinGenApi/NodeMapRef.h File Reference

Include dependency graph for NodeMapRef.h:



Classes

- class [CNodeMapRefT< TCameraParams >](#)
Smartpointer template for NodeMaps with create function.
- class [CGeneric_XMLLoaderParams](#)
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CNodeMapRef](#)
Smartpointer for NodeMaps with create function.

Namespaces

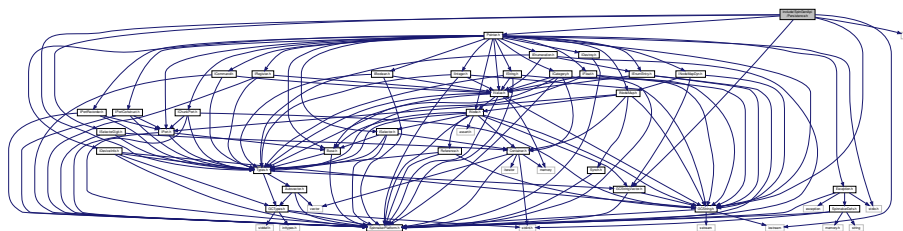
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

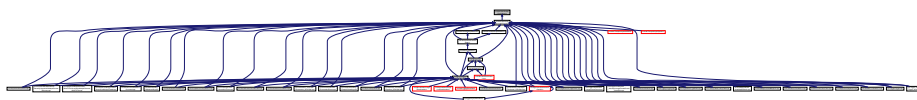
- [SPINNAKER_API](#) IDestroy * [CastToIDestroy](#) (INodeMap *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- template<class TCameraParams >
void [_LoadXMLFromFile](#) (const GenICam::gcstring &FileName)
- template<class TCameraParams >
void [_LoadXMLFromZIPFile](#) (const GenICam::gcstring &ZipFileName)
- template<class TCameraParams >
void [_LoadXMLFromFileInject](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectFileName)
- template<class TCameraParams >
void [_LoadXMLFromString](#) (const GenICam::gcstring &XMLData)
- template<class TCameraParams >
void [_LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)
- template<class TCameraParams >
void [_LoadXMLFromStringInject](#) (const GenICam::gcstring &TargetXMLData, const GenICam::gcstring &InjectXMLData)
- template<class TCameraParams >
void [_GetSupportedSchemaVersions](#) (GenICam::gcstring_vector &SchemaVersions)
- template<class TCameraParams >
GenICam::gcstring [_GetDeviceName](#) ()
- template<class TCameraParams >
void [_Poll](#) (int64_t ElapsedTime)
- template<class TCameraParams >
void [_GetNodes](#) (NodeList_t &Nodes)
- template<class TCameraParams >
INode * [_GetNode](#) (const GenICam::gcstring &key)
- template<class TCameraParams >
void [_InvalidateNodes](#) ()
- template<class TCameraParams >
bool [_Connect](#) (IPort *pPort, const GenICam::gcstring &PortName)
- template<class TCameraParams >
bool [_Connect](#) (IPort *pPort)
- template<class TCameraParams >
bool [_ClearXMLCache](#) ()

16.127 include/SpinGenApi/Persistence.h File Reference

Include dependency graph for Persistence.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CFeatureBag](#)
Bag holding streamable features of a nodetree.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

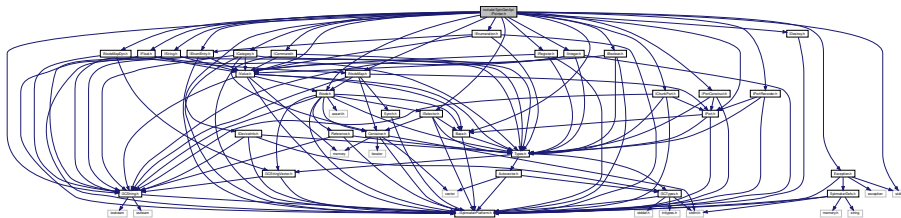
- virtual void [PersistFeature](#) (IValue &item)=0
Stores a feature.
- [SPINNAKER_API](#) std::istream & [EatComments](#) (std::istream &is)
Helper function ignoring lines starting with comment character '#'.
- [SPINNAKER_API](#) std::istream & [operator>>](#) (std::istream &is, CFeatureBag &FeatureBag)
Reads in persistent data from a stream.
- [SPINNAKER_API](#) std::ostream & [operator<<](#) (std::ostream &os, const CFeatureBag &FeatureBag)
writes out persistent data to a stream

Variables

- interface [SPINNAKER_API_ABSTRACT IPersistScript](#)
Basic interface to persist values to.

16.128 include/SpinGenApi/Pointer.h File Reference

Include dependency graph for Pointer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CPointer< T, B >](#)
Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.
- class [CFloatPtr](#)
SmartPointer for IFloat interface pointer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef CPointer< IBase > [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef CPointer< INode, IBase > [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef CPointer< IValue > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef CPointer< ICategory > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef CPointer< IBoolean > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef CPointer< IInteger > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef CPointer< IString > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef CPointer< IRegister > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef CPointer< IEnumeration > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef CPointer< IEnumEntry > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef CPointer< IPort > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef CPointer< IPortReplay > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef CPointer< IPortRecorder > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef CPointer< IPortWriteList, IPortWriteList > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef CPointer< IChunkPort > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef CPointer< INodeMap, INodeMap > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef CPointer< INodeMapDyn, INodeMap > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef CPointer< IDeviceInfo, INodeMap > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.

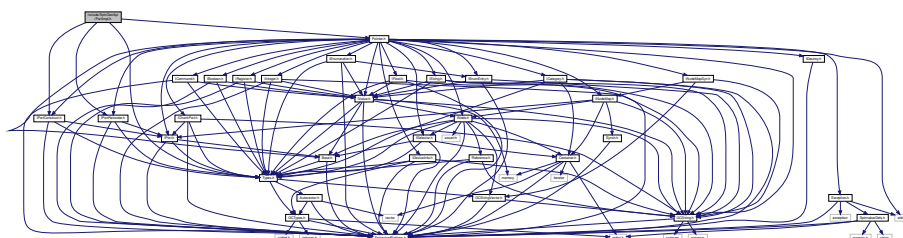
- typedef CPointer< ISelector > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef CPointer< ICommand > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef CPointer< IPortConstruct > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.

Functions

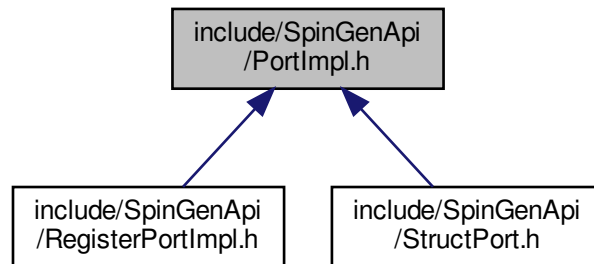
- template<class T , class B >
bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is readable.
- template<class T , class B >
bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Writable.
- template<class T , class B >
bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Implemented.
- template<class T , class B >
bool [IsAvailable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Available.
- GenICam::gcstring [GetInterfaceName](#) (IBase *pBase)
Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.

16.129 include/SpinGenApi/PortImpl.h File Reference

Include dependency graph for PortImpl.h:



This graph shows which files directly or indirectly include this file:



Classes

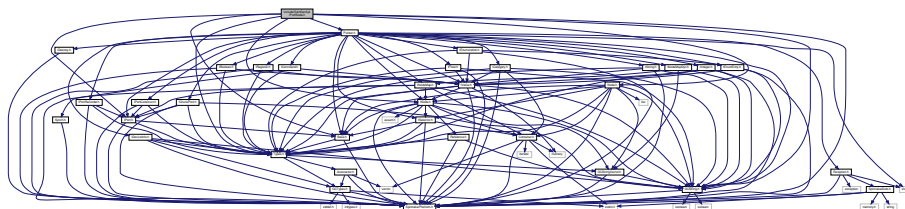
- class [CPortImpl](#)
Standard implementation for a port.

Namespaces

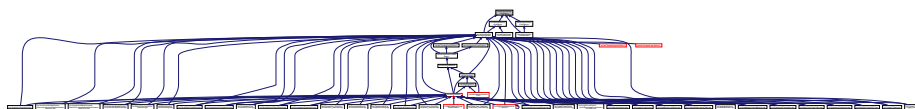
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.130 include/SpinGenApi/PortNode.h File Reference

Include dependency graph for PortNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [PortNode](#)
Interface for value properties.

Namespaces

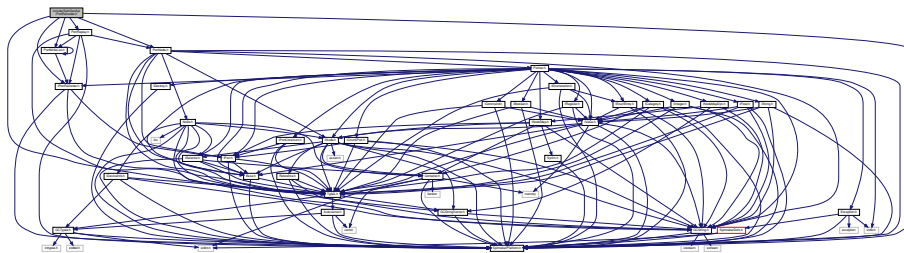
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef PortNode [CPortRef](#)

16.131 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



Classes

- class [PortRecorder](#)
Interface for recording write commands on a port.

Namespaces

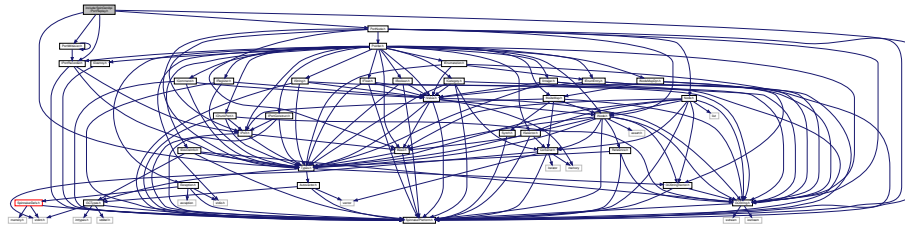
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

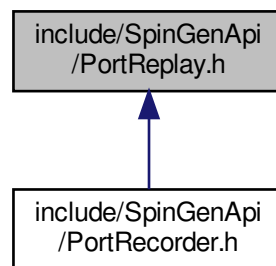
- typedef PortRecorder [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.

16.132 include/SpinGenApi/PortReplay.h File Reference

Include dependency graph for PortReplay.h:



This graph shows which files directly or indirectly include this file:



Classes

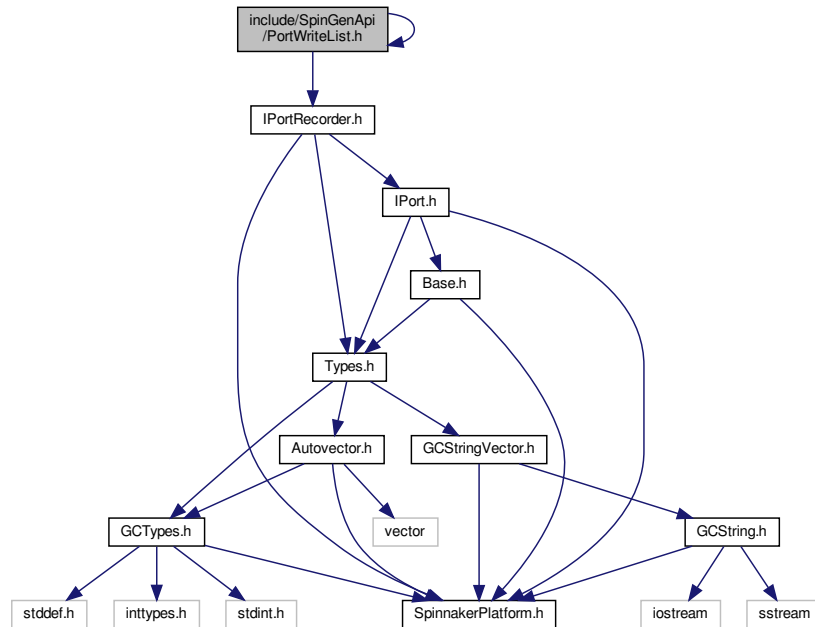
- class [PortReplay](#)
Interface for replaying write commands on a port.

Namespaces

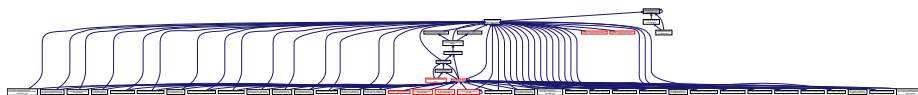
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.133 include/SpinGenApi/PortWriteList.h File Reference

Include dependency graph for PortWriteList.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CPortWriteList](#)

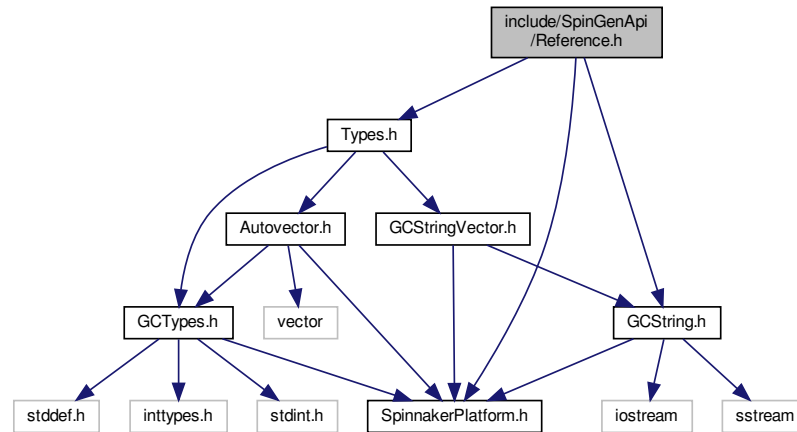
Container holding a list of port write commands.

Namespaces

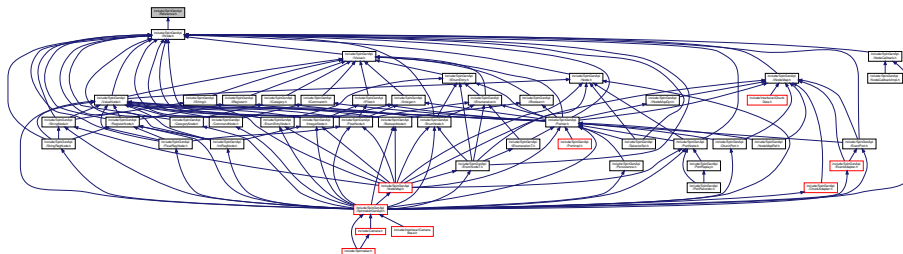
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.134 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



Namespaces

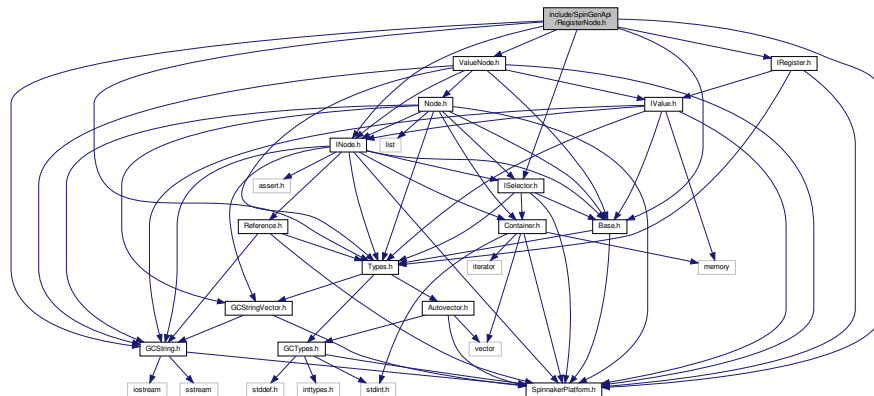
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

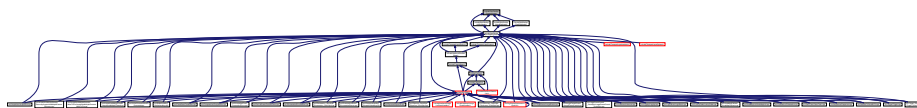
- virtual void [SetNumEnums](#) (int NumEnums)=0
sets the number of enum values

16.135 include/SpinGenApi/RegisterNode.h File Reference

Include dependency graph for RegisterNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `RegisterNode`
Interface for string properties.

Namespaces

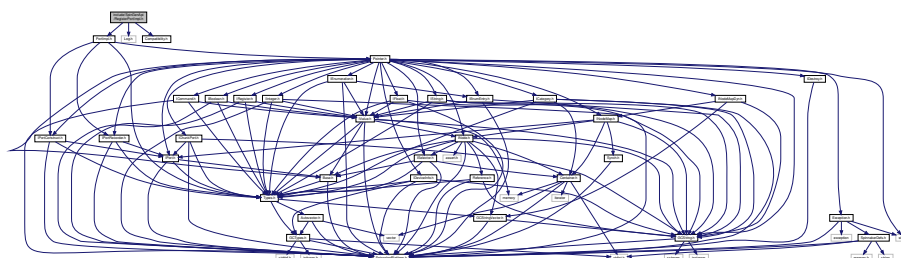
- Spinnaker
- Spinnaker::GenApi

Typedefs

- typedef RegisterNode CRegisterRef

16.136 include/SpinGenApi/RegisterPortImpl.h File Reference

Include dependency graph for RegisterPortImpl.h:



Classes

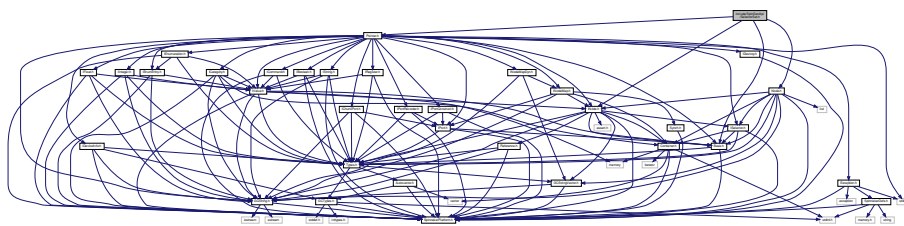
- class [CRegisterPortImpl](#)
Standard implementation for a port using a register based transport layer.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.137 include/SpinGenApi/SelectorSet.h File Reference

Include dependency graph for SelectorSet.h:



Classes

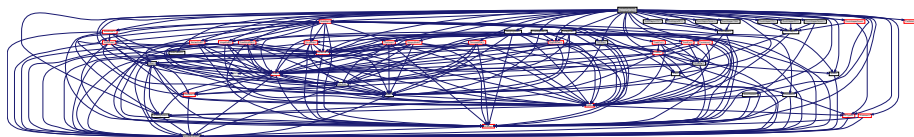
- class [CSelectorSet](#)
The set of selectors selecting a given node.

Namespaces

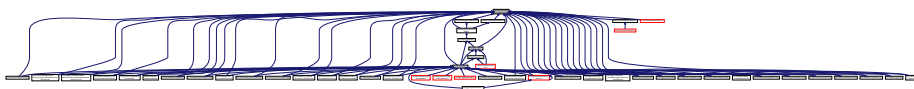
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.138 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

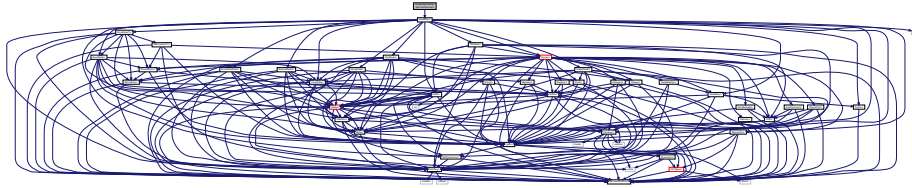


This graph shows which files directly or indirectly include this file:



16.139 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



Classes

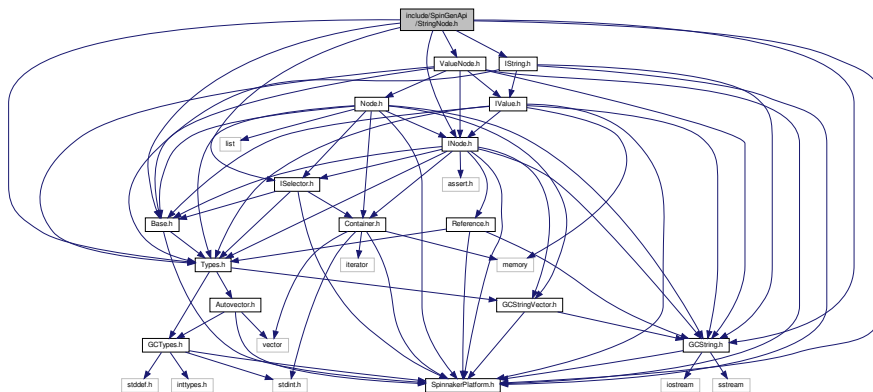
- class [SpinTestCamera](#)

Namespaces

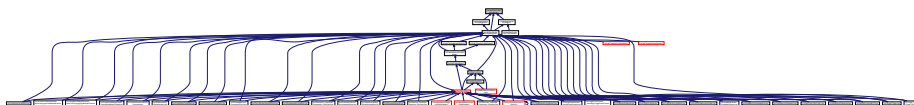
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.140 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:

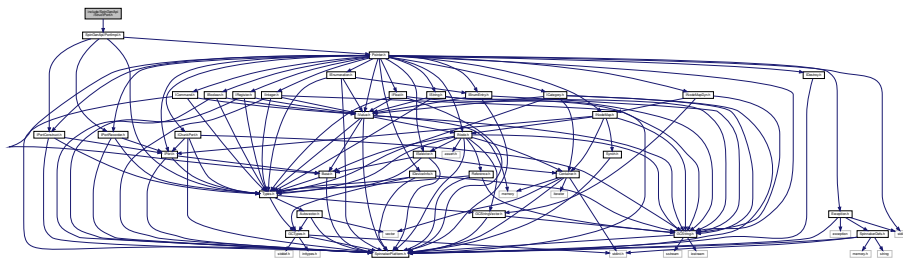


This graph shows which files directly or indirectly include this file:



16.142 include/SpinGenApi/StructPort.h File Reference

Include dependency graph for StructPort.h:



Classes

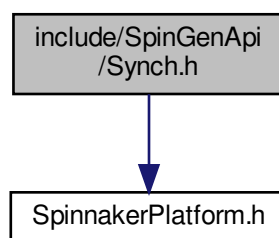
- class `CTestPortStruct< CDataStruct >`
Implements a register spaces based on a C++ struct.

Namespaces

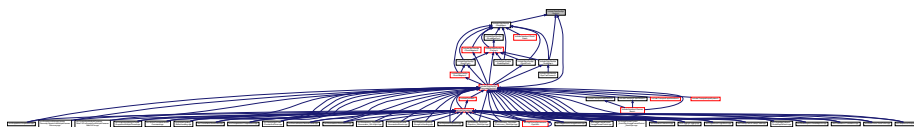
- `Spinnaker`
- `Spinnaker::GenApi`

16.143 include/SpinGenApi/Synch.h File Reference

Include dependency graph for Synch.h:



This graph shows which files directly or indirectly include this file:



Classes

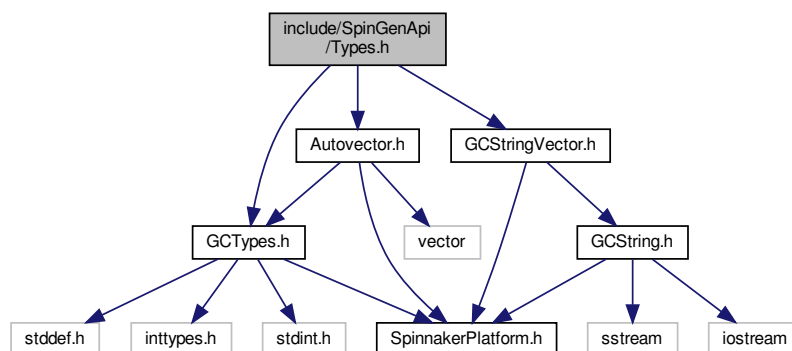
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)

Namespaces

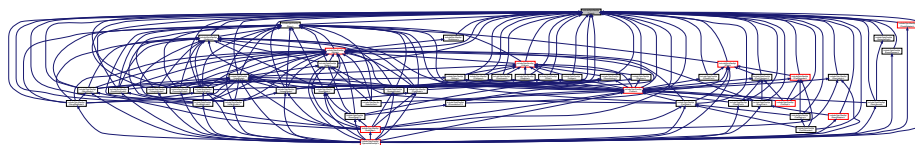
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

16.144 include/SpinGenApi/Types.h File Reference

Include dependency graph for Types.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Macros

- `#define` [interface](#) struct
- `#define` [_UndefinedRepresentation](#) _UndefinedRepresentation

Typedefs

- typedef GenICam::gcstring_vector [StringList_t](#)
A list of strings.

Enumerations

- enum [_ESign](#) {
Signed,
Unsigned,
_UndefinedSign }
signed or unsigned integers
- enum [_EAccessMode](#) {
NI,
NA,
WO,
RO,
RW,
_UndefinedAccesMode,
_CycleDetectAccesMode }
access mode of a node
- enum [_EVisibility](#) {
Beginner = 0,
Expert = 1,
Guru = 2,
Invisible = 3,
_UndefinedVisibility = 99 }
recommended visibility of a node
- enum [_ECachingMode](#) {
NoCache,
WriteThrough,
WriteAround,
_UndefinedCachingMode }
caching mode of a register
- enum [_ERepresentation](#) {
Linear,
Logarithmic,
Boolean,
PureNumber,
HexNumber,
IPV4Address,
MACAddress,
_UndefinedRepresentation }
recommended representation of a node value
- enum [_EEndianess](#) {
BigEndian,
LittleEndian,
_UndefinedEndian }
Endianess of a value in a register.
- enum [_ENameSpace](#) {
Custom,
Standard,
_UndefinedNameSpace }
Defines if a node name is standard or custom.

- enum `_EStandardNameSpace` {
`None`,
`GEV`,
`IIDC`,
`CL`,
`USB`,
`_UndefinedStandardNameSpace` }

Defines from which standard namespace a node name comes from.

- enum `_EYesNo` {
`Yes` = 1,
`No` = 0,
`_UndefinedYesNo` = 2 }

Defines the choices of a Yes/No alternative.

- enum `_ESlope` {
`Increasing`,
`Decreasing`,
`Varying`,
`Automatic`,
`_UndefinedESlope` }

typedef for formula type

- enum `_EXMLValidation` {
`xvLoad` = 0x00000001L,
`xvCycles` = 0x00000002L,
`xvSFNC` = 0x00000004L,
`xvDefault` = 0x00000000L,
`xvAll` = 0xffffffffL,
`_UndefinedEXMLValidation` = 0x80000000L }

typedef describing the different validity checks which can be performed on an XML file

- enum `_EDisplayNotation` {
`fnAutomatic`,
`fnFixed`,
`fnScientific`,
`_UndefinedEDisplayNotation` }

typedef for float notation

- enum `_EInterfaceType` {
`intflValue`,
`intflBase`,
`intflInteger`,
`intflBoolean`,
`intflCommand`,
`intflFloat`,
`intflString`,
`intflRegister`,
`intflCategory`,
`intflEnumeration`,
`intflEnumEntry`,
`intflPort` }

typedef for interface type

- enum `_ELinkType` {
`ctParentNodes`,
`ctReadingChildren`,
`ctWritingChildren`,
`ctInvalidatingChildren`,
`ctDependingNodes`,
`ctTerminalNodes` }

typedef for link type

- enum `_EIncMode` {
`noIncrement`,
`fixedIncrement`,
`listIncrement` }
typedef for increment mode
- enum `_EInputDirection` {
`idFrom`,
`idTo`,
`idNone` }
typedef for link type
- enum `_EGenApiSchemaVersion` {
`v1_0` = 1,
`v1_1` = 2,
`_Undefined` = -1 }
GenApi schema version.

16.144.1 Macro Definition Documentation

16.144.1.1 `_UndefinedRepresentation`

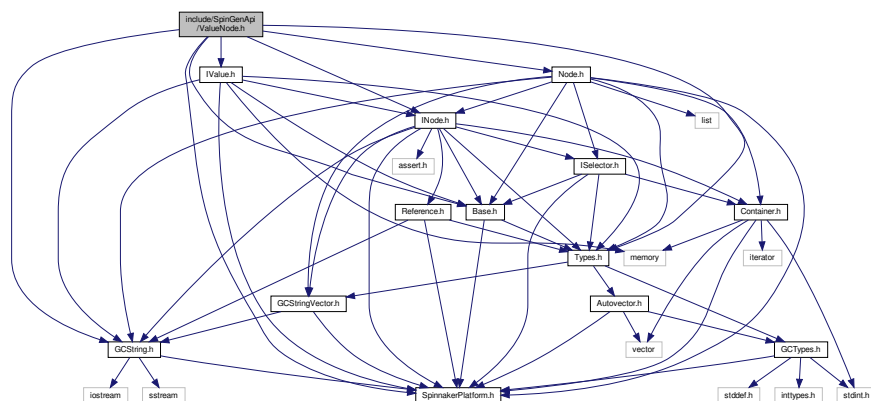
```
#define _UndefinedRepresentation _UndefinedRepresentation
```

16.144.1.2 interface

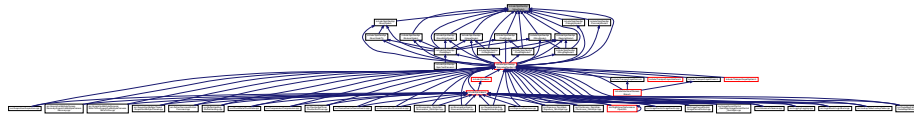
```
#define interface struct
```

16.145 include/SpinGenApi/ValueNode.h File Reference

Include dependency graph for ValueNode.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ValueNode](#)
Interface for value properties.

Namespaces

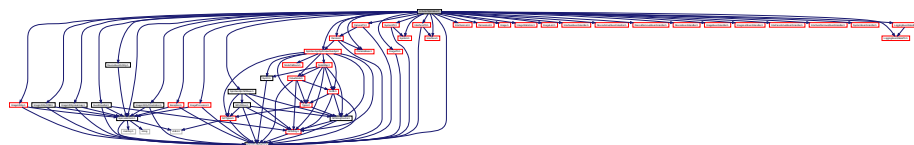
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef ValueNode [CValueRef](#)

16.146 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

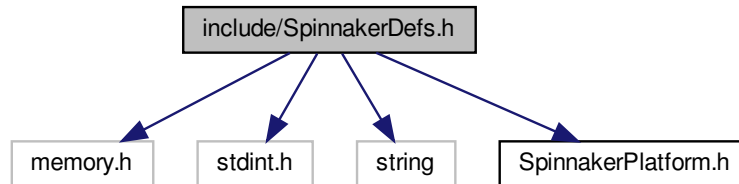


This graph shows which files directly or indirectly include this file:



16.147 include/SpinnakerDefs.h File Reference

Include dependency graph for SpinnakerDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [PNGOption](#)
Spinnaker struct definitions.
- struct [PPMOption](#)
Options for saving PPM images.
- struct [PGMOption](#)
Options for saving PGM images.
- struct [TIFFOption](#)
Options for saving TIFF images.
- struct [JPEGOption](#)
Options for saving JPEG image.
- struct [JPG2Option](#)
Options for saving JPEG2000 image.
- struct [BMPOption](#)
Options for saving Bitmap image.
- struct [SIOption](#)
Options for saving Spinnaker image.
- struct [LibraryVersion](#)
Provides easier access to the current version of Spinnaker.
- struct [ActionCommandResult](#)
Action Command Result.
- struct [CCMSettings](#)
- struct [DeviceEventInferenceData](#)
Data fields for Device Event payload for EventInference.
- struct [DeviceEventExposureEndData](#)
Data Fields for Device Event payload for EventExposureEnd.

Namespaces

- [Spinnaker](#)

Enumerations

- enum [Error](#) {
[SPINNAKER_ERR_SUCCESS](#) = 0,
[SPINNAKER_ERR_ERROR](#) = -1001,
[SPINNAKER_ERR_NOT_INITIALIZED](#) = -1002,
[SPINNAKER_ERR_NOT_IMPLEMENTED](#) = -1003,
[SPINNAKER_ERR_RESOURCE_IN_USE](#) = -1004,
[SPINNAKER_ERR_ACCESS_DENIED](#) = -1005,
[SPINNAKER_ERR_INVALID_HANDLE](#) = -1006,
[SPINNAKER_ERR_INVALID_ID](#) = -1007,
[SPINNAKER_ERR_NO_DATA](#) = -1008,
[SPINNAKER_ERR_INVALID_PARAMETER](#) = -1009,
[SPINNAKER_ERR_IO](#) = -1010,
[SPINNAKER_ERR_TIMEOUT](#) = -1011,
[SPINNAKER_ERR_ABORT](#) = -1012,
[SPINNAKER_ERR_INVALID_BUFFER](#) = -1013,
[SPINNAKER_ERR_NOT_AVAILABLE](#) = -1014,
[SPINNAKER_ERR_INVALID_ADDRESS](#) = -1015,
[SPINNAKER_ERR_BUFFER_TOO_SMALL](#) = -1016,
[SPINNAKER_ERR_INVALID_INDEX](#) = -1017,
[SPINNAKER_ERR_PARSING_CHUNK_DATA](#) = -1018,
[SPINNAKER_ERR_INVALID_VALUE](#) = -1019,
[SPINNAKER_ERR_RESOURCE_EXHAUSTED](#) = -1020,
[SPINNAKER_ERR_OUT_OF_MEMORY](#) = -1021,
[SPINNAKER_ERR_BUSY](#) = -1022,
[SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT](#) = -2001,
[SPINNAKER_ERR_GENICAM_OUT_OF_RANGE](#) = -2002,
[SPINNAKER_ERR_GENICAM_PROPERTY](#) = -2003,
[SPINNAKER_ERR_GENICAM_RUN_TIME](#) = -2004,
[SPINNAKER_ERR_GENICAM_LOGICAL](#) = -2005,
[SPINNAKER_ERR_GENICAM_ACCESS](#) = -2006,
[SPINNAKER_ERR_GENICAM_TIMEOUT](#) = -2007,
[SPINNAKER_ERR_GENICAM_DYNAMIC_CAST](#) = -2008,
[SPINNAKER_ERR_GENICAM_GENERIC](#) = -2009,
[SPINNAKER_ERR_GENICAM_BAD_ALLOCATION](#) = -2010,
[SPINNAKER_ERR_IM_CONVERT](#) = -3001,
[SPINNAKER_ERR_IM_COPY](#) = -3002,
[SPINNAKER_ERR_IM_MALLOC](#) = -3003,
[SPINNAKER_ERR_IM_NOT_SUPPORTED](#) = -3004,
[SPINNAKER_ERR_IM_HISTOGRAM_RANGE](#) = -3005,
[SPINNAKER_ERR_IM_HISTOGRAM_MEAN](#) = -3006,
[SPINNAKER_ERR_IM_MIN_MAX](#) = -3007,
[SPINNAKER_ERR_IM_COLOR_CONVERSION](#) = -3008,
[SPINNAKER_ERR_IM_DECOMPRESSION](#) = -3009,
[SPINNAKER_ERR_CUSTOM_ID](#) = -10000 }

Spinnaker enum definitions.

- enum [EventType](#) {
[SPINNAKER_EVENT_ARRIVAL_REMOVAL](#),
[SPINNAKER_EVENT_DEVICE](#),
[SPINNAKER_EVENT_DEVICE_SPECIFIC](#),
[SPINNAKER_EVENT_NEW_BUFFER](#),
[SPINNAKER_EVENT_NEW_BUFFER_LIST](#),

```
SPINNAKER_EVENT_LOGGING_EVENT,
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL,
SPINNAKER_EVENT_UNKNOWN }
```

Event types in Spinnaker.

- enum `TLPixelFormatNamespace` {
`SPINNAKER_TLPixelFormatNamespace_UNKNOWN` = 0,
`SPINNAKER_TLPixelFormatNamespace_GEV` = 1,
`SPINNAKER_TLPixelFormatNamespace_IIDC` = 2,
`SPINNAKER_TLPixelFormatNamespace_PFNC_16BIT` = 3,
`SPINNAKER_TLPixelFormatNamespace_PFNC_32BIT` = 4,
`SPINNAKER_TLPixelFormatNamespace_CUSTOM_ID` = 1000 }

This enum represents the namespace in which the TL specific pixel format resides.

- enum `ColorProcessingAlgorithm` {
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONE`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_AVG`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_LINEAR`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECTIONAL_FILTER`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGOROUS`,
`SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEIGHTED_DIRECTIONAL_FILTER` }

Color processing algorithms.

- enum `ImageFileFormat` {
`SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT` = -1,
`SPINNAKER_IMAGE_FILE_FORMAT_PGM`,
`SPINNAKER_IMAGE_FILE_FORMAT_PPM`,
`SPINNAKER_IMAGE_FILE_FORMAT_BMP`,
`SPINNAKER_IMAGE_FILE_FORMAT_JPEG`,
`SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000`,
`SPINNAKER_IMAGE_FILE_FORMAT_TIFF`,
`SPINNAKER_IMAGE_FILE_FORMAT_PNG`,
`SPINNAKER_IMAGE_FILE_FORMAT_RAW`,
`SPINNAKER_IMAGE_FILE_FORMAT_JPEG12_C`,
`SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS` = 0x7FFFFFFF }

File formats to be used for saving images to disk.

- enum `ImageStatus` {
`SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR` = -1,
`SPINNAKER_IMAGE_STATUS_NO_ERROR` = 0,
`SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED` = 1,
`SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW` = 2,
`SPINNAKER_IMAGE_STATUS_MISSING_PACKETS`,
`SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT`,
`SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT`,
`SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT`,
`SPINNAKER_IMAGE_STATUS_MISSING_LEADER` = 7,
`SPINNAKER_IMAGE_STATUS_MISSING_TRAILER`,
`SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE` = 9,
`SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT`,
`SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID` = 11,
`SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES` = 12 }

Status of images returned from `GetNextImage()` or `Convert()` calls.

- enum `StatisticsChannel` {
`SPINNAKER_STATISTICS_CHANNEL_GREY`,
`SPINNAKER_STATISTICS_CHANNEL_RED`,
`SPINNAKER_STATISTICS_CHANNEL_GREEN`,

```
SPINNAKER_STATISTICS_CHANNEL_BLUE,
SPINNAKER_STATISTICS_CHANNEL_HUE,
SPINNAKER_STATISTICS_CHANNEL_SATURATION,
SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS,
SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS }
```

Channels that allow statistics to be calculated.

- enum SpinnakerLogLevel {
 SPINNAKER_LOG_LEVEL_OFF = -1,
 SPINNAKER_LOG_LEVEL_FATAL = 0,
 SPINNAKER_LOG_LEVEL_ALERT = 100,
 SPINNAKER_LOG_LEVEL_CRIT = 200,
 SPINNAKER_LOG_LEVEL_ERROR = 300,
 SPINNAKER_LOG_LEVEL_WARN = 400,
 SPINNAKER_LOG_LEVEL_NOTICE = 500,
 SPINNAKER_LOG_LEVEL_INFO = 600,
 SPINNAKER_LOG_LEVEL_DEBUG = 700,
 SPINNAKER_LOG_LEVEL_NOTSET = 800 }

Log levels.

- enum ImagePayloadType {
 SPINNAKER_IMAGE_PAYLOAD_TYPE_UNKNOWN = -1,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_IMAGE,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_EXTENDED_CHUNK,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSLESS_COMPRESSED,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSY_COMPRESSED,
 SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED }
- enum TLPayloadType {
 SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN = 0,
 SPINNAKER_TLPAYLOAD_TYPE_IMAGE = 1,
 SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA = 2,
 SPINNAKER_TLPAYLOAD_TYPE_FILE = 3,
 SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA = 4,
 SPINNAKER_TLPAYLOAD_TYPE_JPEG = 5,
 SPINNAKER_TLPAYLOAD_TYPE_JPEG2000 = 6,
 SPINNAKER_TLPAYLOAD_TYPE_H264 = 7,
 SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY = 8,
 SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC = 9,
 SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART = 10,
 SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID = 1000,
 SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID + 1,
 SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID + 2,
 SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED = SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID + 3 }
- enum ActionCommandStatus {
 SPINNAKER_ACTION_COMMAND_STATUS_OK = 0,
 SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME,
 SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW = 0x8015,
 SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE,
 SPINNAKER_ACTION_COMMAND_STATUS_ERROR }

Possible Status Codes Returned from Action Command.

- enum TIFFCompressionMethod {
 SPINNAKER_TIFF_COMPRESS_METHOD_NONE = 1,
 SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS,
 SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE,
 SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE,


```
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3,
SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4,
SPINNAKER_TIFF_COMPRESS_METHOD_LZW,
SPINNAKER_TIFF_COMPRESS_METHOD_JPEG }
```

Compression method to use for encoding TIFF images.

- enum `PixelFormatIntType` {
`SPINNAKER_INT_TYPE_UINT8`,
`SPINNAKER_INT_TYPE_INT8`,
`SPINNAKER_INT_TYPE_UINT10`,
`SPINNAKER_INT_TYPE_UINT10p`,
`SPINNAKER_INT_TYPE_UINT10P`,
`SPINNAKER_INT_TYPE_UINT12`,
`SPINNAKER_INT_TYPE_UINT12p`,
`SPINNAKER_INT_TYPE_UINT12P`,
`SPINNAKER_INT_TYPE_UINT14`,
`SPINNAKER_INT_TYPE_UINT16`,
`SPINNAKER_INT_TYPE_INT16`,
`SPINNAKER_INT_TYPE_FLOAT32`,
`SPINNAKER_INT_TYPE_UNKNOWN` }

Possible integer types and packing used in a pixel format.

- enum `BufferOwnership` {
`SPINNAKER_BUFFER_OWNERSHIP_SYSTEM`,
`SPINNAKER_BUFFER_OWNERSHIP_USER` }
- enum `ImageScalingAlgorithm` { `SPINNAKER_IMAGE_SCALING_ALGORITHM_NEAREST_NEIGHBOR` }

Image scaling algorithms.

- enum `SourceDataRange` {
`SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE`,
`SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_DATA_RANGE`,
`SPINNAKER_SOURCE_DATA_RANGE_IMAGE_MIN_ABSOLUTE_MAX`,
`SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_MIN_IMAGE_MAX` }

Image normalization source data options.

- enum `HeatmapColor` {
`SPINNAKER_HEATMAP_COLOR_BLACK` = 1,
`SPINNAKER_HEATMAP_COLOR_BLUE` = 2,
`SPINNAKER_HEATMAP_COLOR_CYAN` = 3,
`SPINNAKER_HEATMAP_COLOR_GREEN` = 4,
`SPINNAKER_HEATMAP_COLOR_YELLOW` = 5,
`SPINNAKER_HEATMAP_COLOR_RED` = 6,
`SPINNAKER_HEATMAP_COLOR_WHITE` = 7 }

Color specifiers for the heatmap color gradient.

- enum `PolarizationQuadrant` {
`SPINNAKER_POLARIZATION_QUADRANT_I0`,
`SPINNAKER_POLARIZATION_QUADRANT_I45`,
`SPINNAKER_POLARIZATION_QUADRANT_I90`,
`SPINNAKER_POLARIZATION_QUADRANT_I135` }

Polarization quadrant specifiers describing the four orientations of linear polarizing filters on polarized cameras.

- enum `CCMColorTemperature` {
`SPINNAKER_CCM_COLOR_TEMP_TUNGSTEN_2800K`,
`SPINNAKER_CCM_COLOR_TEMP_WARM_FLUORESCENT_3000K`,
`SPINNAKER_CCM_COLOR_TEMP_COOL_FLUORESCENT_4000K`,
`SPINNAKER_CCM_COLOR_TEMP_SUNNY_5000K`,
`SPINNAKER_CCM_COLOR_TEMP_CLOUDY_6500K`,
`SPINNAKER_CCM_COLOR_TEMP_SHADE_8000K`,
`SPINNAKER_CCM_COLOR_TEMP_GENERAL` }
- enum `CCMType` {
`SPINNAKER_CCM_TYPE_LINEAR`,
`SPINNAKER_CCM_TYPE_ADVANCED` }

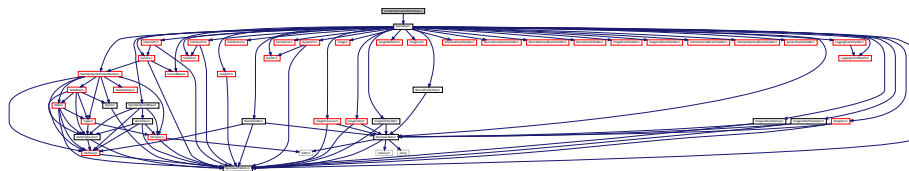
- enum [CCMSensor](#) { [SPINNAKER_CCM_SENSOR_IMX250](#) }
- enum [CCMColorSpace](#) { [SPINNAKER_CCM_COLOR_SPACE_OFF](#), [SPINNAKER_CCM_COLOR_SPACE_SRGB](#) }
- enum [CCMApplication](#) { [SPINNAKER_CCM_APPLICATION_GENERIC](#), [SPINNAKER_CCM_APPLICATION_MICROSCOPY](#) }

Variables

- const uint64_t [EVENT_TIMEOUT_NONE](#) = 0
Timeout values for getting next image, device, or interface event.
- const uint64_t [EVENT_TIMEOUT_INFINITE](#) = 0xFFFFFFFFFFFFFFFF

16.148 include/SpinnakerDirectShow.h File Reference

Include dependency graph for SpinnakerDirectShow.h:



Functions

- `STDMETHOD()` [GetSelectedCameraIndex](#) (unsigned int *selectedIndex)=0
Retrieves an integer index to the currently selected camera.
- `STDMETHOD()` [SetSelectedCameraIndex](#) (unsigned int index, bool needsRelease=true)=0
Sets the currently selected camera to the index specified.
- `STDMETHOD()` [GetCameraInfo](#) (unsigned int index, char *model, char *type, char *serial, size_t bufferSize)=0
Retrieves general information about the camera at the specified index.
- `STDMETHOD()` [IsStreaming](#) (bool *isStreaming)=0
Checks if the selected camera is currently streaming.
- `STDMETHOD()` [BeginAcquisition](#) ()=0
Starts the image acquisition engine for the currently selected camera.
- `STDMETHOD()` [EndAcquisition](#) ()=0
Stops the image acquisition engine for the currently selected camera.

Spinnaker GenAPI Functions

These functions deal with [Spinnaker](#) GenAPI NodeMap and Node Accesses for setting camera properties.

- `STDMETHOD()` [NodeMapGetNumNodes](#) (size_t *numNodes)=0
Retrieves the number of nodes available in the node map for the currently selected camera.
- `STDMETHOD()` [NodeMapGetNodeAtIndex](#) (size_t index, char *nodeName, size_t bufferSize)=0
Retrieves the string representation of the node at the specified index.
- `STDMETHOD()` [NodeIsImplemented](#) (const char *nodeName, bool *isImplemented)=0

- Checks if a node is implemented.*
- STDMETHOD() [NodesAvailable](#) (const char *nodeName, bool *isAvailable)=0
- Checks if a node is available.*
- STDMETHOD() [NodesReadable](#) (const char *nodeName, bool *isReadable)=0
- Checks if a node is readable.*
- STDMETHOD() [NodesWritable](#) (const char *nodeName, bool *isWritable)=0
- Checks if a node is writable.*
- STDMETHOD() [NodeGetType](#) (const char *nodeName, char *typeName, size_t bufferSize)=0
- Gets the type of the node.*
- STDMETHOD() [NodeToString](#) (const char *nodeName, char *valueAsString, size_t bufferSize)=0
- Gets content of the node as string.*
- STDMETHOD() [NodeGetDisplayName](#) (const char *nodeName, char *displayName, size_t bufferSize)=0
- Gets a name string for display.*

Integer Node Functions

These functions deal with [Spinnaker](#) GenAPI Integer Node Accesses

- STDMETHOD() [IntegerGetValue](#) (const char *nodeName, int64_t *value)=0
- Gets integer node value for the specified feature.*
- STDMETHOD() [IntegerSetValue](#) (const char *nodeName, int64_t value)=0
- Sets integer node value for the specified feature.*
- STDMETHOD() [IntegerGetMax](#) (const char *nodeName, int64_t *maxValue)=0
- Gets maximum integer value allowed for the specified feature.*
- STDMETHOD() [IntegerGetMin](#) (const char *nodeName, int64_t *minValue)=0
- Gets minimum integer value allowed for the specified feature.*
- STDMETHOD() [IntegerGetIncMode](#) (const char *nodeName, Spinnaker::GenApi::ElncMode *incMode)=0
- Gets integer increment mode for the specified feature.*
- STDMETHOD() [IntegerGetInc](#) (const char *nodeName, int64_t *increment)=0
- Gets integer step increment for the specified feature.*

Float Node Functions

These functions deal with [Spinnaker](#) GenAPI Float Node Accesses

- STDMETHOD() [FloatGetValue](#) (const char *nodeName, double *value)=0
- Gets float node value for the specified feature.*
- STDMETHOD() [FloatSetValue](#) (const char *nodeName, double value)=0
- Sets float node value for the specified feature.*
- STDMETHOD() [FloatGetMax](#) (const char *nodeName, double *floatMax)=0
- Gets maximum float value allowed for the specified feature.*
- STDMETHOD() [FloatGetMin](#) (const char *nodeName, double *floatMin)=0
- Gets minimum float value allowed for the specified feature.*
- STDMETHOD() [FloatGetIncMode](#) (const char *nodeName, Spinnaker::GenApi::ElncMode *incMode)=0
- Gets float increment mode for the specified feature.*
- STDMETHOD() [FloatGetInc](#) (const char *nodeName, double *increment)=0
- Gets float step increment for the specified feature.*

Boolean Node Functions

These functions deal with [Spinnaker](#) GenAPI Boolean Node Accesses

- STDMETHOD() [BooleanGetValue](#) (const char *nodeName, bool *value)=0
- Gets boolean node value for the specified feature.*
- STDMETHOD() [BooleanSetValue](#) (const char *nodeName, bool value)=0
- Sets boolean node value for the specified feature.*

String Node Functions

These functions deal with [Spinnaker](#) GenAPI String Node Accesses

- STDMETHOD() [StringGetValue](#) (const char *nodeName, char *value, size_t bufferSize)=0
Gets string node value for the specified feature.
- STDMETHOD() [StringSetValue](#) (const char *nodeName, const char *value)=0
Sets string node value for the specified feature.

Command Node Functions

These functions deal with [Spinnaker](#) GenAPI Command Node Accesses

- STDMETHOD() [CommandExecute](#) (const char *nodeName)=0
Executes the command for the specified feature.

Variables

- const unsigned int [MAX_LENGTH](#) = 256
- static const GUID [IID_ISpinnakerInterface](#)
This is the Interface that allows users to get and set device node properties on the camera.

Enumeration Node Functions

These functions deal with [Spinnaker](#) GenAPI Enumeration and EnumeratEntry Node Accesses

- const char * [enumerationName](#)
Checks if an enumeration entry exists for the specified enumeration feature.
- const char const char * [enumerationEntryName](#)
- const char const char bool * [entryExist](#) = 0
- const char unsigned int [entryIndex](#)
- const char unsigned int char size_t [bufferSize](#) = 0
- STDMETHOD() [EnumerationGetEntry](#) (const char *nodeName, char *value, size_t bufferSize)=0
Gets enumeration entry string for the specified enumeration feature.
- STDMETHOD() [EnumerationSetEntry](#) (const char *nodeName, const char *value)=0
Sets enumeration entry for the specified enumeration feature.
- STDMETHOD() [EnumerationGetNumEntries](#) (const char *enumerationName, size_t *numEntries)=0
Retrieves the number of enumeration entry nodes available for the specified enumeration feature.

16.148.1 Function Documentation

16.148.1.1 BeginAcquisition()

```
STDMETHOD() BeginAcquisition ( ) [pure virtual]
```

Starts the image acquisition engine for the currently selected camera.

Nothing occurs if the camera is already streaming.

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.2 BooleanGetValue()

```
STDMETHOD() BooleanGetValue (
    const char * nodeName,
    bool * value ) [pure virtual]
```

Gets boolean node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.3 BooleanSetValue()

```
STDMETHOD() BooleanSetValue (  
    const char * nodeName,  
    bool value ) [pure virtual]
```

Sets boolean node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.4 CommandExecute()

```
STDMETHOD() CommandExecute (  
    const char * nodeName ) [pure virtual]
```

Executes the command for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name

Returns

An HRESULT error code indicating the success or failure of the function.

16.148.1.5 EndAcquisition()

```
STDMETHOD() EndAcquisition ( ) [pure virtual]
```

Stops the image acquisition engine for the currently selected camera.

Nothing occurs if there were no prior call to [BeginAcquisition\(\)](#). Note that [EndAcquisition\(\)](#) needs to be called before selecting a new camera.

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.6 EnumerationGetEntry()

```
STDMETHOD() EnumerationGetEntry (
    const char * nodeName,
    char * value,
    size_t bufferSize ) [pure virtual]
```

Gets enumeration entry string for the specified enumeration feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The enumeration entry string read from the device
<i>bufferSize</i>	Size of the provided value buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.7 EnumerationGetNumEntries()

```
STDMETHOD() EnumerationGetNumEntries (
    const char * enumerationName,
    size_t * numEntries ) [pure virtual]
```

Retrieves the number of enumeration entry nodes available for the specified enumeration feature.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>numEntries</i>	Number of nodes available for the specified enumeration feature

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.8 EnumerationSetEntry()

```
STDMETHOD() EnumerationSetEntry (
    const char * nodeName,
    const char * value ) [pure virtual]
```

Sets enumeration entry for the specified enumeration feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The enumeration entry string to set to device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.9 FloatGetInc()

```
STDMETHOD() FloatGetInc (
    const char * nodeName,
    double * increment ) [pure virtual]
```

Gets float step increment for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>increment</i>	The step increment value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.10 FloatGetIncMode()

```
STDMETHOD() FloatGetIncMode (
    const char * nodeName,
    Spinnaker::GenApi::EIncMode * incMode ) [pure virtual]
```

Gets float increment mode for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>incMode</i>	Spinnaker GenAPI increment mode for the specified feature

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.11 FloatGetMax()

```
STDMETHOD() FloatGetMax (  
    const char * nodeName,  
    double * floatMax ) [pure virtual]
```

Gets maximum float value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>floatMax</i>	The maximum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.12 FloatGetMin()

```
STDMETHOD() FloatGetMin (  
    const char * nodeName,  
    double * floatMin ) [pure virtual]
```

Gets minimum float value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>floatMin</i>	The minimum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.13 FloatGetValue()

```
STDMETHOD() FloatGetValue (
    const char * nodeName,
    double * value ) [pure virtual]
```

Gets float node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.14 FloatSetValue()

```
STDMETHOD() FloatSetValue (
    const char * nodeName,
    double value ) [pure virtual]
```

Sets float node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.15 GetCameraInfo()

```
STDMETHOD() GetCameraInfo (
    unsigned int index,
    char * model,
    char * type,
    char * serial,
    size_t bufferSize ) [pure virtual]
```

Retrieves general information about the camera at the specified index.

Parameters

<i>index</i>	The index at which to retrieve the camera information
<i>model</i>	Model description of the camera
<i>type</i>	Transport layer description of the camera
<i>serial</i>	Serial number of the camera
<i>bufferSize</i>	Size of the individual provided model, sensor, serial buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.16 GetSelectedCameraIndex()

```
STDMETHOD() GetSelectedCameraIndex (  
    unsigned int * selectedIndex ) [pure virtual]
```

Retrieves an integer index to the currently selected camera.

Parameters

<i>selectedIndex</i>	An integer that represents the index of the selected camera

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.17 IntegerGetInc()

```
STDMETHOD() IntegerGetInc (  
    const char * nodeName,  
    int64_t * increment ) [pure virtual]
```

Gets integer step increment for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>increment</i>	The step increment value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.18 IntegerGetIncMode()

```
STDMETHOD() IntegerGetIncMode (
    const char * nodeName,
    Spinnaker::GenApi::EIncMode * incMode ) [pure virtual]
```

Gets integer increment mode for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>incMode</i>	Spinnaker GenAPI increment mode for the specified feature

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.19 IntegerGetMax()

```
STDMETHOD() IntegerGetMax (
    const char * nodeName,
    int64_t * maxValve ) [pure virtual]
```

Gets maximum integer value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>maxValue</i>	The maximum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.20 IntegerGetMin()

```
STDMETHOD() IntegerGetMin (
    const char * nodeName,
    int64_t * minValve ) [pure virtual]
```

Gets minimum integer value allowed for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>minValue</i>	The minimum allowed value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.21 IntegerGetValue()

```
STDMETHOD() IntegerGetValue (
    const char * nodeName,
    int64_t * value ) [pure virtual]
```

Gets integer node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.22 IntegerSetValue()

```
STDMETHOD() IntegerSetValue (
    const char * nodeName,
    int64_t value ) [pure virtual]
```

Sets integer node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.23 IsStreaming()

```
STDMETHOD() IsStreaming (
    bool * isStreaming ) [pure virtual]
```

Checks if the selected camera is currently streaming.

Parameters

<i>isStreaming</i>	Flag indicating whether camera is streaming or not

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.24 NodeGetDisplayName()

```
STDMETHOD() NodeGetDisplayName (
    const char * nodeName,
    char * displayName,
    size_t bufferSize ) [pure virtual]
```

Gets a name string for display.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>displayName</i>	Display name of the device feature
<i>bufferSize</i>	Size of the provided displayName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.25 NodeGetType()

```
STDMETHOD() NodeGetType (
    const char * nodeName,
    char * typeName,
    size_t bufferSize ) [pure virtual]
```

Gets the type of the node.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>typeName</i>	The type of the node. Available type names are: Integer, Float, Boolean, String, Enumeration, Entry, Command and Other
<i>bufferSize</i>	Size of the provided typeName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.26 NodeIsAvailable()

```
STDMETHOD() NodeIsAvailable (
    const char * nodeName,
    bool * isAvailable ) [pure virtual]
```

Checks if a node is available.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isAvailable</i>	Flag indicating whether node is available or not

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.27 NodeIsImplemented()

```
STDMETHOD() NodeIsImplemented (
    const char * nodeName,
    bool * isImplemented ) [pure virtual]
```

Checks if a node is implemented.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isImplemented</i>	Flag indicating whether node is implemented or not

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.28 NodeIsReadable()

```
STDMETHOD() NodeIsReadable (
    const char * nodeName,
    bool * isReadable ) [pure virtual]
```

Checks if a node is readable.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isReadable</i>	Flag indicating whether node is readable or not

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.29 NodeIsWritable()

```
STDMETHOD() NodeIsWritable (
    const char * nodeName,
    bool * isWritable ) [pure virtual]
```

Checks if a node is writable.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>isWritable</i>	Flag indicating whether node is writable or not

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.30 NodeMapGetNodeAtIndex()

```
STDMETHOD() NodeMapGetNodeAtIndex (
    size_t index,
    char * nodeName,
    size_t bufferSize ) [pure virtual]
```

Retrieves the string representation of the node at the specified index.

Parameters

<i>index</i>	Node index in the NodeMap
<i>nodeName</i>	GenICam node name at specified index to be populated
<i>bufferSize</i>	Size of the provided nodeName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.31 NodeMapGetNumNodes()

```
STDMETHOD() NodeMapGetNumNodes (
    size_t * numNodes ) [pure virtual]
```

Retrieves the number of nodes available in the node map for the currently selected camera.

Parameters

<i>numNodes</i>	Number of nodes available for the currently selected device

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.32 NodeToString()

```
STDMETHOD() NodeToString (
    const char * nodeName,
    char * valueAsString,
    size_t bufferSize ) [pure virtual]
```

Gets content of the node as string.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>valueAsString</i>	String representation of the node value
<i>bufferSize</i>	Size of the provided valueAsString buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.33 SetSelectedCameraIndex()

```
STDMETHOD() SetSelectedCameraIndex (
    unsigned int index,
    bool needsRelease = true ) [pure virtual]
```

Sets the currently selected camera to the index specified.

This function will do nothing if the selected index is out of range.

Parameters

<i>index</i>	The index at which to retrieve the camera object
<i>needsRelease</i>	Specifies whether to release the previously selected camera object

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.34 StringGetValue()

```
STDMETHOD() StringGetValue (  
    const char * nodeName,  
    char * value,  
    size_t bufferSize ) [pure virtual]
```

Gets string node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value read from the device
<i>bufferSize</i>	Size of the provided value buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.1.35 StringSetValue()

```
STDMETHOD() StringSetValue (  
    const char * nodeName,  
    const char * value ) [pure virtual]
```

Sets string node value for the specified feature.

Parameters

<i>nodeName</i>	GenICam device feature name
<i>value</i>	The value to set to device

Returns

An HRESULT indicating the success or failure of the function.

16.148.2 Variable Documentation

16.148.2.1 bufferSize

```
const char unsigned int char size_t bufferSize = 0
```

16.148.2.2 entryExist

```
const char const char bool* entryExist = 0
```

16.148.2.3 entryIndex

```
const char unsigned int entryIndex
```

16.148.2.4 enumerationEntryName

```
const char unsigned int char * enumerationEntryName
```

16.148.2.5 enumerationName

```
const char * enumerationName
```

Checks if an enumeration entry exists for the specified enumeration feature.

Gets enumeration entry string for the specified enumeration feature at the specified index.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>enumerationEntryName</i>	GenICam device enumeration entry name to check
<i>entryExist</i>	Flag indicating whether the specified entry node exists for the enumeration node

Returns

An HRESULT indicating the success or failure of the function.

Parameters

<i>enumerationName</i>	GenICam device enumeration feature name
<i>entryIndex</i>	Enumeration entry index
<i>enumerationEntryName</i>	GenICam device enumeration entry name at specified entryIndex to be populated
<i>bufferSize</i>	Size of the provided enumerationEntryName buffer in bytes

Returns

An HRESULT indicating the success or failure of the function.

16.148.2.6 IID_ISpinnakerInterface

```
const GUID IID_ISpinnakerInterface [static]
```

Initial value:

```
= {0x5143ad36,
    0x51d1,
    0x422d,
    {0x9b, 0x17, 0x81, 0xdd, 0x3d, 0xb0, 0x60, 0xe}}
```

This is the Interface that allows users to get and set device node properties on the camera.

You can query the SpinnakerDirectShow Capture Filter for IID_ISpinnakerInterface and it will return a pointer to the ISpinnakerInterface interface. The GUID is equivalent to {5143AD36-51D1-422D-9B17-81DD3DB0600E}.

16.148.2.7 MAX_LENGTH

```
const unsigned int MAX_LENGTH = 256
```

16.149 include/SpinnakerPlatform.h File Reference**Macros**

- `#define SPINNAKER_API_ABSTRACT /*nothing*/`
- `#define SPINNAKER_API __attribute__((visibility("default")))`
- `#define SPINNAKER_LOCAL __attribute__((visibility("hidden")))`

16.149.1 Macro Definition Documentation**16.149.1.1 SPINNAKER_API**

```
#define SPINNAKER_API __attribute__((visibility("default")))
```

16.149.1.2 SPINNAKER_API_ABSTRACT

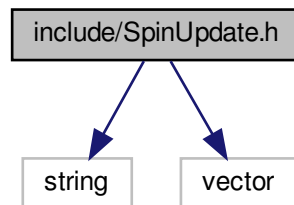
```
#define SPINNAKER_API_ABSTRACT /*nothing*/
```

16.149.1.3 SPINNAKER_LOCAL

```
#define SPINNAKER_LOCAL __attribute__((visibility("hidden")))
```

16.150 include/SpinUpdate.h File Reference

Include dependency graph for SpinUpdate.h:



Macros

- `#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT`

Functions

- `SPINUPDATE_API int UpdateFirmwareConsole (unsigned int numArgs, char **argList)`
Updates the firmware for the device.
- `SPINUPDATE_API int UpdateFirmwareGUI (std::string args)`
- `SPINUPDATE_API int UpdateFirmware (std::vector< std::string > args)`
- `SPINUPDATE_API void SetMessageCallback (UpdaterMessageCallback messageCallbackFunction)`
- `SPINUPDATE_API void SetProgressCallback (UpdaterProgressCallback progressCallbackFunction)`
- `const SPINUPDATE_API char * GetErrorMessage ()`

Variables

- `SPINUPDATE_API typedef int(* UpdaterMessageCallback )(const char *message)`
- `SPINUPDATE_API typedef int(* UpdaterProgressCallback )(const char *action, unsigned int address, int globalPercent, int currPercent)`

16.150.1 Macro Definition Documentation

16.150.1.1 SPINUPDATE_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```

16.150.2 Function Documentation

16.150.2.1 GetErrorMessage()

```
const SPINUPDATE_API char* GetErrorMessage ( )
```

16.150.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (
    UpdatorMessageCallback messageCallbackFunction )
```

16.150.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (
    UpdatorProgressCallback progressCallbackFunction )
```

16.150.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
    std::vector< std::string > args )
```

16.150.2.5 UpdateFirmwareConsole()

```
SPINUPDATE_API int UpdateFirmwareConsole (
    unsigned int numArgs,
    char ** argList )
```

Updates the firmware for the device.

Parameters

<i>numArgs</i>	Number of strings pointed to by argv
<i>argList</i>	Pointer to list of string options for the firmware update

Returns

0 for success, otherwise non zero for failures.

Typical usage for updating is as follows: -R{serial number} [-{options} ..] {firmware zim file} -R{serial number} -UU -B {firmware zim file}

Option definitions: -B = Reboots the camera after the update has completed. If this argument is not provided, a manual power cycle will be required. -A = Updates individual portions of the firmware in flash. The code section of camera at location 0xFF08000 will be updated. ./sample_app -AFF08000 camera.zim -U = Downgrade the firmware. Multiple U's can be used to overwrite the ROM header. -F = Force program and EEprom reload. -R = Enter a regular expression for camera serial match. For example: ./sample_app -R.* camera.zim Results in matching any camera serial -P = Checks the progress of the updater. -epromsave = Save the content of the EEprom to a file.

16.150.2.6 UpdateFirmwareGUI()

```
SPINUPDATE_API int UpdateFirmwareGUI (
    std::string args )
```

16.150.3 Variable Documentation

16.150.3.1 UpdatorMessageCallback

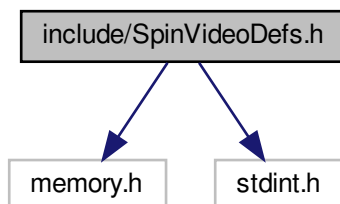
```
SPINUPDATE_API typedef int(* UpdatorMessageCallback) (const char *message)
```

16.150.3.2 UpdatorProgressCallback

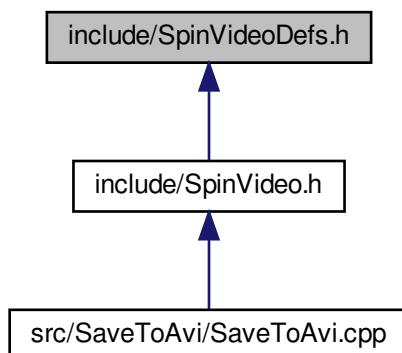
```
SPINUPDATE_API typedef int( * UpdatorProgressCallback) (const char *action, unsigned int address,
int globalPercent, int currPercent)
```


16.152 include/SpinVideoDefs.h File Reference

Include dependency graph for SpinVideoDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

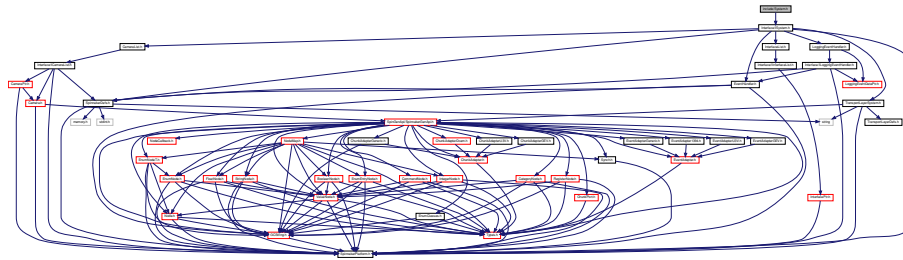
- struct [MJPEGOption](#)
Options for saving MJPG files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [AVIOption](#)
Options for saving AVI files.

Namespaces

- [Spinnaker](#)
- [Spinnaker::Video](#)

16.153 include/System.h File Reference

Include dependency graph for System.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [System](#)

The system object is used to retrieve the list of interfaces and cameras available.

Namespaces

- [Spinnaker](#)

Macros

- `#define FLIR_SPINNAKER_VERSION_MAJOR 3`
- `#define FLIR_SPINNAKER_VERSION_MINOR 0`
- `#define FLIR_SPINNAKER_VERSION_TYPE 0`
- `#define FLIR_SPINNAKER_VERSION_BUILD 118`

16.153.1 Macro Definition Documentation

16.153.1.1 FLIR_SPINNAKER_VERSION_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 118
```

16.153.1.2 FLIR_SPINNAKER_VERSION_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 3
```

16.153.1.3 FLIR_SPINNAKER_VERSION_MINOR

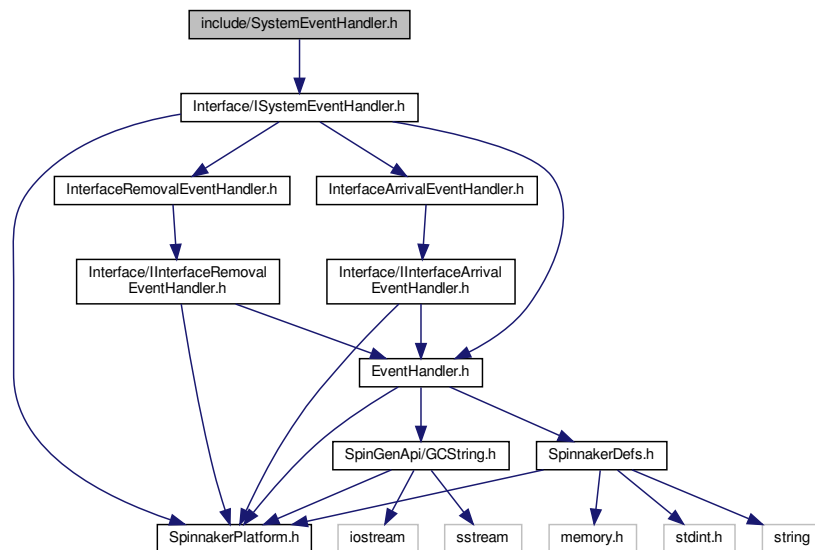
```
#define FLIR_SPINNAKER_VERSION_MINOR 0
```

16.153.1.4 FLIR_SPINNAKER_VERSION_TYPE

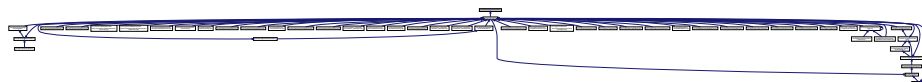
```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

16.154 include/SystemEventHandler.h File Reference

Include dependency graph for SystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



Classes

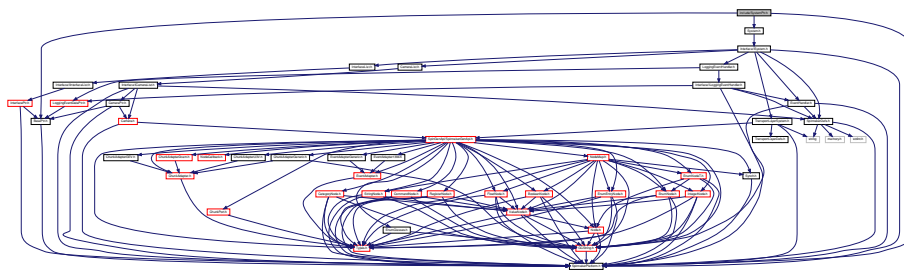
- class [SystemEventHandler](#)
A handler to interface arrival and removal events on the system.

Namespaces

- [Spinnaker](#)

16.155 include/SystemPtr.h File Reference

Include dependency graph for SystemPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

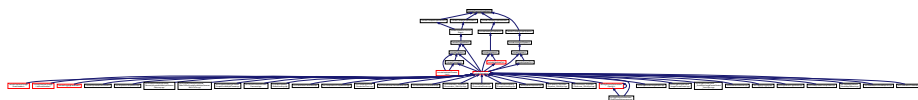
- class [SystemPtr](#)
A reference tracked pointer to a system object.

Namespaces

- [Spinnaker](#)

16.156 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)

Enumerations

- enum [StreamTypeEnum](#) {
[StreamType_GigEVision](#),
[StreamType_CameraLink](#),
[StreamType_CameraLinkHS](#),
[StreamType_CoaXPress](#),
[StreamType_USB3Vision](#),
[StreamType_Custom](#),
[NUMSTREAMTYPE](#) }

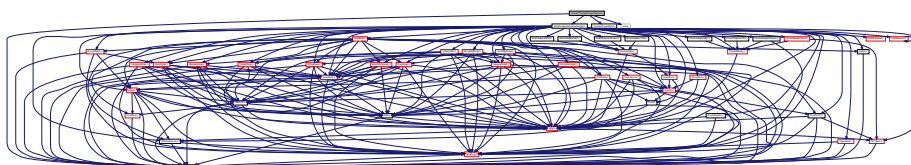
The enum definitions for TL Device nodes from the transport layer .xml files.

- enum [StreamModeEnum](#) {
[StreamMode_Socket](#),
[StreamMode_LWF](#),
[StreamMode_MVA](#),
[NUMSTREAMMODE](#) }
- enum [StreamBufferCountModeEnum](#) {
[StreamBufferCountMode_Manual](#),
[NUMSTREAMBUFFERCOUNTMODE](#) }
- enum [StreamBufferHandlingModeEnum](#) {
[StreamBufferHandlingMode_OldestFirst](#),
[StreamBufferHandlingMode_OldestFirstOverwrite](#),
[StreamBufferHandlingMode_NewestOnly](#),
[StreamBufferHandlingMode_NewestFirst](#),
[NUMSTREAMBUFFERHANDLINGMODE](#) }
- enum [DeviceTypeEnum](#) {
[DeviceType_GigEVision](#),
[DeviceType_CameraLink](#),
[DeviceType_CameraLinkHS](#),
[DeviceType_CoaXPress](#),
[DeviceType_USB3Vision](#),
[DeviceType_Custom](#),
[NUMDEVICETYPE](#) }
- enum [DeviceAccessStatusEnum](#) {
[DeviceAccessStatus_Unknown](#),
[DeviceAccessStatus_ReadWrite](#),
[DeviceAccessStatus_ReadOnly](#),
[DeviceAccessStatus_NoAccess](#),
[DeviceAccessStatus_Busy](#),
[DeviceAccessStatus_OpenReadWrite](#),
[DeviceAccessStatus_OpenReadOnly](#),
[NUMDEVICEACCESSSTATUS](#) }
- enum [GenICamXMLLocationEnum](#) {
[GenICamXMLLocation_Device](#),
[GenICamXMLLocation_Host](#),
[NUMGENICAMXMLLOCATION](#) }
- enum [GUIXMLLocationEnum](#) {
[GUIXMLLocation_Device](#),
[GUIXMLLocation_Host](#),
[NUMGUIXMLLOCATION](#) }

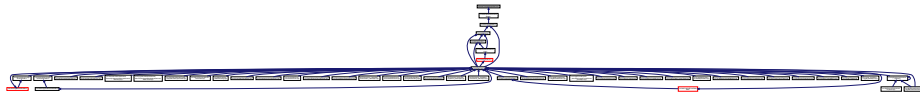
- enum `GevCCPEnum` {
`GevCCP_EnumEntry_GevCCP_OpenAccess`,
`GevCCP_EnumEntry_GevCCP_ExclusiveAccess`,
`GevCCP_EnumEntry_GevCCP_ControlAccess`,
`NUMGEVCCP` }
- enum `DeviceEndiannessMechanismEnum` {
`DeviceEndiannessMechanism_Legacy`,
`DeviceEndiannessMechanism_Standard`,
`NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {
`DeviceCurrentSpeed_UnknownSpeed`,
`DeviceCurrentSpeed_LowSpeed`,
`DeviceCurrentSpeed_FullSpeed`,
`DeviceCurrentSpeed_HighSpeed`,
`DeviceCurrentSpeed_SuperSpeed`,
`NUMDEVICECURRENTSPEED` }
- enum `InterfaceTypeEnum` {
`InterfaceType_GigEVision`,
`InterfaceType_CameraLink`,
`InterfaceType_CameraLinkHS`,
`InterfaceType_CoaXPress`,
`InterfaceType_USB3Vision`,
`InterfaceType_Custom`,
`NUMINTERFACETYPE` }
- enum `POEStatusEnum` {
`POEStatus_NotSupported`,
`POEStatus_PowerOff`,
`POEStatus_PowerOn`,
`NUMPOESTATUS` }
- enum `FilterDriverStatusEnum` {
`FilterDriverStatus_NotSupported`,
`FilterDriverStatus_Disabled`,
`FilterDriverStatus_Enabled`,
`NUMFILTERDRIVERSTATUS` }
- enum `TLTypeEnum` {
`TLType_GigEVision`,
`TLType_CameraLink`,
`TLType_CameraLinkHS`,
`TLType_CoaXPress`,
`TLType_USB3Vision`,
`TLType_Mixed`,
`TLType_Custom`,
`NUMTLTYPE` }

16.157 include/TransportLayerDevice.h File Reference

Include dependency graph for TransportLayerDevice.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerDevice](#)

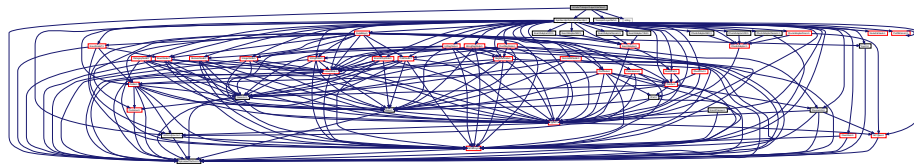
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

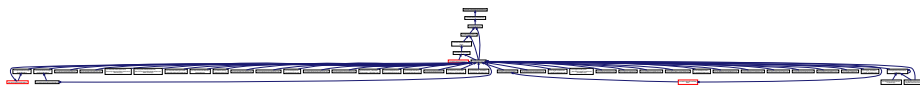
- [Spinnaker](#)

16.158 include/TransportLayerInterface.h File Reference

Include dependency graph for TransportLayerInterface.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerInterface](#)

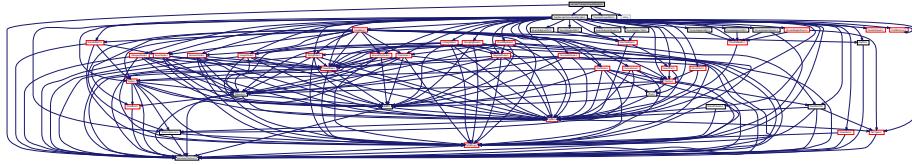
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

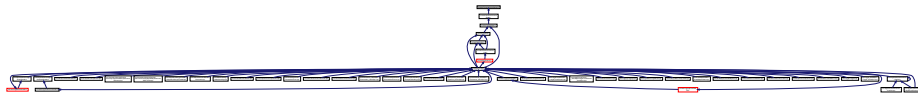
- [Spinnaker](#)

16.159 include/TransportLayerStream.h File Reference

Include dependency graph for TransportLayerStream.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerStream](#)

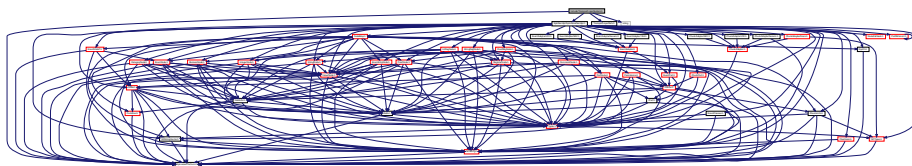
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

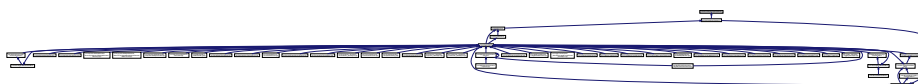
- [Spinnaker](#)

16.160 include/TransportLayerSystem.h File Reference

Include dependency graph for TransportLayerSystem.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [TransportLayerSystem](#)

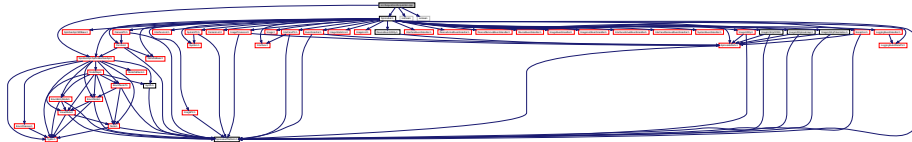
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

- [Spinnaker](#)

16.161 src/Acquisition/Acquisition.cpp File Reference

Include dependency graph for Acquisition.cpp:



Functions

- int [ConfigureGVCPHeartbeat](#) ([CameraPtr](#) pCam, bool enable)
- int [ResetGVCPHeartbeat](#) ([CameraPtr](#) pCam)
- int [DisableGVCPHeartbeat](#) ([CameraPtr](#) pCam)
- int [AcquireImages](#) ([CameraPtr](#) pCam, [INodeMap](#) &nodeMap, [INodeMap](#) &nodeMapTLDevice)
- int [PrintDeviceInfo](#) ([INodeMap](#) &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.161.1 Function Documentation

16.161.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.161.1.2 ConfigureGVCPHeartbeat()

```
int ConfigureGVCPHeartbeat (
    CameraPtr pCam,
    bool enable )
```

16.161.1.3 DisableGVCPHeartbeat()

```
int DisableGVCPHeartbeat (
    CameraPtr pCam )
```

16.161.1.4 main()

```
int main (
    int ,
    char ** )
```

16.161.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.161.1.6 ResetGVCPHeartbeat()

```
int ResetGVCPHeartbeat (
    CameraPtr pCam )
```

16.161.1.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

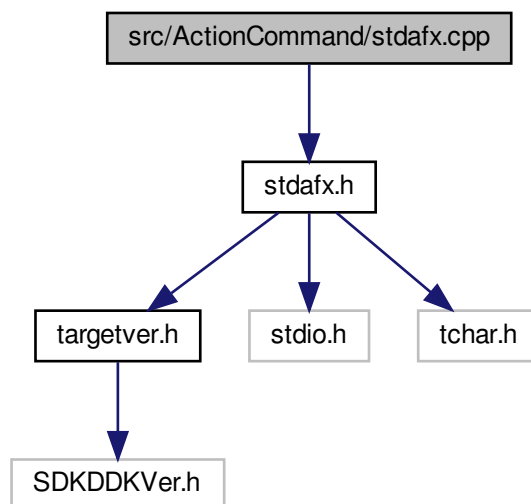

- 16.162 [src/Acquisition/resource.h](#) File Reference**
- 16.163 [src/AcquisitionMultipleCameraRecovery/resource.h](#) File Reference**
- 16.164 [src/AcquisitionMultipleCamerasWriteToFile/resource.h](#) File Reference**
- 16.165 [src/AcquisitionMultipleThread/resource.h](#) File Reference**
- 16.166 [src/ActionCommand/resource.h](#) File Reference**
- 16.167 [src/BufferHandling/resource.h](#) File Reference**
- 16.168 [src/ChunkData/resource.h](#) File Reference**
- 16.169 [src/Compression/resource.h](#) File Reference**
- 16.170 [src/CounterAndTimer/resource.h](#) File Reference**
- 16.171 [src/DeviceEvents/resource.h](#) File Reference**
- 16.172 [src/Enumeration/resource.h](#) File Reference**
- 16.173 [src/Enumeration_QuickSpin/resource.h](#) File Reference**
- 16.174 [src/EnumerationEvents/resource.h](#) File Reference**
- 16.175 [src/ExceptionHandling/resource.h](#) File Reference**
- 16.176 [src/Exposure/resource.h](#) File Reference**
- 16.177 [src/Exposure_QuickSpin/resource.h](#) File Reference**
- 16.178 [src/FileAccess_QuickSpin/resource.h](#) File Reference**
- 16.179 [src/GigEVisionPerformance/resource.h](#) File Reference**
- 16.180 [src/ImageEvents/resource.h](#) File Reference**
- 16.181 [src/ImageFormatControl/resource.h](#) File Reference**
- 16.182 [src/ImageFormatControl_QuickSpin/resource.h](#) File Reference**

-
- 16.183 [src/Inference/resource.h](#) File Reference**

- 16.184 [src/Logging/resource.h](#) File Reference**

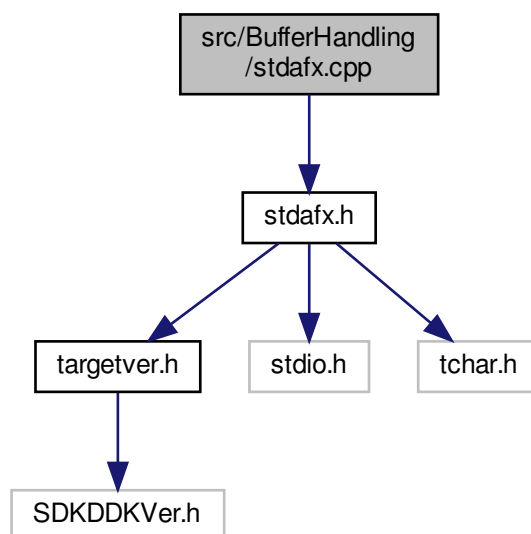
16.196 src/ActionCommand/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



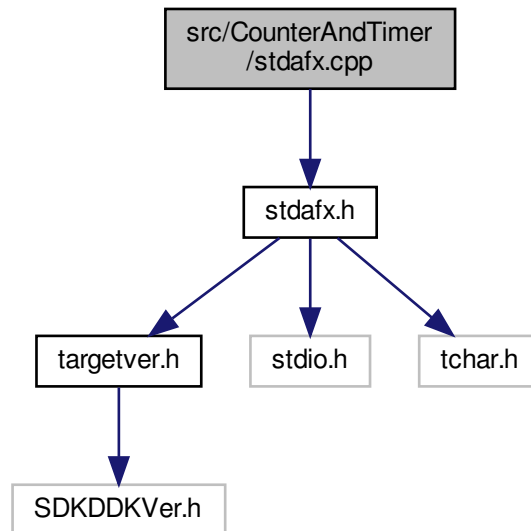
16.197 src/BufferHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



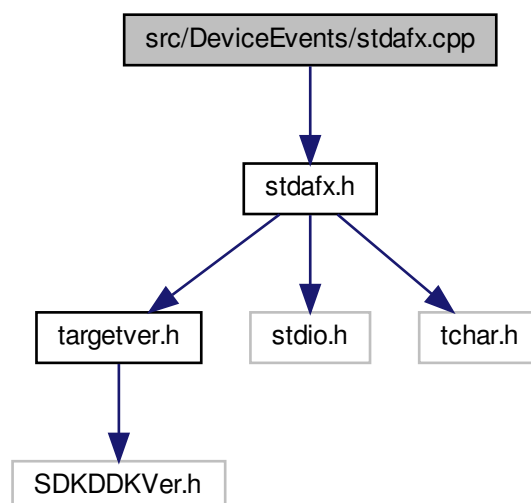
16.198 src/CounterAndTimer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



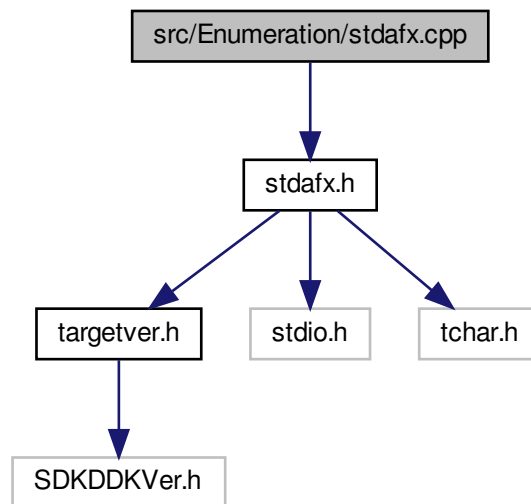
16.199 src/DeviceEvents/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



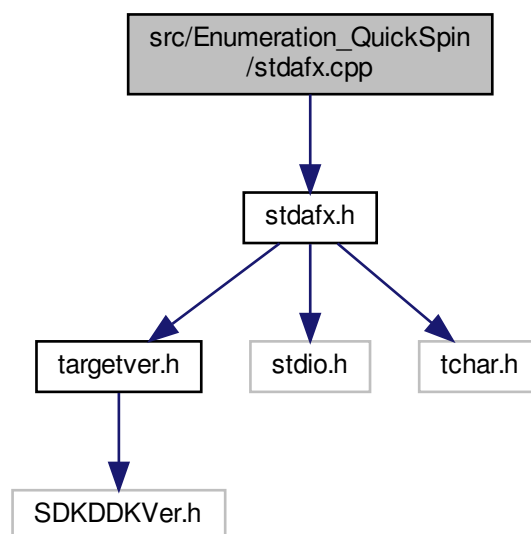
16.200 src/Enumeration/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



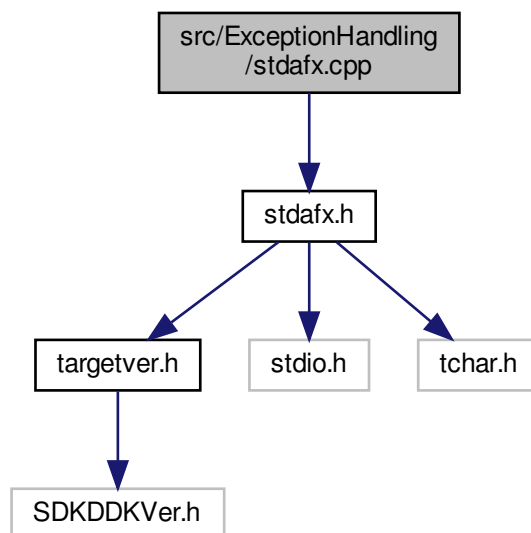
16.201 src/Enumeration_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



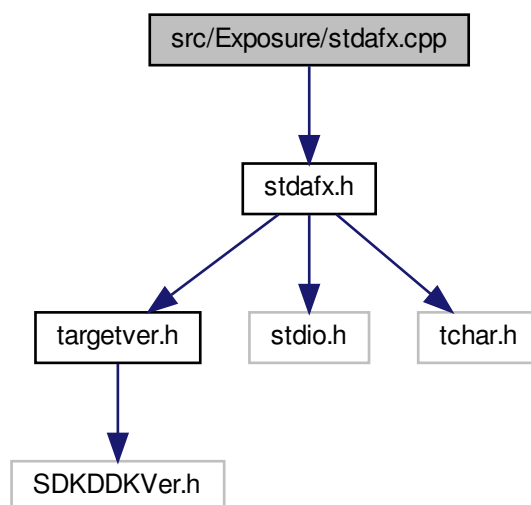
16.202 src/ExceptionHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



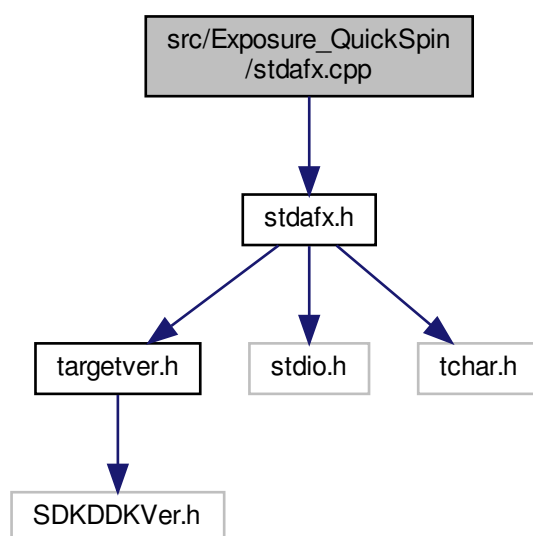
16.203 src/Exposure/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



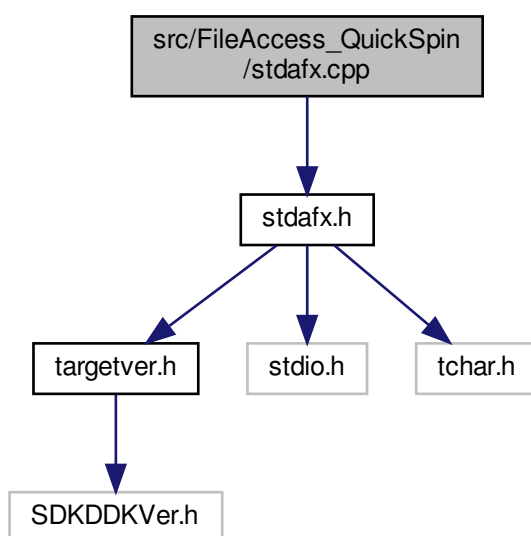
16.204 src/Exposure_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



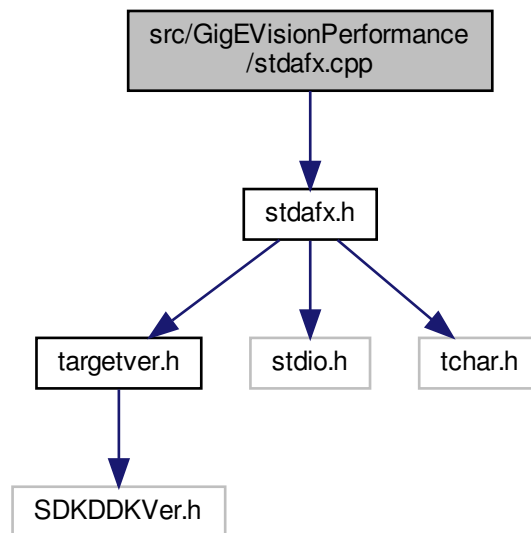
16.205 src/FileAccess_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



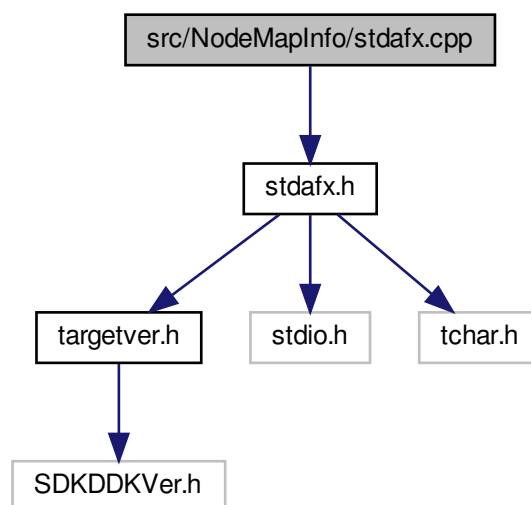
16.206 src/GigEVisionPerformance/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



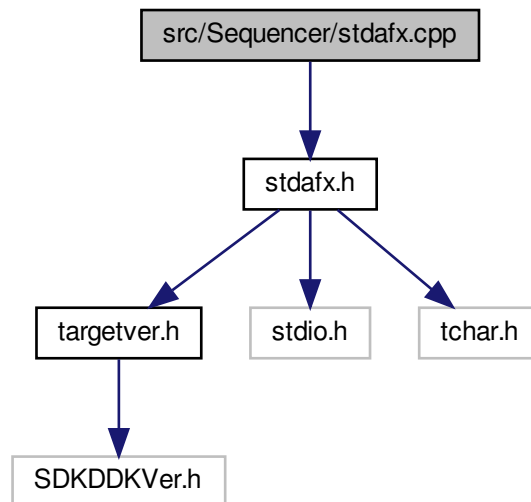
16.207 src/NodeMapInfo/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



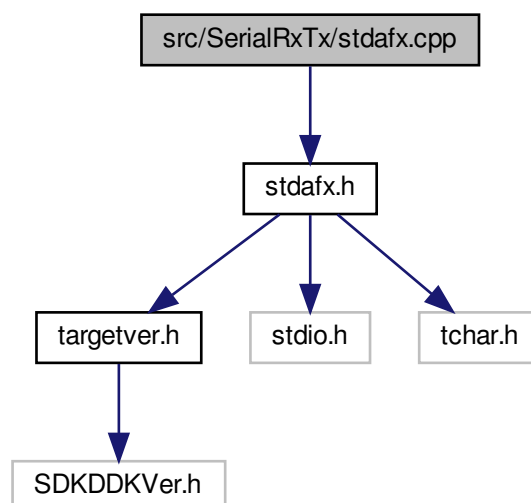
16.208 src/Sequencer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



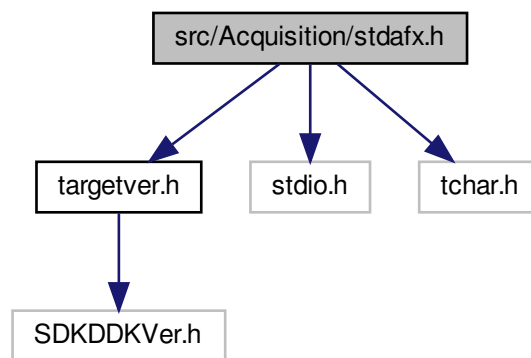
16.209 src/SerialRxTx/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

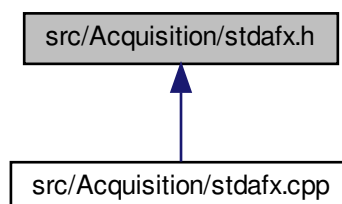


16.210 src/Acquisition/stdafx.h File Reference

Include dependency graph for stdafx.h:

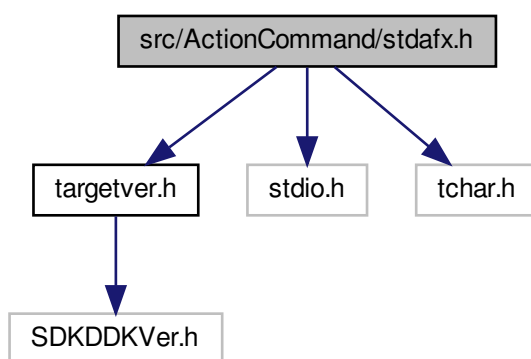


This graph shows which files directly or indirectly include this file:

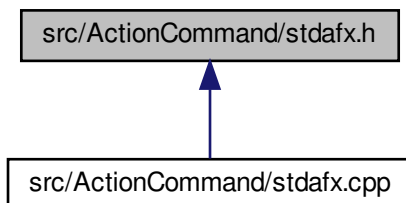


16.211 src/ActionCommand/stdafx.h File Reference

Include dependency graph for stdafx.h:

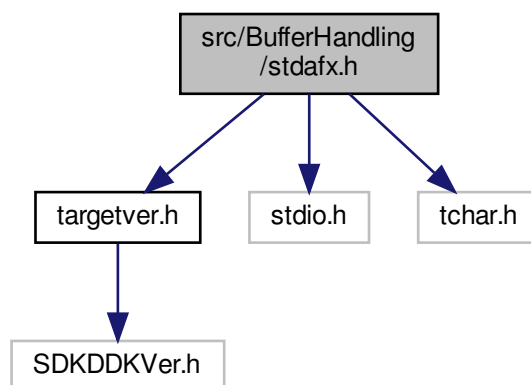


This graph shows which files directly or indirectly include this file:

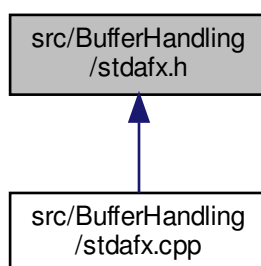


16.212 src/BufferHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

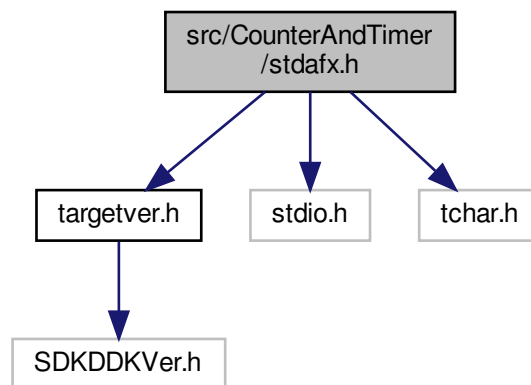


This graph shows which files directly or indirectly include this file:

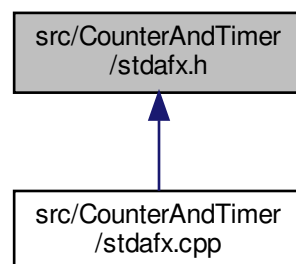


16.213 src/CounterAndTimer/stdafx.h File Reference

Include dependency graph for stdafx.h:

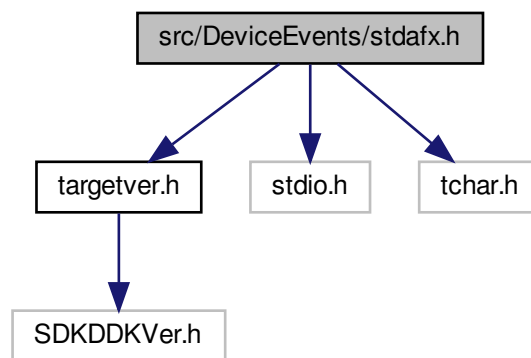


This graph shows which files directly or indirectly include this file:

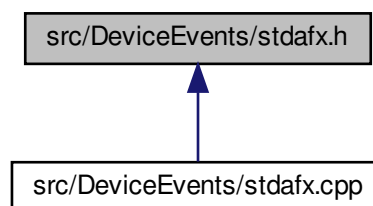


16.214 src/DeviceEvents/stdafx.h File Reference

Include dependency graph for stdafx.h:

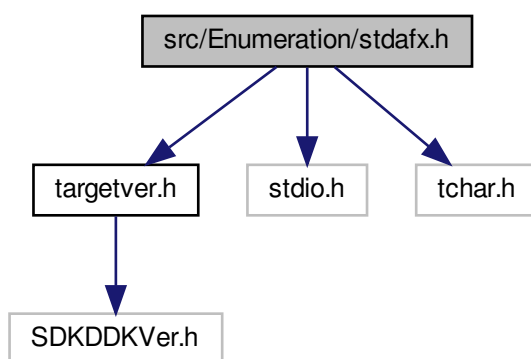


This graph shows which files directly or indirectly include this file:

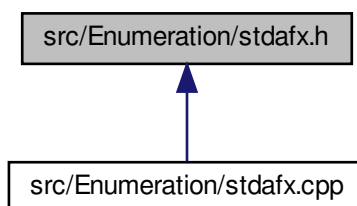


16.215 src/Enumeration/stdafx.h File Reference

Include dependency graph for stdafx.h:

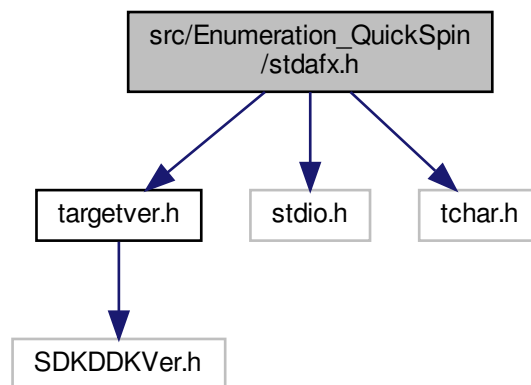


This graph shows which files directly or indirectly include this file:

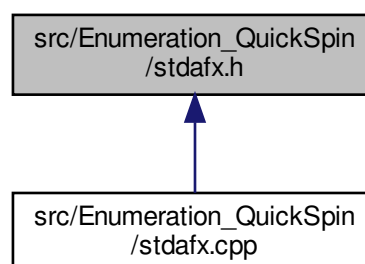


16.216 src/Enumeration_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

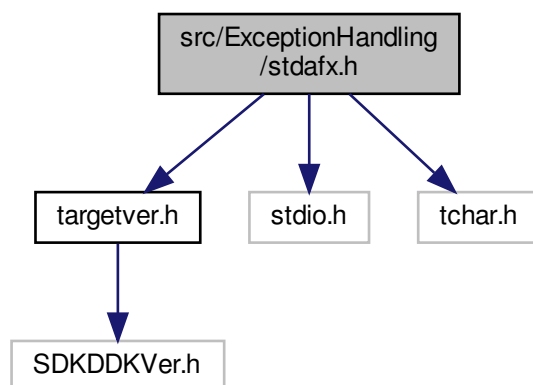


This graph shows which files directly or indirectly include this file:

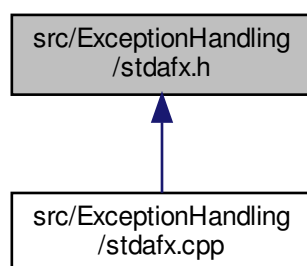


16.217 src/ExceptionHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

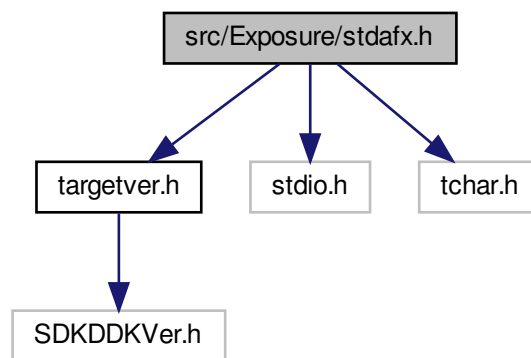


This graph shows which files directly or indirectly include this file:

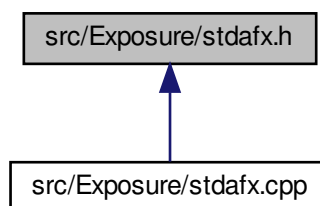


16.218 src/Exposure/stdafx.h File Reference

Include dependency graph for stdafx.h:

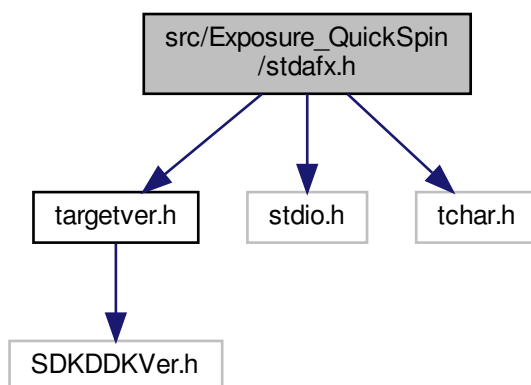


This graph shows which files directly or indirectly include this file:

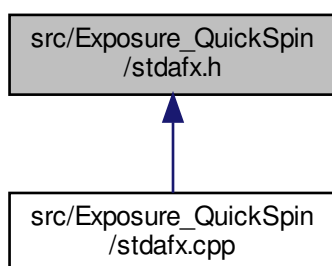


16.219 src/Exposure_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

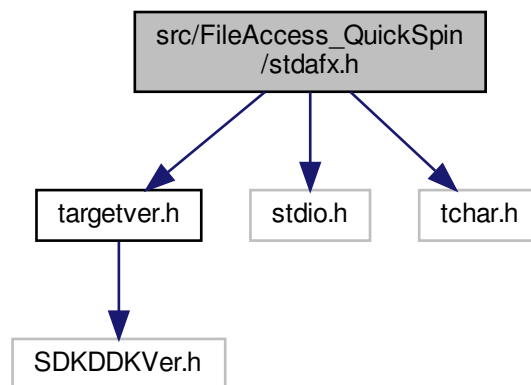


This graph shows which files directly or indirectly include this file:

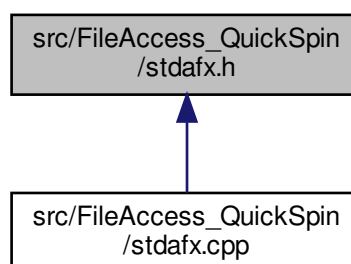


16.220 src/FileAccess_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

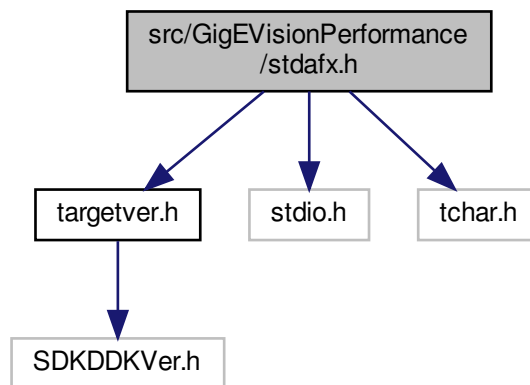


This graph shows which files directly or indirectly include this file:

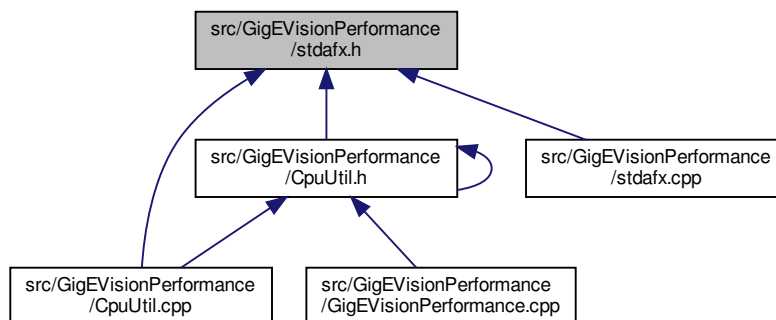


16.221 src/GigEVisionPerformance/stdafx.h File Reference

Include dependency graph for stdafx.h:

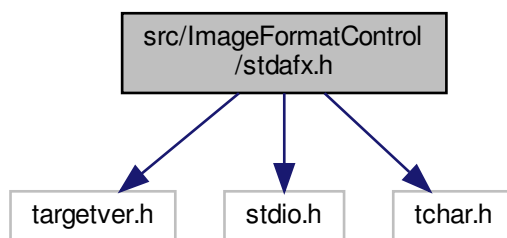


This graph shows which files directly or indirectly include this file:



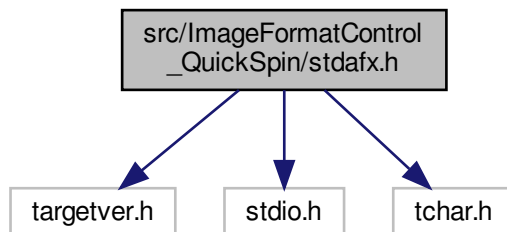
16.222 src/ImageFormatControl/stdafx.h File Reference

Include dependency graph for stdafx.h:



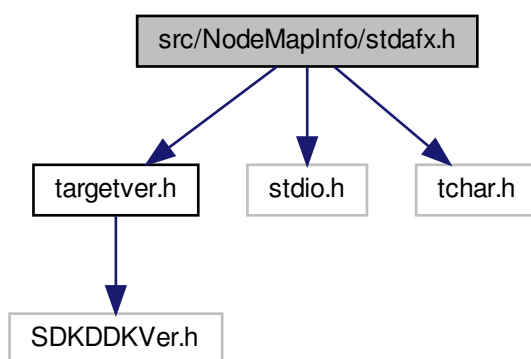
16.223 src/ImageFormatControl_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

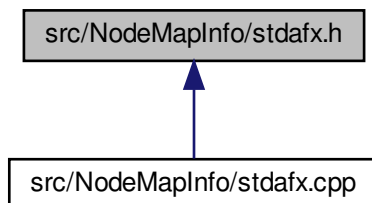


16.224 src/NodeMapInfo/stdafx.h File Reference

Include dependency graph for stdafx.h:

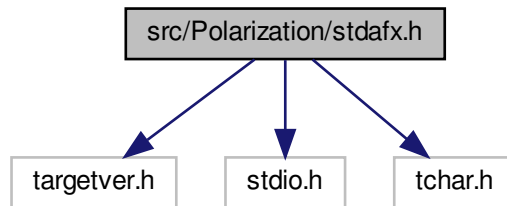


This graph shows which files directly or indirectly include this file:



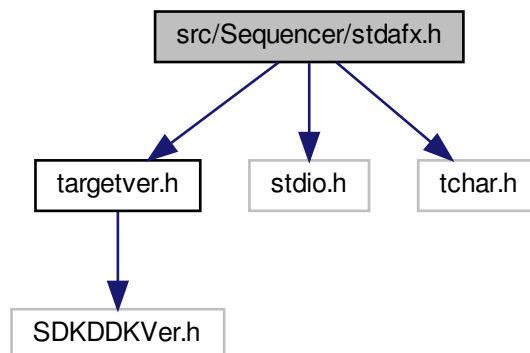
16.225 src/Polarization/stdafx.h File Reference

Include dependency graph for stdafx.h:

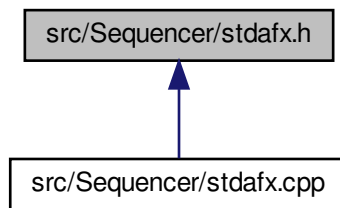


16.226 src/Sequencer/stdafx.h File Reference

Include dependency graph for stdafx.h:

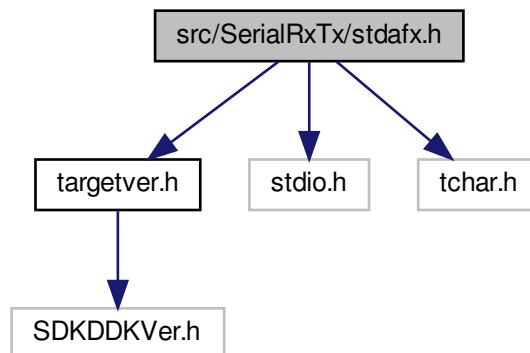


This graph shows which files directly or indirectly include this file:

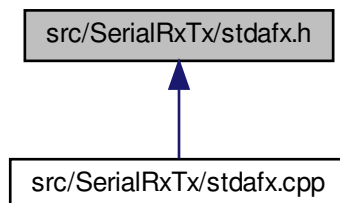


16.227 src/SerialRxTx/stdafx.h File Reference

Include dependency graph for stdafx.h:

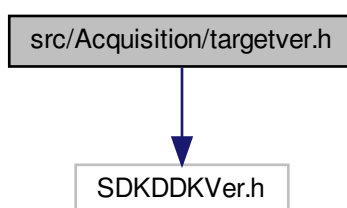


This graph shows which files directly or indirectly include this file:

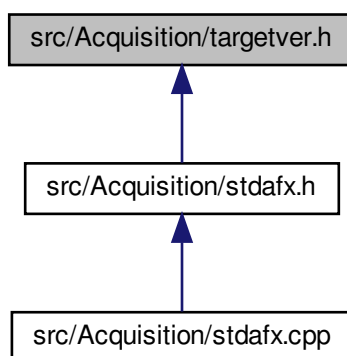


16.228 src/Acquisition/targetver.h File Reference

Include dependency graph for targetver.h:

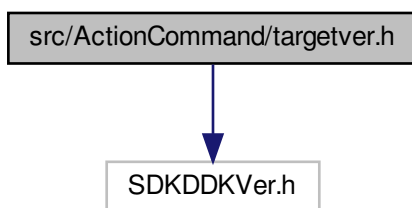


This graph shows which files directly or indirectly include this file:

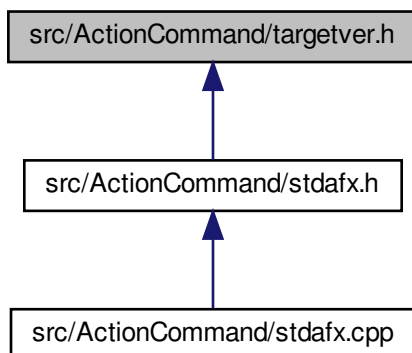


16.229 src/ActionCommand/targetver.h File Reference

Include dependency graph for targetver.h:

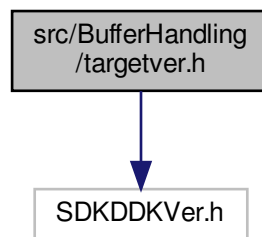


This graph shows which files directly or indirectly include this file:

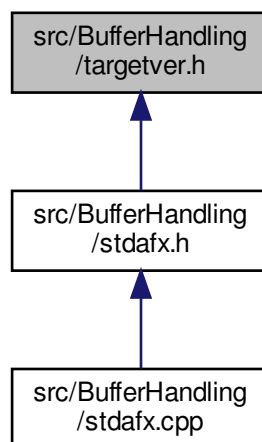


16.230 src/BufferHandling/targetver.h File Reference

Include dependency graph for targetver.h:

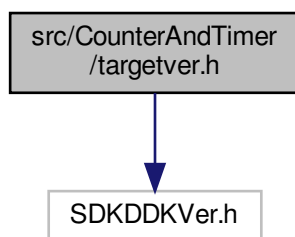


This graph shows which files directly or indirectly include this file:

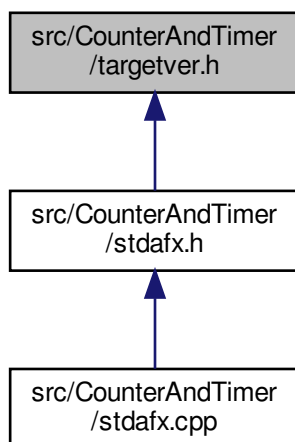


16.231 src/CounterAndTimer/targetver.h File Reference

Include dependency graph for targetver.h:

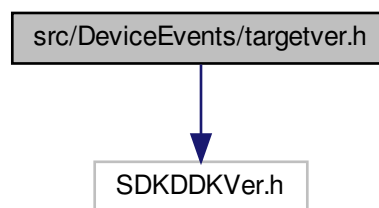


This graph shows which files directly or indirectly include this file:

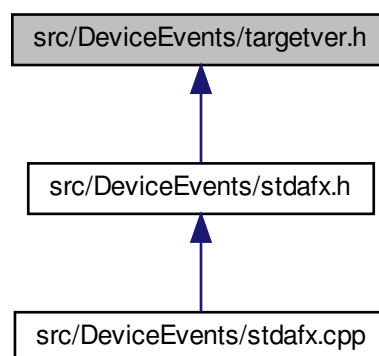


16.232 src/DeviceEvents/targetver.h File Reference

Include dependency graph for targetver.h:

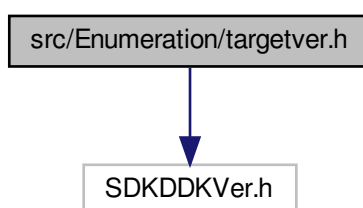


This graph shows which files directly or indirectly include this file:

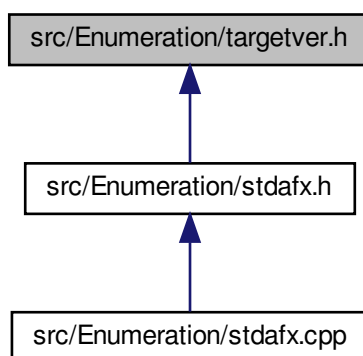


16.233 src/Enumeration/targetver.h File Reference

Include dependency graph for targetver.h:

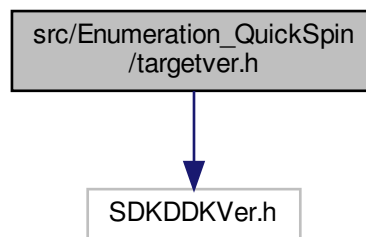


This graph shows which files directly or indirectly include this file:

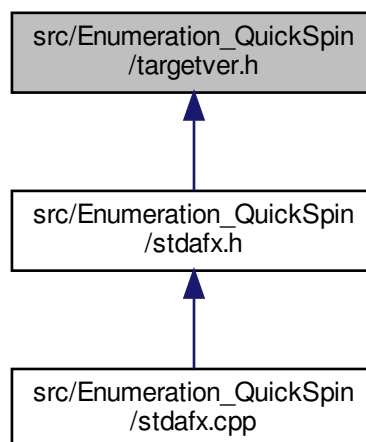


16.234 src/Enumeration_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

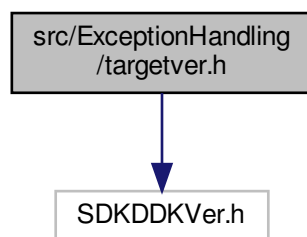


This graph shows which files directly or indirectly include this file:

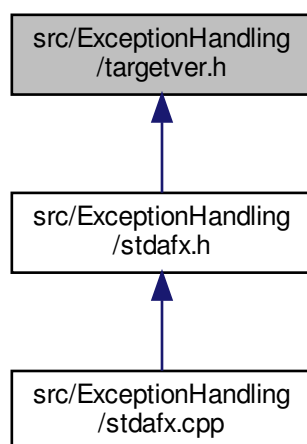


16.235 src/ExceptionHandling/targetver.h File Reference

Include dependency graph for targetver.h:

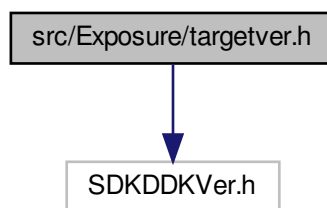


This graph shows which files directly or indirectly include this file:

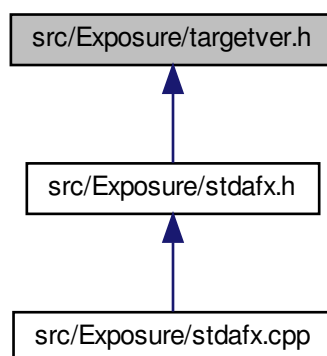


16.236 src/Exposure/targetver.h File Reference

Include dependency graph for targetver.h:

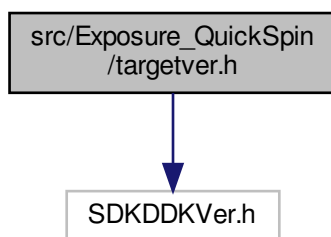


This graph shows which files directly or indirectly include this file:

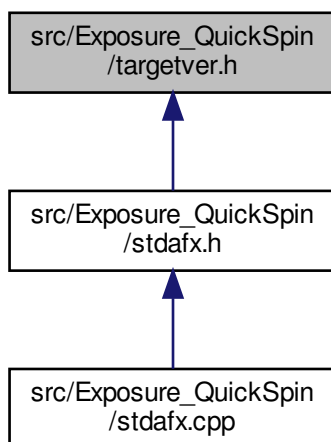


16.237 src/Exposure_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

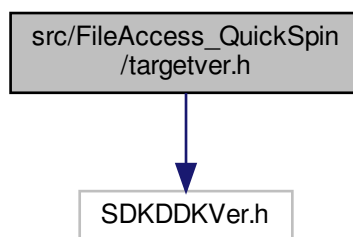


This graph shows which files directly or indirectly include this file:

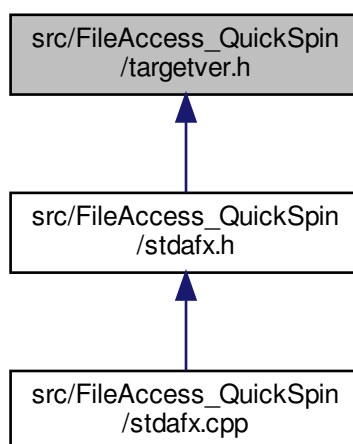


16.238 src/FileAccess_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

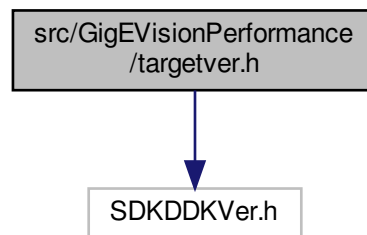


This graph shows which files directly or indirectly include this file:

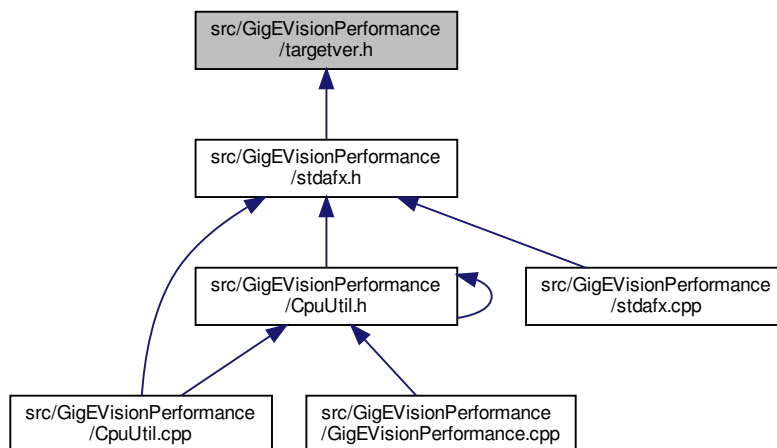


16.239 src/GigEVisionPerformance/targetver.h File Reference

Include dependency graph for targetver.h:

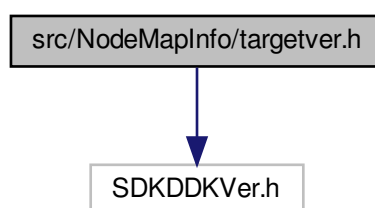


This graph shows which files directly or indirectly include this file:

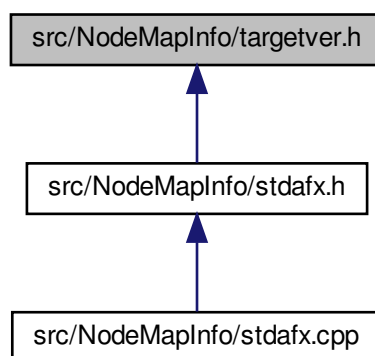


16.240 src/NodeMapInfo/targetver.h File Reference

Include dependency graph for targetver.h:

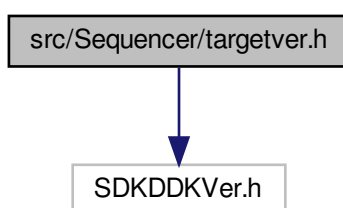


This graph shows which files directly or indirectly include this file:

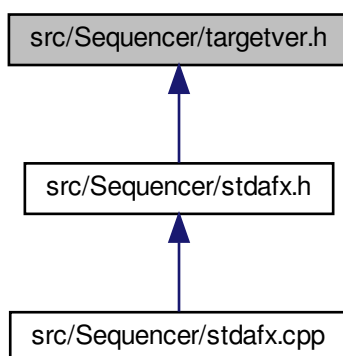


16.241 src/Sequencer/targetver.h File Reference

Include dependency graph for targetver.h:

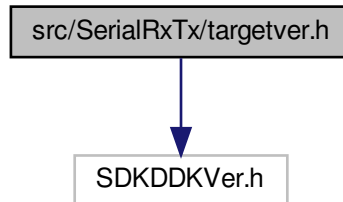


This graph shows which files directly or indirectly include this file:

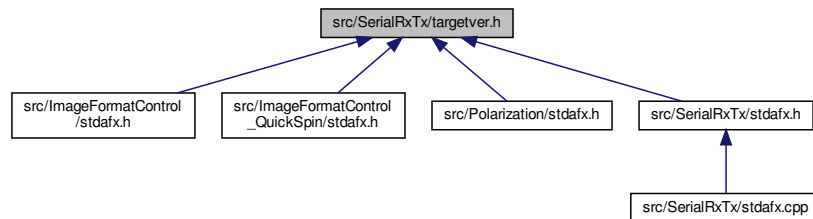


16.242 src/SerialRxTx/targetver.h File Reference

Include dependency graph for targetver.h:

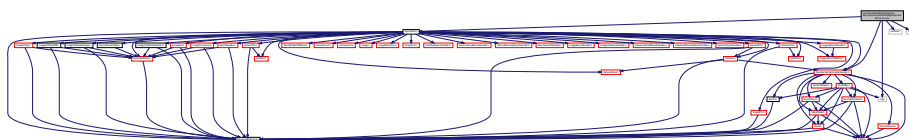


This graph shows which files directly or indirectly include this file:



16.243 src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference

Include dependency graph for AcquisitionMultipleCameraRecovery.cpp:



Classes

- class ImageEventHandlerImpl
- struct GrabInfo
- class InterfaceEventHandlerImpl

Functions

- void [SleepyWrapper](#) (int milliseconds)
- void [RefreshCameraList](#) ([SystemPtr](#) system)
- bool [ConfigureCamera](#) ([CameraPtr](#) pCam)
- bool [ConfigureUserSet1](#) ([CameraPtr](#) pCam)
- void [ResetCameraUserSetToDefault](#) ([CameraPtr](#) pCam)
- string [GetDeviceSerial](#) ([CameraPtr](#) pCam)
- void [PrintExampleStatistics](#) ()
- int [main](#) (int, char **)

Variables

- std::map< std::string, [GrabInfo](#) > [cameraGrabInfoMap](#)
- [CameraList](#) [globalCamList](#)

16.243.1 Function Documentation

16.243.1.1 [ConfigureCamera\(\)](#)

```
bool ConfigureCamera (  
    CameraPtr pCam )
```

16.243.1.2 [ConfigureUserSet1\(\)](#)

```
bool ConfigureUserSet1 (  
    CameraPtr pCam )
```

16.243.1.3 [GetDeviceSerial\(\)](#)

```
string GetDeviceSerial (  
    CameraPtr pCam )
```

16.243.1.4 [main\(\)](#)

```
int main (  
    int ,  
    char ** )
```

16.243.1.5 PrintExampleStatistics()

```
void PrintExampleStatistics ( )
```

16.243.1.6 RefreshCameraList()

```
void RefreshCameraList (
    SystemPtr system )
```

16.243.1.7 ResetCameraUserSetToDefault()

```
void ResetCameraUserSetToDefault (
    CameraPtr pCam )
```

16.243.1.8 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.243.2 Variable Documentation

16.243.2.1 cameraGrabInfoMap

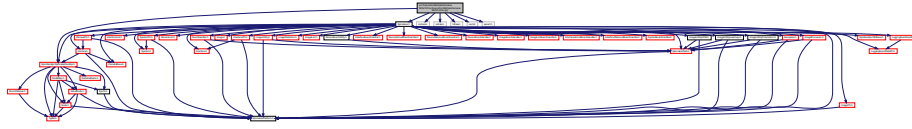
```
std::map<std::string, GrabInfo> cameraGrabInfoMap
```

16.243.2.2 globalCamList

```
CameraList globalCamList
```

16.244 src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.cpp File Reference

Include dependency graph for AcquisitionMultipleCamerasWriteToFile.cpp:



Classes

- struct [ImageInfo](#)

Functions

- bool [CreateFiles](#) (unsigned int numCameras)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- bool [ConfigureCameras](#) ([CameraList](#) &camList, unsigned int numCameras)
- bool [AcquireImagesAndSaveToFile](#) ([CameraList](#) &camList, unsigned int numCameras)
- bool [RetrieveImagesFromFiles](#) (unsigned int numCameras, string fileFormat="bmp")
- int [RunCameras](#) ([CameraList](#) &camList, unsigned int numCameras)
- int [main](#) (int, char **)

Variables

- const string [kDestinationDirectory](#) = ""
- const unsigned int [k_numImages](#) = 30
- vector< [ImageInfo](#) > [imageInfos](#)

16.244.1 Function Documentation

16.244.1.1 AcquireImagesAndSaveToFile()

```
bool AcquireImagesAndSaveToFile (
    CameraList & camList,
    unsigned int numCameras )
```

16.244.1.2 ConfigureCameras()

```
bool ConfigureCameras (
    CameraList & camList,
    unsigned int numCameras )
```

16.244.1.3 CreateFiles()

```
bool CreateFiles (
    unsigned int numCameras )
```

16.244.1.4 main()

```
int main (
    int ,
    char ** )
```

16.244.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.244.1.6 RetrievalImagesFromFiles()

```
bool RetrieveImagesFromFiles (
    unsigned int numCameras,
    string fileFormat = "bmp" )
```

16.244.1.7 RunCameras()

```
int RunCameras (
    CameraList & camList,
    unsigned int numCameras )
```

16.244.2 Variable Documentation

16.244.2.1 imageInfos

```
vector<ImageInfo> imageInfos
```

16.244.2.2 k_numImages

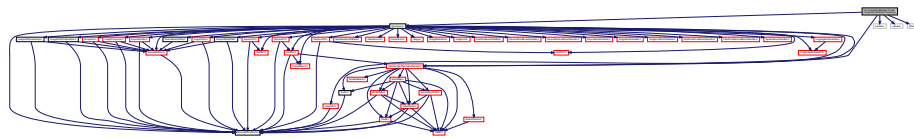
```
const unsigned int k_numImages = 30
```

16.244.2.3 kDestinationDirectory

```
const string kDestinationDirectory = ""
```

16.245 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference

Include dependency graph for AcquisitionMultipleThread.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap, std::string camSerial)
- int [ConfigureGVCPHeartbeat](#) (CameraPtr pCam, bool enable)
- int [ResetGVCPHeartbeat](#) (CameraPtr pCam)
- int [DisableGVCPHeartbeat](#) (CameraPtr pCam)
- void * [AcquireImages](#) (void *arg)
- int [RunMultipleCameras](#) (CameraList camList)
- int [main](#) (int, char **)

16.245.1 Function Documentation

16.245.1.1 AcquireImages()

```
void* AcquireImages (
    void * arg )
```


16.245.1.2 ConfigureGVCPHeartbeat()

```
int ConfigureGVCPHeartbeat (
    CameraPtr pCam,
    bool enable )
```

16.245.1.3 DisableGVCPHeartbeat()

```
int DisableGVCPHeartbeat (
    CameraPtr pCam )
```

16.245.1.4 main()

```
int main (
    int ,
    char ** )
```

16.245.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    std::string camSerial )
```

16.245.1.6 ResetGVCPHeartbeat()

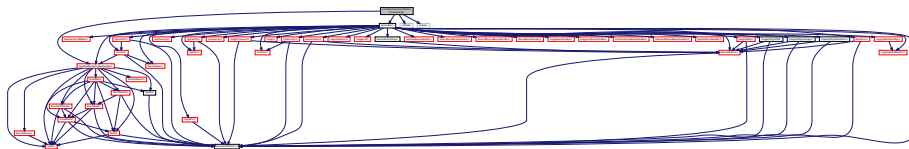
```
int ResetGVCPHeartbeat (
    CameraPtr pCam )
```

16.245.1.7 RunMultipleCameras()

```
int RunMultipleCameras (
    CameraList camList )
```

16.246 src/ActionCommand/ActionCommand.cpp File Reference

Include dependency graph for ActionCommand.cpp:



Functions

- void [SleepyWrapper](#) (int milliseconds)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap, const unsigned int camNum)
- int [ConfigureInterface](#) (const [InterfaceList](#) &interfaceList)
- int [ConfigureIEEE1588](#) (const [CameraList](#) &camList)
- int [ConfigureActionControl](#) (const [CameraList](#) &camList)
- int [ConfigureOtherNodes](#) (const [CameraList](#) &camList)
- int [ConfigureTrigger](#) (const [CameraList](#) camList)
- int [ConfigureChunkData](#) (const [CameraList](#) &camList)
- int [AcquireImages](#) (const [SystemPtr](#) &system, const [InterfaceList](#) &interfaceList, [CameraList](#) camList)
- int [RunMultipleCameras](#) (const [SystemPtr](#) &system, const [InterfaceList](#) &interfaceList, const [CameraList](#) &camList)
- int [main](#) (int, char **)

16.246.1 Function Documentation

16.246.1.1 AcquireImages()

```
int AcquireImages (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    CameraList camList )
```

16.246.1.2 ConfigureActionControl()

```
int ConfigureActionControl (
    const CameraList & camList )
```

16.246.1.3 ConfigureChunkData()

```
int ConfigureChunkData (
    const CameraList & camList )
```

16.246.1.4 ConfigureIEEE1588()

```
int ConfigureIEEE1588 (
    const CameraList & camList )
```

16.246.1.5 ConfigureInterface()

```
int ConfigureInterface (
    const InterfaceList & interfaceList )
```

16.246.1.6 ConfigureOtherNodes()

```
int ConfigureOtherNodes (
    const CameraList & camList )
```

16.246.1.7 ConfigureTrigger()

```
int ConfigureTrigger (
    const CameraList camList )
```

16.246.1.8 main()

```
int main (
    int ,
    char ** )
```

16.246.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    const unsigned int camNum )
```

16.246.1.10 RunMultipleCameras()

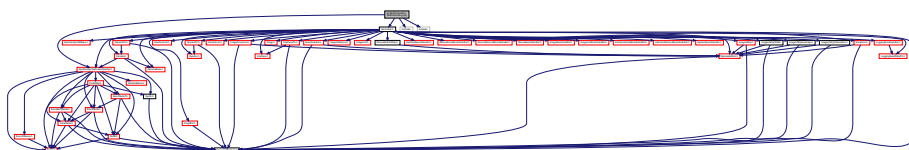
```
int RunMultipleCameras (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    const CameraList & camList )
```

16.246.1.11 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.247 src/BufferHandling/BufferHandling.cpp File Reference

Include dependency graph for BufferHandling.cpp:



Macros

- `#define` [numBuffers](#) 3
- `#define` [z_numTriggers](#) 6
- `#define` [k_numLoops](#) 9

Functions

- void [SleepyWrapper](#) (int milliseconds)
- int [ConfigureTrigger](#) (INodeMap &nodeMap)
- int [GrabNextImageByTrigger](#) (INodeMap &nodeMap)
- int [ResetTrigger](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.247.1 Macro Definition Documentation

16.247.1.1 k_numLoops

```
#define k_numLoops 9
```

16.247.1.2 numBuffers

```
#define numBuffers 3
```

16.247.1.3 z_numTriggers

```
#define z_numTriggers 6
```

16.247.2 Function Documentation

16.247.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.247.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.247.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap )
```

16.247.2.4 main()

```
int main (
    int ,
    char ** )
```

16.247.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.247.2.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.247.2.7 RunSingleCamera()

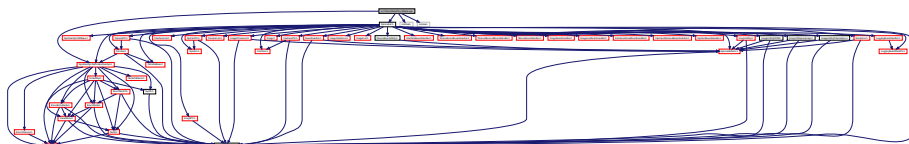
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.247.2.8 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

16.248 src/ChunkData/ChunkData.cpp File Reference

Include dependency graph for ChunkData.cpp:



Enumerations

- enum `chunkDataType` {
 `IMAGE`,
 `NODEMAP` }

Functions

- int `ConfigureChunkData` (INodeMap &nodeMap)
- int `DisplayChunkData` (ImagePtr pImage)
- int `DisplayChunkData` (INodeMap &nodeMap)
- int `PrintDeviceInfo` (INodeMap &nodeMap)
- int `AcquireImages` (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int `DisableChunkData` (INodeMap &nodeMap)
- int `RunSingleCamera` (CameraPtr pCam)
- int `main` (int, char **)

Variables

- const `chunkDataType` `chosenChunkData` = `IMAGE`

16.248.1 Enumeration Type Documentation

16.248.1.1 chunkDataType

enum `chunkDataType`

Enumerator

<code>IMAGE</code>	
<code>NODEMAP</code>	

16.248.2 Function Documentation

16.248.2.1 AcquireImages()

```
int AcquireImages (  
    CameraPtr pCam,  
    INodeMap & nodeMap,  
    INodeMap & nodeMapTLDevice )
```

16.248.2.2 ConfigureChunkData()

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

16.248.2.3 DisableChunkData()

```
int DisableChunkData (
    INodeMap & nodeMap )
```

16.248.2.4 DisplayChunkData() [1/2]

```
int DisplayChunkData (
    ImagePtr pImage )
```

16.248.2.5 DisplayChunkData() [2/2]

```
int DisplayChunkData (
    INodeMap & nodeMap )
```

16.248.2.6 main()

```
int main (
    int ,
    char ** )
```

16.248.2.7 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.248.2.8 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

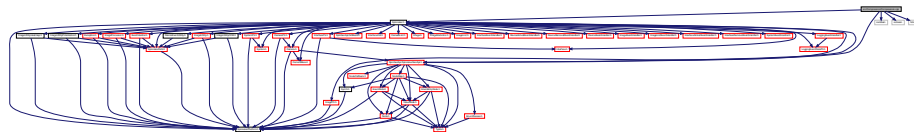

16.248.3 Variable Documentation

16.248.3.1 chosenChunkData

```
const chunkDataType chosenChunkData = IMAGE
```

16.249 src/Compression/Compression.cpp File Reference

Include dependency graph for Compression.cpp:



Classes

- struct [CompressedImageInfo](#)

Functions

- int [ConfigureGVCPHeartbeat](#) ([CameraPtr](#) pCam, bool enable)
- int [ResetGVCPHeartbeat](#) ([CameraPtr](#) pCam)
- int [DisableGVCPHeartbeat](#) ([CameraPtr](#) pCam)
- int [AcquireImages](#) ([CameraPtr](#) pCam, [INodeMap](#) &nodeMap, [INodeMap](#) &nodeMapTLDevice, vector< [CompressedImageInfo](#) > &compressedImageInfos)
- int [PrintDeviceInfo](#) ([INodeMap](#) &nodeMap)
- bool [EnableImageChunkData](#) ([INodeMap](#) &nodeMap)
- bool [DisableImageChunkData](#) ([INodeMap](#) &nodeMap)
- bool [EnableImageCompression](#) ([INodeMap](#) &nodeMap)
- bool [DisableImageCompression](#) ([INodeMap](#) &nodeMap)
- int [ProcessCompressedImagesFromFile](#) (const vector< [CompressedImageInfo](#) > &compressedImageInfos)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- const bool [enableChunkData](#) = false

16.249.1 Function Documentation

16.249.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    vector< CompressedImageInfo > & compressedImageInfos )
```

16.249.1.2 ConfigureGVCPHeartbeat()

```
int ConfigureGVCPHeartbeat (
    CameraPtr pCam,
    bool enable )
```

16.249.1.3 DisableGVCPHeartbeat()

```
int DisableGVCPHeartbeat (
    CameraPtr pCam )
```

16.249.1.4 DisableImageChunkData()

```
bool DisableImageChunkData (
    INodeMap & nodeMap )
```

16.249.1.5 DisableImageCompression()

```
bool DisableImageCompression (
    INodeMap & nodeMap )
```

16.249.1.6 EnableImageChunkData()

```
bool EnableImageChunkData (
    INodeMap & nodeMap )
```

16.249.1.7 EnableImageCompression()

```
bool EnableImageCompression (
    INodeMap & nodeMap )
```

16.249.1.8 main()

```
int main (
    int ,
    char ** )
```

16.249.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.249.1.10 ProcessCompressedImagesFromFile()

```
int ProcessCompressedImagesFromFile (
    const vector< CompressedImageInfo > & compressedImageInfos )
```

16.249.1.11 ResetGVCPHeartbeat()

```
int ResetGVCPHeartbeat (
    CameraPtr pCam )
```

16.249.1.12 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

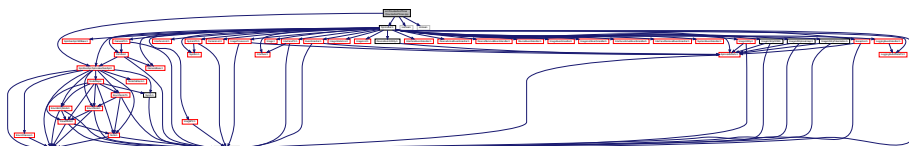
16.249.2 Variable Documentation

16.249.2.1 enableChunkData

```
const bool enableChunkData = false
```

16.250 src/CounterAndTimer/CounterAndTimer.cpp File Reference

Include dependency graph for CounterAndTimer.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [SetupCounterAndTimer](#) (INodeMap &nodeMap)
- int [ConfigureDigitalIO](#) (INodeMap &nodeMap)
- int [ConfigureExposureandTrigger](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [ResetTrigger](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.250.1 Function Documentation

16.250.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.250.1.2 ConfigureDigitalIO()

```
int ConfigureDigitalIO (
    INodeMap & nodeMap )
```

16.250.1.3 ConfigureExposureandTrigger()

```
int ConfigureExposureandTrigger (
    INodeMap & nodeMap )
```

16.250.1.4 main()

```
int main (
    int ,
    char ** )
```

16.250.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.250.1.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.250.1.7 RunSingleCamera()

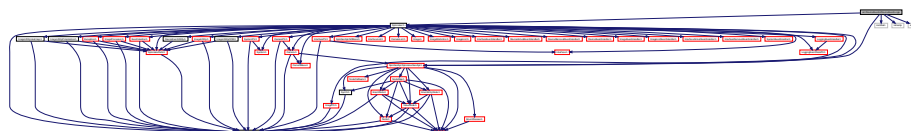
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.250.1.8 SetupCounterAndTimer()

```
int SetupCounterAndTimer (
    INodeMap & nodeMap )
```

16.251 src/DeviceEvents/DeviceEvents.cpp File Reference

Include dependency graph for DeviceEvents.cpp:



Classes

- class [DeviceEventHandlerImpl](#)

Enumerations

- enum [eventType](#) {
[GENERIC](#),
[SPECIFIC](#) }

Functions

- int [ConfigureDeviceEvents](#) (INodeMap &nodeMap, [CameraPtr](#) pCam, [DeviceEventHandlerImpl](#) *&device↔
 EventHandler, const [gcstring](#) &deviceEventName)
- int [ResetDeviceEvents](#) ([CameraPtr](#) pCam, [DeviceEventHandlerImpl](#) *&deviceEventHandler)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- bool [InferenceAvailable](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- const [eventType](#) chosenEvent = [GENERIC](#)

16.251.1 Enumeration Type Documentation

16.251.1.1 eventType

enum [eventType](#)

Enumerator

GENERIC	
SPECIFIC	

16.251.2 Function Documentation

16.251.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
```

```
INodeMap & nodeMap,  
INodeMap & nodeMapTLDevice )
```

16.251.2.2 ConfigureDeviceEvents()

```
int ConfigureDeviceEvents (  
    INodeMap & nodeMap,  
    CameraPtr pCam,  
    DeviceEventHandlerImpl *& deviceEventHandler,  
    const gcstring & deviceEventName )
```

16.251.2.3 InferenceAvailable()

```
bool InferenceAvailable (  
    INodeMap & nodeMap )
```

16.251.2.4 main()

```
int main (  
    int ,  
    char ** )
```

16.251.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (  
    INodeMap & nodeMap )
```

16.251.2.6 ResetDeviceEvents()

```
int ResetDeviceEvents (  
    CameraPtr pCam,  
    DeviceEventHandlerImpl *& deviceEventHandler )
```

16.251.2.7 RunSingleCamera()

```
int RunSingleCamera (  
    CameraPtr pCam )
```

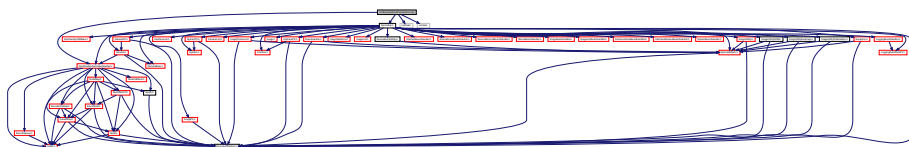
16.251.3 Variable Documentation

16.251.3.1 chosenEvent

```
const eventType chosenEvent = GENERIC
```

16.252 src/Enumeration/Enumeration.cpp File Reference

Include dependency graph for Enumeration.cpp:



Functions

- int [QueryInterface](#) ([InterfacePtr](#) pInterface)
- int [main](#) (int, char **)

16.252.1 Function Documentation

16.252.1.1 main()

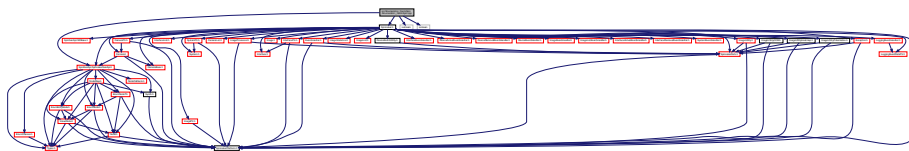
```
int main (  
    int ,  
    char ** )
```

16.252.1.2 QueryInterface()

```
int QueryInterface (  
    InterfacePtr pInterface )
```


16.253 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference

Include dependency graph for Enumeration_QuickSpin.cpp:



Functions

- int [QueryInterface](#) ([InterfacePtr](#) pInterface)
- int [main](#) (int, char **)

16.253.1 Function Documentation

16.253.1.1 main()

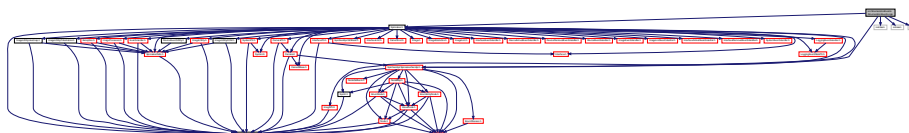
```
int main (  
    int ,  
    char ** )
```

16.253.1.2 QueryInterface()

```
int QueryInterface (  
    InterfacePtr pInterface )
```

16.254 src/EnumerationEvents/EnumerationEvents.cpp File Reference

Include dependency graph for EnumerationEvents.cpp:



Classes

- class [InterfaceEventHandlerImpl](#)
- class [SystemEventHandlerImpl](#)

Functions

- void [CheckGevEnabled](#) ([SystemPtr](#) &pSystem)
- int [main](#) (int, char **)

16.254.1 Function Documentation

16.254.1.1 CheckGevEnabled()

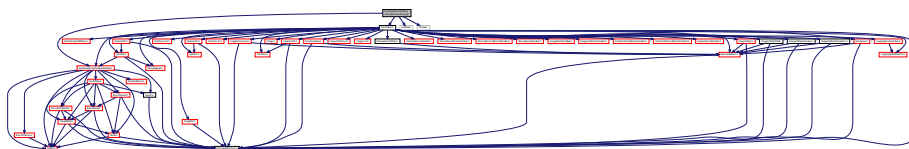
```
void CheckGevEnabled (  
    SystemPtr & pSystem )
```

16.254.1.2 main()

```
int main (  
    int ,  
    char ** )
```

16.255 src/ExceptionHandling/ExceptionHandling.cpp File Reference

Include dependency graph for ExceptionHandling.cpp:



Enumerations

- enum [exceptionType](#) {
 [SPINNAKER_EXCEPTION](#),
 [STANDARD_EXCEPTION](#),
 [STANDARD_CAST_TO_SPINNAKER](#) }

Functions

- void `causeSpinnakerException` ()
- void `causeStandardException` ()
- int `main` (int, char **)

Variables

- const `exceptionType chosenException` = `SPINNAKER_EXCEPTION`

16.255.1 Enumeration Type Documentation

16.255.1.1 exceptionType

enum `exceptionType`

Enumerator

<code>SPINNAKER_EXCEPTION</code>	
<code>STANDARD_EXCEPTION</code>	
<code>STANDARD_CAST_TO_SPINNAKER</code>	

16.255.2 Function Documentation

16.255.2.1 causeSpinnakerException()

```
void causeSpinnakerException ( )
```

16.255.2.2 causeStandardException()

```
void causeStandardException ( )
```

16.255.2.3 main()

```
int main (
    int ,
    char ** )
```

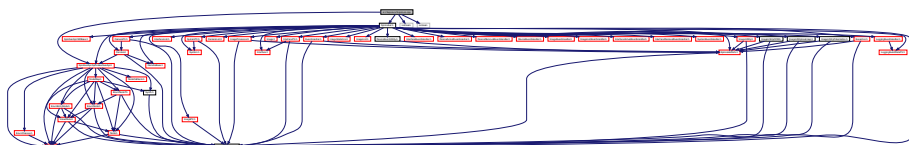
16.255.3 Variable Documentation

16.255.3.1 chosenException

```
const exceptionType chosenException = SPINNAKER_EXCEPTION
```

16.256 src/Exposure/Exposure.cpp File Reference

Include dependency graph for Exposure.cpp:



Functions

- int [ConfigureExposure](#) (INodeMap &nodeMap)
- int [ResetExposure](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.256.1 Function Documentation

16.256.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.256.1.2 ConfigureExposure()

```
int ConfigureExposure (
    INodeMap & nodeMap )
```

16.256.1.3 main()

```
int main (
    int ,
    char ** )
```

16.256.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.256.1.5 ResetExposure()

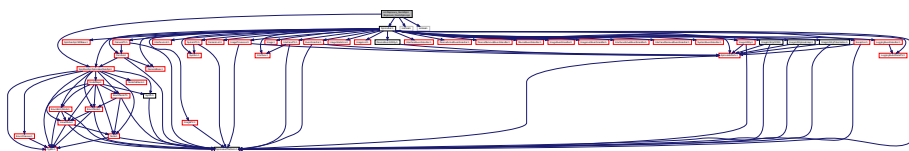
```
int ResetExposure (
    INodeMap & nodeMap )
```

16.256.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.257 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference

Include dependency graph for Exposure_QuickSpin.cpp:



Functions

- int [ConfigureExposure](#) ([CameraPtr](#) pCam)
- int [ResetExposure](#) ([CameraPtr](#) pCam)
- int [PrintDeviceInfo](#) ([CameraPtr](#) pCam)
- int [AcquireImages](#) ([CameraPtr](#) pCam)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.257.1 Function Documentation

16.257.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

16.257.1.2 ConfigureExposure()

```
int ConfigureExposure (
    CameraPtr pCam )
```

16.257.1.3 main()

```
int main (
    int ,
    char ** )
```

16.257.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

16.257.1.5 ResetExposure()

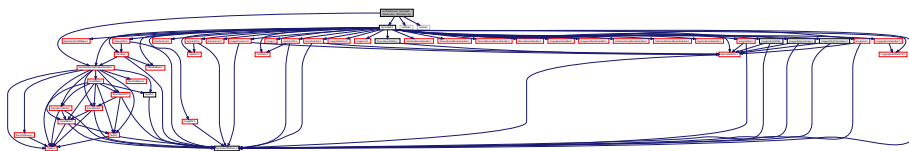
```
int ResetExposure (
    CameraPtr pCam )
```

16.257.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.258 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference

Include dependency graph for FileAccess_QuickSpin.cpp:



Functions

- static void [PrintResultMessage](#) (bool result)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- bool [InitializeSystem](#) (SystemPtr &system, [CameraList](#) &camList, [CameraPtr](#) &pCam)
- static void [PrintDebugMessage](#) (string msg)
- bool [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, [ImagePtr](#) pReferenceImage)
- bool [ExecuteDeleteCommand](#) ([CameraPtr](#) pCam)
- bool [OpenFileToWrite](#) ([CameraPtr](#) pCam)
- bool [ExecuteWriteCommand](#) ([CameraPtr](#) pCam)
- bool [CloseFile](#) ([CameraPtr](#) pCam)
- bool [UploadImage](#) ()
- bool [OpenFileToRead](#) ([CameraPtr](#) pCamera)
- bool [ExecuteReadCommand](#) ([CameraPtr](#) pCamera)
- bool [DownloadImage](#) ()
- void [PrintUsage](#) ()
- int [main](#) (int argc, char *argv[])

Variables

- static bool [_enableDebug](#) = false
- static [gcstring](#) [_fileSelector](#) = "UserFile1"

16.258.1 Function Documentation

16.258.1.1 AcquireImages()

```
bool AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImagePtr pReferenceImage )
```

16.258.1.2 CloseFile()

```
bool CloseFile (
    CameraPtr pCam )
```

16.258.1.3 DownloadImage()

```
bool DownloadImage ( )
```

16.258.1.4 ExecuteDeleteCommand()

```
bool ExecuteDeleteCommand (
    CameraPtr pCam )
```

16.258.1.5 ExecuteReadCommand()

```
bool ExecuteReadCommand (
    CameraPtr pCamera )
```

16.258.1.6 ExecuteWriteCommand()

```
bool ExecuteWriteCommand (
    CameraPtr pCam )
```

16.258.1.7 InitializeSystem()

```
bool InitializeSystem (
    SystemPtr & system,
    CameraList & camList,
    CameraPtr & pCam )
```

16.258.1.8 main()

```
int main (
    int argc,
    char * argv[] )
```


16.258.1.9 OpenFileToRead()

```
bool OpenFileToRead (
    CameraPtr pCamera )
```

16.258.1.10 OpenFileToWrite()

```
bool OpenFileToWrite (
    CameraPtr pCam )
```

16.258.1.11 PrintDebugMessage()

```
static void PrintDebugMessage (
    string msg ) [static]
```

16.258.1.12 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.258.1.13 PrintResultMessage()

```
static void PrintResultMessage (
    bool result ) [static]
```

16.258.1.14 PrintUsage()

```
void PrintUsage ( )
```

16.258.1.15 UploadImage()

```
bool UploadImage ( )
```

16.258.2 Variable Documentation

16.258.2.1 `_enableDebug`

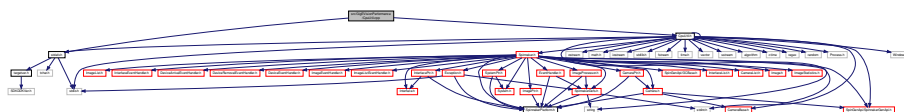
```
bool _enableDebug = false [static]
```

16.258.2.2 `_fileSelector`

```
gcstring _fileSelector = "UserFile1" [static]
```

16.259 `src/GigEVisionPerformance/CpuUtil.cpp` File Reference

Include dependency graph for `CpuUtil.cpp`:



Namespaces

- [CpuUtil](#)
- [PerformanceCounter](#)
- [SecondsCounter](#)
- [Conversion](#)

Functions

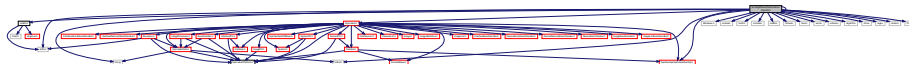
- bool [StartCpuTracing](#) (CpuUsageInfo *cpuUsage)
- bool [StopCpuTracing](#) (CpuUsageInfo *cpuUsage)
- std::string [GetCpuStats](#) (CpuUsageInfo *cpuUsage)
- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()
- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()
- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)
- string [NumToCString](#) (float number)

Variables

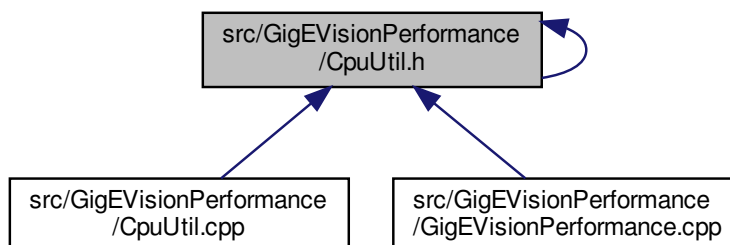
- double [PCFreq](#)
- __int64 [CounterStart](#)
- time_t [startTime](#)
- time_t [endTime](#)
- double [timeDiff](#)

16.260 src/GigEVisionPerformance/CpuUtil.h File Reference

Include dependency graph for CpuUtil.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [CpuUsageInfo](#)

Namespaces

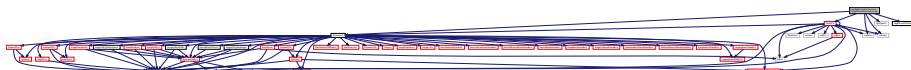
- [CpuUtil](#)
- [PerformanceCounter](#)
- [SecondsCounter](#)
- [Conversion](#)

Functions

- bool [StartCpuTracing](#) (CpuUsageInfo *cpuUsage)
- bool [StopCpuTracing](#) (CpuUsageInfo *cpuUsage)
- std::string [GetCpuStats](#) (CpuUsageInfo *cpuUsage)
- void [StartPerformanceCounter](#) ()
- double [GetPerformanceCounter](#) ()
- void [StartSecondsCounter](#) ()
- int [GetSecondsCounter](#) ()
- string [NumToCString](#) (int number)
- string [NumToCString](#) (double number)

16.261 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference

Include dependency graph for GigEVisionPerformance.cpp:



Functions

- void [PrintUsage](#) ()
- bool [ParseArguments](#) (int argc, char *argv[])
- void [getCameraCategory](#) (INodeMap &nodeMap, string categoryString)
- void [PrintDataStreamInfo](#) (const [Spinnaker::CameraPtr](#) pCamera)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, int numImagesToAcquire, int iteration)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- void [PrintCPUUsage](#) ()
- void [PrintAllNodes](#) ([CameraPtr](#) pCam)
- bool [EnableManualFramerate](#) ([CameraPtr](#) pCam)
- bool [SetFrameRate](#) ([CameraPtr](#) pCam)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int argc, char *argv[])

Variables

- [CpuUtil::CpuUsageInfo](#) cpuUsageInfo
- int [TestDuration](#) = 0
- char * [PixelFormatToSet](#) = nullptr
- int [PacketSizeToSet](#) = 9000
- int [PacketDelayToSet](#) = 0
- bool [IsRelease](#) = false
- bool [UseDuration](#) = false
- bool [UseMaxFramerate](#) = false
- float [UserSetFramerate](#) = 0.0
- int [NumImagesToGrab](#) = 100
- const char * [argNumImages](#) = "-numimages"

- `const char * argDuration = "-duration"`
- `const char * argRelease = "-callrelease"`
- `const char * argBayerRG = "-bayerrg"`
- `const char * argPacketSize = "-packetsize"`
- `const char * argPacketDelay = "-packetdelay"`
- `const char * argMaxFrames = "-maxfps"`
- `const char * argUserSetFrames = "-fps"`
- `const char * argPrintUsage = "-?"`

16.261.1 Function Documentation

16.261.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    int numImagesToAcquire,
    int iteration )
```

16.261.1.2 EnableManualFramerate()

```
bool EnableManualFramerate (
    CameraPtr pCam )
```

16.261.1.3 getCameraCategory()

```
void getCameraCategory (
    INodeMap & nodeMap,
    string categoryString )
```

16.261.1.4 main()

```
int main (
    int argc,
    char * argv[] )
```

16.261.1.5 ParseArguments()

```
bool ParseArguments (
    int argc,
    char * argv[] )
```

16.261.1.6 PrintAllNodes()

```
void PrintAllNodes (
    CameraPtr pCam )
```

16.261.1.7 PrintCPUUsage()

```
void PrintCPUUsage ( )
```

16.261.1.8 PrintDataStreamInfo()

```
void PrintDataStreamInfo (
    const Spinnaker::CameraPtr pCamera )
```

16.261.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.261.1.10 PrintUsage()

```
void PrintUsage ( )
```

16.261.1.11 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.261.1.12 SetFrameRate()

```
bool SetFrameRate (
    CameraPtr pCam )
```

16.261.2 Variable Documentation

16.261.2.1 argBayerRG

```
const char* argBayerRG = "-bayerrg"
```

16.261.2.2 argDuration

```
const char* argDuration = "-duration"
```

16.261.2.3 argMaxFrames

```
const char* argMaxFrames = "-maxfps"
```

16.261.2.4 argNumImages

```
const char* argNumImages = "-numimages"
```

16.261.2.5 argPacketDelay

```
const char* argPacketDelay = "-packetdelay"
```

16.261.2.6 argPacketSize

```
const char* argPacketSize = "-packetsize"
```

16.261.2.7 argPrintUsage

```
const char* argPrintUsage = "-?"
```

16.261.2.8 argRelease

```
const char* argRelease = "-callrelease"
```

16.261.2.9 argUserSetFrames

```
const char* argUserSetFrames = "-fps"
```

16.261.2.10 cpuUsageInfo

```
CpuUtil::CpuUsageInfo cpuUsageInfo
```

16.261.2.11 IsRelease

```
bool IsRelease = false
```

16.261.2.12 NumImagesToGrab

```
int NumImagesToGrab = 100
```

16.261.2.13 PacketDelayToSet

```
int PacketDelayToSet = 0
```

16.261.2.14 PacketSizeToSet

```
int PacketSizeToSet = 9000
```


16.261.2.15 PixelFormatToSet

```
char* PixelFormatToSet = nullptr
```

16.261.2.16 TestDuration

```
int TestDuration = 0
```

16.261.2.17 UseDuration

```
bool UseDuration = false
```

16.261.2.18 UseMaxFramerate

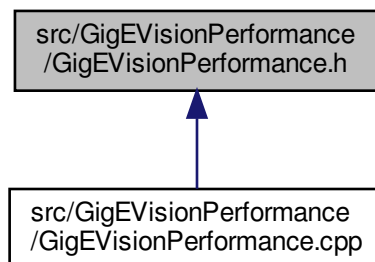
```
bool UseMaxFramerate = false
```

16.261.2.19 UserSetFramerate

```
float UserSetFramerate = 0.0
```

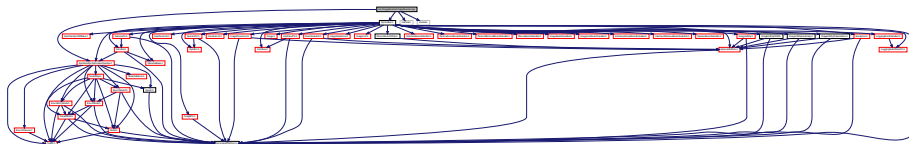
16.262 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference

This graph shows which files directly or indirectly include this file:



16.263 src/ImageEvents/ImageEvents.cpp File Reference

Include dependency graph for ImageEvents.cpp:



Classes

- class [ImageEventHandlerImpl](#)

Functions

- void [SleepyWrapper](#) (int milliseconds)
- int [ConfigureImageEvents](#) (CameraPtr pCam, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [WaitForImages](#) ([ImageEventHandlerImpl](#) *&imageEventHandler)
- int [ResetImageEvents](#) (CameraPtr pCam, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice, [ImageEventHandlerImpl](#) *&imageEventHandler)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.263.1 Function Documentation

16.263.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.263.1.2 ConfigureImageEvents()

```
int ConfigureImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.263.1.3 main()

```
int main (
    int ,
    char ** )
```

16.263.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.263.1.5 ResetImageEvents()

```
int ResetImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

16.263.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.263.1.7 SleepyWrapper()

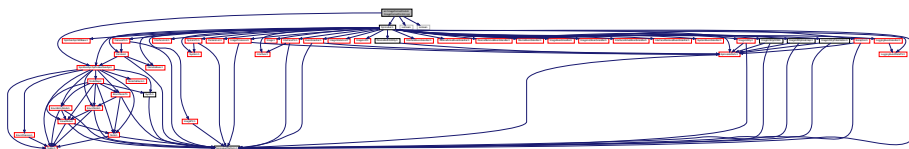
```
void SleepyWrapper (
    int milliseconds )
```

16.263.1.8 WaitForImages()

```
int WaitForImages (
    ImageEventHandlerImpl *& imageEventHandler )
```

16.264 src/ImageFormatControl/ImageFormatControl.cpp File Reference

Include dependency graph for ImageFormatControl.cpp:



Functions

- int [ConfigureCustomImageSettings](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.264.1 Function Documentation

16.264.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.264.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
    INodeMap & nodeMap )
```

16.264.1.3 main()

```
int main (
    int ,
    char ** )
```

16.264.1.4 PrintDeviceInfo()

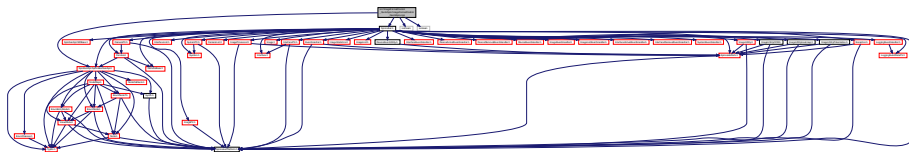
```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.264.1.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.265 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference

Include dependency graph for ImageFormatControl_QuickSpin.cpp:



Functions

- int [ConfigureCustomImageSettings](#) (CameraPtr pCam)
- int [PrintDeviceInfo](#) (CameraPtr pCam)
- int [AcquireImages](#) (CameraPtr pCam)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.265.1 Function Documentation

16.265.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

16.265.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (  
    CameraPtr pCam )
```

16.265.1.3 main()

```
int main (  
    int ,  
    char ** )
```

16.265.1.4 PrintDeviceInfo()

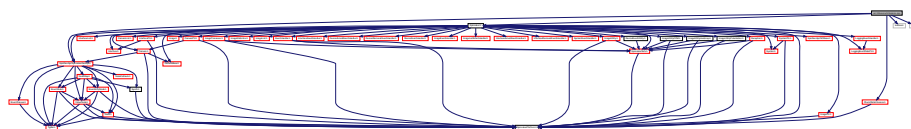
```
int PrintDeviceInfo (  
    CameraPtr pCam )
```

16.265.1.5 RunSingleCamera()

```
int RunSingleCamera (  
    CameraPtr pCam )
```

16.266 src/Inference/Inference.cpp File Reference

Include dependency graph for Inference.cpp:



Enumerations

- enum `InferenceNetworkType` {
 DETECTION,
 CLASSIFICATION }
- enum `FileUploadPersistence` {
 FLASH,
 DDR }

Functions

- `const std::vector< std::string > labelClassification (arrayLabelClassification, end(arrayLabelClassification))`
- `const std::vector< std::string > labelDetection (arrayLabelDetection, end(arrayLabelDetection))`
- `int PrintDeviceInfo (INodeMap &nodeMap)`
- `bool CameraDeleteFile (INodeMap &nodeMap)`
- `bool CameraOpenFile (INodeMap &nodeMap)`
- `bool CameraWriteToFile (INodeMap &nodeMap)`
- `bool CameraCloseFile (INodeMap &nodeMap)`
- `std::vector< char > LoadFileIntoMemory (const string &filename)`
- `int UploadFileToCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName, const std::string &filePath)`
- `int DeleteFileOnCamera (INodeMap &nodeMap, const std::string &fileSelectorEntryName)`
- `int SetChunkEnable (INodeMap &nodeMap, const gcstring &entryName, const bool enable)`
- `int ConfigureChunkData (INodeMap &nodeMap)`
- `int DisableChunkData (INodeMap &nodeMap)`
- `int DisplayChunkData (const ImagePtr plmage)`
- `int DisableTrigger (INodeMap &nodeMap)`
- `int ConfigureTrigger (INodeMap &nodeMap)`
- `int ConfigureInference (INodeMap &nodeMap, bool isEnabled)`
- `int ConfigureTestPattern (INodeMap &nodeMap, bool isEnabled)`
- `int AcquireImages (const CameraPtr &pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)`
- `int RunSingleCamera (const CameraPtr &pCam)`
- `int main (int, char **)`

Variables

- `const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION`
- `const FileUploadPersistence chosenFileUploadPersistence = DDR`
- `const std::string networkFilePath`
- `const std::string injectedImageFilePath`
- `const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640 : 720)`
- `const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400 : 540)`
- `const char * arrayLabelClassification [] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}`
- `const char * arrayLabelDetection []`

16.266.1 Enumeration Type Documentation

16.266.1.1 FileUploadPersistence

enum `FileUploadPersistence`

Enumerator

FLASH	
DDR	

16.266.1.2 InferenceNetworkType

enum [InferenceNetworkType](#)

Enumerator

DETECTION	This network determines the most likely class given a set of predetermined, trained options. Object detection can also provide a location within the image (in the form of a "bounding box" surrounding the class), and can detect multiple objects.
CLASSIFICATION	This network determines the best option from a list of predetermined options; the camera gives a percentage that determines the likelihood of the currently perceived image being one of the classes it has been trained to recognize.

16.266.2 Function Documentation

16.266.2.1 AcquireImages()

```
int AcquireImages (
    const CameraPtr & pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTTLDevice )
```

16.266.2.2 CameraCloseFile()

```
bool CameraCloseFile (
    INodeMap & nodeMap )
```

16.266.2.3 CameraDeleteFile()

```
bool CameraDeleteFile (
    INodeMap & nodeMap )
```

16.266.2.4 CameraOpenFile()

```
bool CameraOpenFile (
    INodeMap & nodeMap )
```


16.266.2.5 CameraWriteToFile()

```
bool CameraWriteToFile (
    INodeMap & nodeMap )
```

16.266.2.6 ConfigureChunkData()

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

16.266.2.7 ConfigureInference()

```
int ConfigureInference (
    INodeMap & nodeMap,
    bool isEnabled )
```

16.266.2.8 ConfigureTestPattern()

```
int ConfigureTestPattern (
    INodeMap & nodeMap,
    bool isEnabled )
```

16.266.2.9 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.266.2.10 DeleteFileOnCamera()

```
int DeleteFileOnCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName )
```

16.266.2.11 DisableChunkData()

```
int DisableChunkData (
    INodeMap & nodeMap )
```

16.266.2.12 DisableTrigger()

```
int DisableTrigger (
    INodeMap & nodeMap )
```

16.266.2.13 DisplayChunkData()

```
int DisplayChunkData (
    const ImagePtr pImage )
```

16.266.2.14 labelClassification()

```
const std::vector<std::string> labelClassification (
    arrayLabelClassification ,
    end(arrayLabelClassification) )
```

16.266.2.15 labelDetection()

```
const std::vector<std::string> labelDetection (
    arrayLabelDetection ,
    end(arrayLabelDetection) )
```

16.266.2.16 LoadFileIntoMemory()

```
std::vector<char> LoadFileIntoMemory (
    const string & filename )
```

16.266.2.17 main()

```
int main (
    int ,
    char ** )
```

16.266.2.18 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.266.2.19 RunSingleCamera()

```
int RunSingleCamera (
    const CameraPtr & pCam )
```

16.266.2.20 SetChunkEnable()

```
int SetChunkEnable (
    INodeMap & nodeMap,
    const gcstring & entryName,
    const bool enable )
```

16.266.2.21 UploadFileToCamera()

```
int UploadFileToCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName,
    const std::string & filePath )
```

16.266.3 Variable Documentation

16.266.3.1 arrayLabelClassification

```
const char* arrayLabelClassification[] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}
```

16.266.3.2 arrayLabelDetection

```
const char* arrayLabelDetection[ ]
```

Initial value:

```
= { "background", "aeroplane", "bicycle", "bird", "boat", "bottle",  
    "bus", "car", "cat", "chair", "cow",  
    "diningtable", "dog", "horse", "motorbike", "person", "pottedplant",  
    "sheep", "sofa", "train", "monitor" }
```

16.266.3.3 chosenFileUploadPersistence

```
const FileUploadPersistence chosenFileUploadPersistence = DDR
```

16.266.3.4 chosenInferenceNetworkType

```
const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION
```

16.266.3.5 injectedImageFilePath

```
const std::string injectedImageFilePath
```

Initial value:

```
=  
    (chosenInferenceNetworkType == CLASSIFICATION ? "Injected_Image_Classification.raw"  
    : "Injected_Image_Detection.raw")
```

16.266.3.6 injectedImageHeight

```
const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400  
: 540)
```

16.266.3.7 injectedImageWidth

```
const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640  
: 720)
```

16.266.3.8 networkFilePath

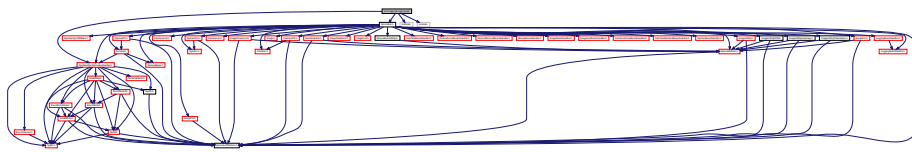
```
const std::string networkFilePath
```

Initial value:

```
=  
    (chosenInferenceNetworkType == CLASSIFICATION ? "Network_Classification" : "Network_Detection")
```

16.267 src/Logging/Logging.cpp File Reference

Include dependency graph for Logging.cpp:



Classes

- class [LoggingEventHandlerImpl](#)

Functions

- int [main](#) (int, char **)

Variables

- const [SpinnakerLogLevel](#) [k_LoggingLevel](#) = SPINNAKER_LOG_LEVEL_DEBUG

16.267.1 Function Documentation

16.267.1.1 main()

```
int main (  
    int ,  
    char ** )
```

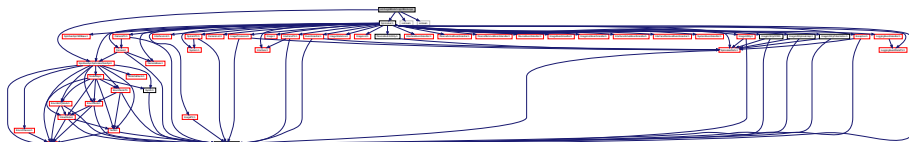
16.267.2 Variable Documentation

16.267.2.1 k_LoggingLevel

```
const SpinnakerLogLevel k_LoggingLevel = SPINNAKER_LOG_LEVEL_DEBUG
```

16.268 src/LogicBlock/LogicBlock.cpp File Reference

Include dependency graph for LogicBlock.cpp:



Functions

- int [ConfigureTrigger](#) (INodeMap &nodeMap)
- int [ConfigureLogicBlock](#) (INodeMap &nodeMap)
- int [GrabTwoImages](#) (INodeMap &nodeMap, [CameraPtr](#) pCam)
- int [ResetTrigger](#) (INodeMap &nodeMap)
- int [ResetExposure](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.268.1 Function Documentation

16.268.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.268.1.2 ConfigureLogicBlock()

```
int ConfigureLogicBlock (
    INodeMap & nodeMap )
```

16.268.1.3 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.268.1.4 GrabTwoImages()

```
int GrabTwoImages (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

16.268.1.5 main()

```
int main (
    int ,
    char ** )
```

16.268.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.268.1.7 ResetExposure()

```
int ResetExposure (
    INodeMap & nodeMap )
```

16.268.1.8 ResetTrigger()

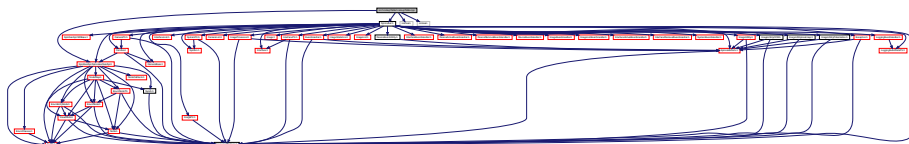
```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.268.1.9 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.269 src/LookupTable/LookupTable.cpp File Reference

Include dependency graph for LookupTable.cpp:



Functions

- void [PrintRetrieveNodeFailure](#) (string node, string name)
- int [ConfigureLookupTables](#) (INodeMap &nodeMap)
- int [ResetLookupTables](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.269.1 Function Documentation

16.269.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.269.1.2 ConfigureLookupTables()

```
int ConfigureLookupTables (
    INodeMap & nodeMap )
```

16.269.1.3 main()

```
int main (
    int ,
    char ** )
```


16.269.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.269.1.5 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

16.269.1.6 ResetLookupTables()

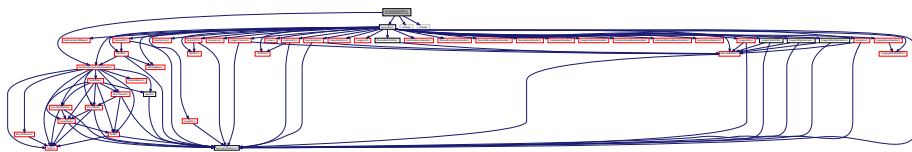
```
int ResetLookupTables (
    INodeMap & nodeMap )
```

16.269.1.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.270 src/NodeMapCallback/NodeMapCallback.cpp File Reference

Include dependency graph for NodeMapCallback.cpp:

**Functions**

- void [OnHeightNodeUpdate](#) (INode *node)
- void [OnGainNodeUpdate](#) (INode *node)
- int [ConfigureCallbacks](#) (INodeMap &nodeMap, [CallbackHandleType](#) &callbackHeight, [CallbackHandleType](#) &callbackGain)
- int [ChangeHeightAndGain](#) (INodeMap &nodeMap)
- int [ResetCallbacks](#) (INodeMap &nodeMap, [CallbackHandleType](#) callbackHeight, [CallbackHandleType](#) callbackGain)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

16.270.1 Function Documentation

16.270.1.1 ChangeHeightAndGain()

```
int ChangeHeightAndGain (
    INodeMap & nodeMap )
```

16.270.1.2 ConfigureCallbacks()

```
int ConfigureCallbacks (
    INodeMap & nodeMap,
    CallbackHandleType & callbackHeight,
    CallbackHandleType & callbackGain )
```

16.270.1.3 main()

```
int main (
    int ,
    char ** )
```

16.270.1.4 OnGainNodeUpdate()

```
void OnGainNodeUpdate (
    INode * node )
```

16.270.1.5 OnHeightNodeUpdate()

```
void OnHeightNodeUpdate (
    INode * node )
```

16.270.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.270.1.7 ResetCallbacks()

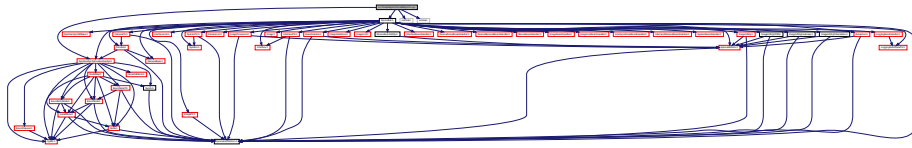
```
int ResetCallbacks (
    INodeMap & nodeMap,
    CallbackHandleType callbackHeight,
    CallbackHandleType callbackGain )
```

16.270.1.8 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.271 src/NodeMapInfo/NodeMapInfo.cpp File Reference

Include dependency graph for NodeMapInfo.cpp:



Enumerations

- enum readType {
 VALUE,
 INDIVIDUAL }

Functions

- int PrintEnumerationSelector (CNodePtr node, unsigned int level)
- void Indent (unsigned int level)
- int PrintValueNode (CNodePtr node, unsigned int level)
- int PrintStringNode (CNodePtr node, unsigned int level)
- int PrintIntegerNode (CNodePtr node, unsigned int level)
- int PrintFloatNode (CNodePtr node, unsigned int level)
- int PrintBooleanNode (CNodePtr node, unsigned int level)
- int PrintCommandNode (CNodePtr node, unsigned int level)
- int PrintEnumerationNodeAndCurrentEntry (CNodePtr node, unsigned int level)
- int PrintNode (CNodePtr node, unsigned int level)
- int PrintCategoryNodeAndAllFeatures (CNodePtr node, unsigned int level)
- int RunSingleCamera (CameraPtr cam)
- int main (int, char **)

Variables

- const unsigned int `maxChars` = 35
- const `readType` `chosenRead` = `VALUE`

16.271.1 Enumeration Type Documentation

16.271.1.1 `readType`

enum `readType`

Enumerator

VALUE	
INDIVIDUAL	

16.271.2 Function Documentation

16.271.2.1 `Indent()`

```
void Indent (
    unsigned int level )
```

16.271.2.2 `main()`

```
int main (
    int ,
    char ** )
```

16.271.2.3 `PrintBooleanNode()`

```
int PrintBooleanNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.4 PrintCategoryNodeAndAllFeatures()

```
int PrintCategoryNodeAndAllFeatures (
    CNodePtr node,
    unsigned int level )
```

16.271.2.5 PrintCommandNode()

```
int PrintCommandNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.6 PrintEnumerationNodeAndCurrentEntry()

```
int PrintEnumerationNodeAndCurrentEntry (
    CNodePtr node,
    unsigned int level )
```

16.271.2.7 PrintEnumerationSelector()

```
int PrintEnumerationSelector (
    CNodePtr node,
    unsigned int level )
```

16.271.2.8 PrintFloatNode()

```
int PrintFloatNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.9 PrintIntegerNode()

```
int PrintIntegerNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.10 PrintNode()

```
int PrintNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.11 PrintStringNode()

```
int PrintStringNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.12 PrintValueNode()

```
int PrintValueNode (
    CNodePtr node,
    unsigned int level )
```

16.271.2.13 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr cam )
```

16.271.3 Variable Documentation**16.271.3.1 chosenRead**

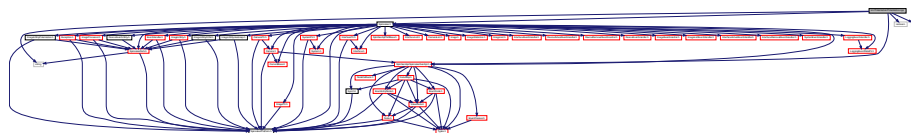
```
const readType chosenRead = VALUE
```

16.271.3.2 maxChars

```
const unsigned int maxChars = 35
```

16.272 src/Polarization/Polarization.cpp File Reference

Include dependency graph for Polarization.cpp:



Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [ConfigureStream](#) (INodeMap &nodeMap)
- int [SaveImage](#) (const [ImagePtr](#) &pImage, const string filename, [gcstring](#) &serialNumber)
- std::string [GetQuadFileNameAppendage](#) (const [PolarizationQuadrant](#) quadrant)
- int [CreateHeatmapImages](#) (const [ImagePtr](#) &mono8Image, const string baseFilename, [gcstring](#) &deviceSerialNumber)
- int [ExtractAndSavePolarQuadImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateAndSaveGlareReducedImage](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateNormalizedImage](#) (const [ImagePtr](#) &imageToNormalize, const string baseFilename, [gcstring](#) &deviceSerialNumber, [SourceDataRange](#) srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)
- int [CreateAndSaveStokesImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [CreateAndSaveAolpDolpImages](#) (const [ImagePtr](#) &pRawPolarizedImage, [gcstring](#) &deviceSerialNumber)
- int [AcquireImages](#) ([CameraPtr](#) pCam, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) ([CameraPtr](#) pCam)
- int [main](#) (int, char **)

Variables

- static bool [isPixelFormatColor](#) = false

16.272.1 Function Documentation

16.272.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMapTLDevice )
```

16.272.1.2 ConfigureStream()

```
int ConfigureStream (
    INodeMap & nodeMap )
```

16.272.1.3 CreateAndSaveAolpDolpImages()

```
int CreateAndSaveAolpDolpImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.272.1.4 CreateAndSaveGlareReducedImage()

```
int CreateAndSaveGlareReducedImage (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.272.1.5 CreateAndSaveStokesImages()

```
int CreateAndSaveStokesImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.272.1.6 CreateHeatmapImages()

```
int CreateHeatmapImages (
    const ImagePtr & mono8Image,
    const string baseFilename,
    gcstring & deviceSerialNumber )
```

16.272.1.7 CreateNormalizedImage()

```
int CreateNormalizedImage (
    const ImagePtr & imageToNormalize,
    const string baseFilename,
    gcstring & deviceSerialNumber,
    SourceDataRange srcDataRange = SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE )
```


16.272.1.8 ExtractAndSavePolarQuadImages()

```
int ExtractAndSavePolarQuadImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

16.272.1.9 GetQuadFileNameAppendage()

```
std::string GetQuadFileNameAppendage (
    const PolarizationQuadrant quadrant )
```

16.272.1.10 main()

```
int main (
    int ,
    char ** )
```

16.272.1.11 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.272.1.12 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.272.1.13 SaveImage()

```
int SaveImage (
    const ImagePtr & pImage,
    const string filename,
    gcstring & serialNumber )
```

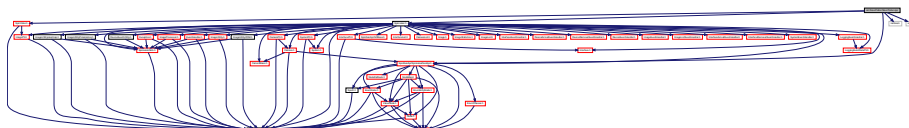
16.272.2 Variable Documentation

16.272.2.1 isPixelFormatColor

```
bool isPixelFormatColor = false [static]
```

16.273 src/SaveToAvi/SaveToAvi.cpp File Reference

Include dependency graph for SaveToAvi.cpp:



Enumerations

- enum `videoType` {
`UNCOMPRESSED`,
`MJPEG`,
`H264` }

Functions

- int `SaveVectorToVideo` (INodeMap &nodeMap, INodeMap &nodeMapTLDevice, vector< `ImagePtr` > &images)
- int `PrintDeviceInfo` (INodeMap &nodeMap)
- int `AcquireImages` (`CameraPtr` pCam, INodeMap &nodeMap, vector< `ImagePtr` > &images)
- int `RunSingleCamera` (`CameraPtr` pCam)
- int `main` (int, char **)

Variables

- const `videoType` `chosenVideoType` = `UNCOMPRESSED`

16.273.1 Enumeration Type Documentation

16.273.1.1 videoType

```
enum videoType
```

Enumerator

UNCOMPRESSED	
MJPEG	
H264	

16.273.2 Function Documentation

16.273.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    vector< ImagePtr > & images )
```

16.273.2.2 main()

```
int main (
    int ,
    char ** )
```

16.273.2.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.273.2.4 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.273.2.5 SaveVectorToVideo()

```
int SaveVectorToVideo (
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice,
    vector< ImagePtr > & images )
```

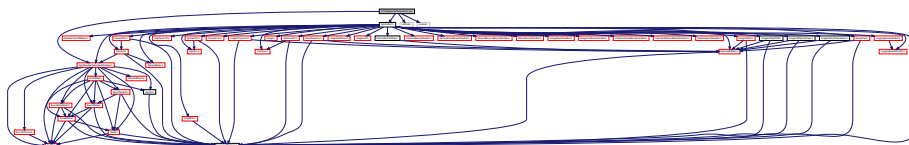
16.273.3 Variable Documentation

16.273.3.1 chosenVideoType

```
const videoType chosenVideoType = UNCOMPRESSED
```

16.274 src/Sequencer/Sequencer.cpp File Reference

Include dependency graph for Sequencer.cpp:



Functions

- void [PrintRetrieveNodeFailure](#) (string node, string name)
- int [ConfigureSequencerPartOne](#) (INodeMap &nodeMap)
- int [SetSingleState](#) (INodeMap &nodeMap, unsigned int sequenceNumber, int64_t widthToSet, int64_t heightToSet, double exposureTimeToSet, double gainToSet)
- int [ConfigureSequencerPartTwo](#) (INodeMap &nodeMap)
- int [ResetSequencer](#) (INodeMap &nodeMap)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapGenTL, uint64_t timeout)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char **)

16.274.1 Function Documentation

16.274.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    uint64_t timeout )
```

16.274.1.2 ConfigureSequencerPartOne()

```
int ConfigureSequencerPartOne (
    INodeMap & nodeMap )
```

16.274.1.3 ConfigureSequencerPartTwo()

```
int ConfigureSequencerPartTwo (
    INodeMap & nodeMap )
```

16.274.1.4 main()

```
int main (
    int ,
    char ** )
```

16.274.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.274.1.6 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

16.274.1.7 ResetSequencer()

```
int ResetSequencer (
    INodeMap & nodeMap )
```

16.274.1.8 RunSingleCamera()

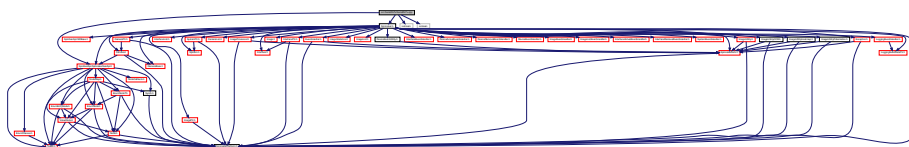
```
int RunSingleCamera (
    CameraPtr pCam )
```

16.274.1.9 SetSingleState()

```
int SetSingleState (
    INodeMap & nodeMap,
    unsigned int sequenceNumber,
    int64_t widthToSet,
    int64_t heightToSet,
    double exposureTimeToSet,
    double gainToSet )
```

16.275 src/SerialRxTx/SerialRxTx.cpp File Reference

Include dependency graph for SerialRxTx.cpp:



Macros

- `#define COM_PORT_COUNT_MAX 256`
- `#define TWO_SECOND_DELAY 2000`
- `#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000`
- `#define SERIAL_PORT_BAUD_RATE 19200`
- `#define SERIAL_PORT_STOP_BITS 0`
- `#define SERIAL_PORT_PARITY_BITS 0`
- `#define SERIAL_PORT_DELAY 1500`
- `#define DATA_BITS 8`
- `#define MILLISECOND 1000`

Functions

- `int PrintDeviceInfo (INodeMap &nodeMap)`
- `int ConfigureDevice (CameraPtr pCam, HANDLE &hFileHandle)`
- `int SerialRx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int SerialTx (CameraPtr pCam, INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int CleanUp (INodeMap &nodeMap, HANDLE &hFileHandle)`
- `int RunSingleCamera (CameraPtr pCam)`
- `int main (int, char **)`

16.275.1 Macro Definition Documentation

16.275.1.1 COM_PORT_COUNT_MAX

```
#define COM_PORT_COUNT_MAX 256
```

16.275.1.2 DATA_BITS

```
#define DATA_BITS 8
```

16.275.1.3 MILLISECOND

```
#define MILLISECOND 1000
```

16.275.1.4 SERIAL_PORT_BAUD_RATE

```
#define SERIAL_PORT_BAUD_RATE 19200
```

16.275.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND

```
#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000
```

16.275.1.6 SERIAL_PORT_DELAY

```
#define SERIAL_PORT_DELAY 1500
```

16.275.1.7 SERIAL_PORT_PARITY_BITS

```
#define SERIAL_PORT_PARITY_BITS 0
```

16.275.1.8 SERIAL_PORT_STOP_BITS

```
#define SERIAL_PORT_STOP_BITS 0
```

16.275.1.9 TWO_SECOND_DELAY

```
#define TWO_SECOND_DELAY 2000
```

16.275.2 Function Documentation

16.275.2.1 CleanUp()

```
int CleanUp (
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.275.2.2 ConfigureDevice()

```
int ConfigureDevice (
    CameraPtr pCam,
    HANDLE & hFileHandle )
```

16.275.2.3 main()

```
int main (
    int ,
    char ** )
```

16.275.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.275.2.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```


16.275.2.6 SerialRx()

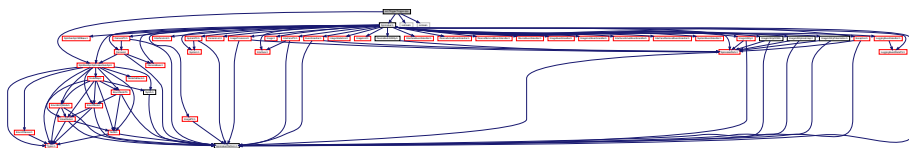
```
int SerialRx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.275.2.7 SerialTx()

```
int SerialTx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

16.276 src/Trigger/Trigger.cpp File Reference

Include dependency graph for Trigger.cpp:



Enumerations

- enum `triggerType` {
`SOFTWARE`,
`HARDWARE`,
`SOFTWARE`,
`HARDWARE` }

Functions

- int `ConfigureTrigger` (INodeMap &nodeMap)
- int `GrabNextImageByTrigger` (INodeMap &nodeMap, CameraPtr pCam)
- int `ResetTrigger` (INodeMap &nodeMap)
- int `PrintDeviceInfo` (INodeMap &nodeMap)
- int `AcquireImages` (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int `RunSingleCamera` (CameraPtr pCam)
- int `main` (int, char **)

Variables

- const `triggerType` `chosenTrigger` = `SOFTWARE`

16.276.1 Enumeration Type Documentation

16.276.1.1 triggerType

enum `triggerType`

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

16.276.2 Function Documentation

16.276.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

16.276.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

16.276.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

16.276.2.4 main()

```
int main (
    int ,
    char ** )
```

16.276.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

16.276.2.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

16.276.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

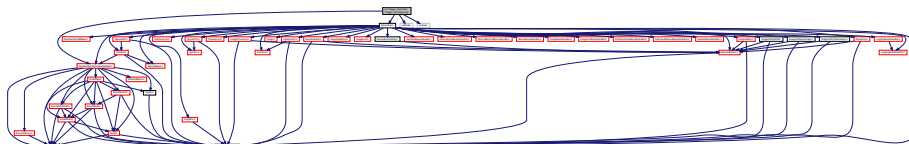
16.276.3 Variable Documentation

16.276.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

16.277 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference

Include dependency graph for Trigger_QuickSpin.cpp:



Enumerations

- enum triggerType {
 SOFTWARE,
 HARDWARE,
 SOFTWARE,
 HARDWARE }

Functions

- int ConfigureTrigger (CameraPtr pCam)
- int GrabNextImageByTrigger (CameraPtr pCam, ImagePtr &pResultImage)
- int ResetTrigger (CameraPtr pCam)
- int PrintDeviceInfo (CameraPtr pCam)
- int AcquireImages (CameraPtr pCam)
- int RunSingleCamera (CameraPtr pCam)
- int main (int, char **)

Variables

- const `triggerType chosenTrigger` = SOFTWARE

16.277.1 Enumeration Type Documentation

16.277.1.1 triggerType

enum `triggerType`

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

16.277.2 Function Documentation

16.277.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

16.277.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    CameraPtr pCam )
```

16.277.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    CameraPtr pCam,
    ImagePtr & pResultImage )
```

16.277.2.4 main()

```
int main (
    int ,
    char ** )
```

16.277.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

16.277.2.6 ResetTrigger()

```
int ResetTrigger (
    CameraPtr pCam )
```

16.277.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

16.277.3 Variable Documentation

16.277.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

Chapter 17

Example Documentation

17.1 Acquisition.cpp

[Acquisition.cpp](#) shows how to acquire images. It relies on information provided in the Enumeration example. Also, check out the ExceptionHandling and NodeMapInfo examples if you haven't already. ExceptionHandling shows the handling of standard and [Spinnaker](#) exceptions while NodeMapInfo explores retrieving information from various node types.

This example touches on the preparation and cleanup of a camera just before and just after the acquisition of images. Image retrieval and conversion, grabbing image data, and saving images are all covered as well.

Once comfortable with Acquisition, we suggest checking out AcquisitionMultipleCamera, NodeMapCallback, or SaveToAvi. AcquisitionMultipleCamera demonstrates simultaneously acquiring images from a number of cameras, NodeMapCallback serves as a good introduction to programming with callbacks and events, and SaveToAvi exhibits video creation.

17.2 AcquisitionMultipleCameraRecovery.cpp

[AcquisitionMultipleCameraRecovery.cpp](#) shows how to continuously acquire images from multiple cameras using image events. It demonstrates the use of User Set Control to save persistent camera configurations, allowing for smooth camera recovery through interface events. This example relies on information provided in the ImageEvents, EnumerationEvents, ImageFormatControl, and Acquisition examples.

This example uses a global map to retain image information, including the number of images grabbed, the number of incomplete images and the number of removals for each camera over the duration of the example. Cameras may be added or removed after the example has started.

The example assumes each camera has a unique serial number and is capable of configuring User Set 1. Note that if a camera was configured and is disconnected before the example ends, it will not be reconfigured to use the default User Set.

17.3 AcquisitionMultipleCamerasWriteToFiles.cpp

AcquisitionMultipleCamerasWriteToFiles.cpp shows how to acquire images from one or more cameras and write the images to a binary file. Thereafter, the acquired images can be retrieved from the file and saved using a desired file format supported by the current [Spinnaker](#) SDK.

This example covers most of the basics for getting started with the [Spinnaker](#) API including acquiring system objects, camera list acquisition and initialization of cameras, acquisition of images and writing to a file, image retrieval from a file, conversion and saving with desired file format is covered as well.

17.4 AcquisitionMultipleThread.cpp

[AcquisitionMultipleThread.cpp](#) shows how to capture images from multiple cameras simultaneously using threads. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example is similar to the Acquisition example, except that threads are used to allow for simultaneous acquisitions.

17.5 ActionCommand.cpp

[ActionCommand.cpp](#) shows how to send action commands to the camera(s). And it also can perform frame synchronization using multiple cameras simultaneously. It uses IEEE 1588 timestamp, triggers and action commands.

One camera will be master and the other camera(s) will be slave(s). All cameras have to be on the same network.

17.6 BufferHandling.cpp

[BufferHandling.cpp](#) shows how the different buffer handling modes work. It relies on information provided in the Acquisition and Trigger examples.

Buffer handling determines the ordering in which images are retrieved, and what occurs when an image is transmitted while the buffer is full. There are four different buffer handling modes available; NewestFirst, NewestOnly, OldestFirst and OldestFirstOverwrite.

This example explores retrieving images in a set pattern; triggering the camera while not retrieving an image (letting the buffer fill up), and retrieving images while not triggering. We cycle through the different buffer handling modes to see which images are retrieved, confirming their identities via their Frame ID values.

17.7 Compression.cpp

[Compression.cpp](#) shows how to utilize the compression features on a supported camera and in the [Spinnaker](#) SDK. It relies on information provided in the Acquisition, Enumeration, ChunkData, and NodeMapInfo examples.

This example covers all of the following: the preparation of a camera to acquire compressed images (or compressed chunk images), image retrieval, image saving, loading compressed images from disk, reconstructing compressed images, and converting compressed images.

17.8 CounterAndTimer.cpp

[CounterAndTimer.cpp](#) shows how to setup a Pulse Width Modulation (PWM) signal using counters and timers. The camera will output the PWM signal via strobe, and capture images at a rate defined by the PWM signal as well. Users should take care to use a PWM signal within the camera's max framerate (by default, the PWM signal is set to 50 Hz).

Counter and Timer functionality is only available for BFS and Oryx Cameras. For details on the hardware setup, see our kb article, "Using Counter and Timer Control"; <https://www.flir.com/support-center/iis/machine-vision/application-note/using-counter-and-timer-control>

17.9 DeviceEvents.cpp

[DeviceEvents.cpp](#) shows how to create a handler to access device events. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

Device events can be thought of as camera-related events. This example creates a user-defined class, [DeviceEventHandlerImpl](#), which allows the user to define any properties, parameters, and the event handler itself while DeviceEventHandler, the parent class, allows the child class to appropriately interface with the [Spinnaker](#) SDK.

17.10 EnumerationEvents.cpp

[EnumerationEvents.cpp](#) explores arrival and removal events on interfaces and the system. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates two user-defined classes: [InterfaceEventHandlerImpl](#) and [SystemEventHandlerImpl](#). These child classes allow the user to define properties, parameters, and the event itself while the parent classes - Device↔ArrivalEventHandler, DeviceRemovalEventHandler, and InterfaceEventHandler - allow the child classes to interface with [Spinnaker](#).

17.11 ExceptionHandling.cpp

[ExceptionHandling.cpp](#) shows the catching of an exception in [Spinnaker](#). Following this, check out the Acquisition or NodeMapInfo examples if you haven't already. Acquisition demonstrates image acquisition while NodeMapInfo explores retrieving information from various node types.

This example shows three typical paths of exception handling in [Spinnaker](#): catching the exception as a [Spinnaker](#) exception, as a standard exception, or as a standard exception which is then cast to a [Spinnaker](#) exception.

Once comfortable with Acquisition, ExceptionHandling, and NodeMapInfo, we suggest checking out Acquisition↔MultipleCamera, NodeMapCallback, or SaveToAvi. AcquisitionMultipleCamera demonstrates simultaneously acquiring images from a number of cameras, NodeMapCallback serves as a good introduction to programming with callbacks and events, and SaveToAvi exhibits video creation.

17.12 Exposure.cpp

[Exposure.cpp](#) shows how to set a custom exposure time on a device. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example shows the processes of preparing the camera, setting a custom exposure time, and restoring the camera to its default state (without power cycling). Ensuring custom values do not fall out of range is also touched on.

Following this, we suggest familiarizing yourself with the ImageFormatControl example if you haven't already. ImageFormatControl is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with Exposure and ImageFormatControl, we suggest checking out any of the longer, more complicated examples related to camera configuration: ChunkData, LookupTable, Sequencer, or Trigger.

17.13 FileAccess_Quickspin.cpp

[FileAccess_Quickspin.cpp](#) shows how to read and write images using camera File Access function. This example uploads an image to the camera File Access storage and also download the image from the camera File Access storage and saves it to the disk. It also provides debug message when debug mode is turned on giving more detail status of the progress and error messages to the users.

It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

17.14 GigEVisionPerformance.cpp

[GigEVisionPerformance.cpp](#) measures GigE Vision performance. It is built on top of Acquisition example.

This example measures CPU related performance statistics and print them out at the end.

17.15 ImageEvents.cpp

[ImageEvents.cpp](#) shows how to acquire images using the image event handler. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, [ImageEventHandlerImpl](#), that inherits from the [Spinnaker](#) class, [ImageEventHandler](#). [ImageEventHandlerImpl](#) allows the user to define any properties, parameters, and the event itself while [ImageEventHandler](#) allows the child class to appropriately interface with [Spinnaker](#).

17.16 ImageFormatControl.cpp

[ImageFormatControl.cpp](#) shows how to apply custom image settings to the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates setting minimums to offsets, X and Y, and maximums to width and height. It also shows the setting of a new pixel format, which is an enumeration type node.

Following this, we suggest familiarizing yourself with the Exposure example if you haven't already. Exposure is another example on camera customization that is shorter and simpler than many of the others. Once comfortable with Exposure and ImageFormatControl, we suggest checking out any of the longer, more complicated examples related to camera configuration: ChunkData, LookupTable, Sequencer, or Trigger.

17.17 ImageFormatControl_QuickSpin.cpp

[ImageFormatControl_QuickSpin.cpp](#) shows how to apply custom image settings to the camera using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control.

This example demonstrates customizing offsets X and Y, width and height, and the pixel format. Ensuring custom values fall within an acceptable range is also touched on. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from ImageFormatControl.

A much wider range of topics is covered in the full [Spinnaker](#) examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full [Spinnaker](#) examples for further or specific knowledge on a topic.

17.18 Inference.cpp

[Inference.cpp](#) shows how to perform the following:

- Upload custom inference neural networks to the camera (DDR or Flash)
- Inject sample test image
- Enable/Configure chunk data
- Enable/Configure trigger inference ready sync
- Acquire images
- Display inference data from acquired image chunk data
- Disable previously configured camera configurations

Inference is only available for Firefly deep learning cameras. See the related content section on the Firefly DL product page for relevant documentation.

<https://www.flir.com/products/firefly-dl/>

It can also be helpful to familiarize yourself with the Acquisition, ChunkData and FileAccess_QuickSpin examples.

17.19 Logging.cpp

[Logging.cpp](#) shows how to create a handler to access logging events. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the NodeMapCallback example, as nodemap callbacks follow the same general procedure as events, but with a few less steps.

This example creates a user-defined class, [LoggingEventHandlerImpl](#), that inherits from the [Spinnaker](#) class, [LoggingEventHandler](#). The child class allows the user to define any properties, parameters, and the event itself while [LoggingEventHandler](#) allows the child class to appropriately interface with the [Spinnaker](#) SDK.

17.20 LogicBlock.cpp

[LogicBlock.cpp](#) shows how to use logic blocks to detect missing triggers and retrigger. It relies on information provided in the Acquisition and Trigger examples.

A logic block is a collection of combinatorial logic and latches that allows users to create new, custom signals inside the camera. These custom signals can be used by the camera (for example to trigger exposure) or sent out to integrate with external systems.

Logic Block functionality is only available for BFS and Oryx Cameras. For details on logic blocks and how this example works, see our kb article, "Using Logic Blocks with Blackfly S and Oryx"; <https://www.flir.com/support-center/iis/machine-vision/application-note/using-logic-blocks-with-blackfly->

17.21 LookupTable.cpp

[LookupTable.cpp](#) shows how to configure lookup tables on the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

Lookup tables allow for the customization and control of individual pixels. This can be a very powerful and deeply useful tool; however, because use cases are context dependent, this example only explores lookup table configuration.

17.22 NodeMapCallback.cpp

[NodeMapCallback.cpp](#) shows how to use nodemap callbacks. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples. As callbacks are very similar to events, it may be a good idea to explore this example prior to tackling the events examples.

This example focuses on creating, registering, using, and unregistering callbacks. A callback requires a function signature, which allows it to be registered to and access a node. Events, while slightly more complex, follow this same pattern.

Once comfortable with NodeMapCallback, we suggest checking out any of the events examples: DeviceEvents, EnumerationEvents, ImageEvents, or Logging.

17.23 NodeMapInfo.cpp

[NodeMapInfo.cpp](#) shows how to retrieve node map information. It relies on information provided in the Enumeration example. Also, check out the Acquisition and ExceptionHandling examples if you haven't already. Acquisition demonstrates image acquisition while ExceptionHandling shows the handling of standard and [Spinnaker](#) exceptions.

This example explores retrieving information from all major node types on the camera. This includes string, integer, float, boolean, command, enumeration, category, and value types. Looping through multiple child nodes is also covered. A few node types are not covered - base, port, and register - as they are not fundamental. The final node type - enumeration entry - is explored and printed for nodes whose parent node is a selector node.

Once comfortable with NodeMapInfo, we suggest checking out ImageFormatControl and Exposure. ImageFormatControl explores customizing image settings on a camera while Exposure introduces the standard structure of configuring a device, acquiring some images, and then returning the device to a default state.

17.24 Polarization.cpp

[Polarization.cpp](#) shows how to extract and create images from a source image of Polarized8 or BayerRGPolarized8 pixel format using methods from the ImageUtilityPolarization, ImageUtility and ImageUtilityHeatmap classes. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example demonstrates some of the methods that can be used to extract polarization quadrant images and create Stokes', AoLP, and DoLP images from the ImageUtilityPolarization class. It then demonstrates how to use some of the available methods in the ImageUtility and ImageUtilityHeatmap classes to create normalized and heatmap images.

Polarization is only available for polarized cameras. For more information please visit our website; <https://www.flir.com/discover/iis/machine-vision/imaging-reflective-surfaces-sonys-first-pola>

17.25 SaveToAvi.cpp

[SaveToAvi.cpp](#) shows how to create a video from a vector of images. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

This example introduces the SpinVideo class, which is used to quickly and easily create various types of video files. It demonstrates the creation of three types: uncompressed, MJPG, and H264.

17.26 Sequencer.cpp

[Sequencer.cpp](#) shows how to use the sequencer to grab images with various settings. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples as these examples provide a strong introduction to camera customization.

The sequencer is another very powerful tool, which can be used to create and store multiple states of customized image settings. A very useful application of the sequencer is creating high dynamic range images.

This example is probably the most complex and definitely the longest. As such, the configuration has been split between three functions. The first prepares the camera to set the sequences, the second sets the settings for a single state (it is run five times), and the third configures the camera to use the sequencer when it acquires images.

17.27 Trigger.cpp

[Trigger.cpp](#) shows how to trigger the camera. It relies on information provided in the Enumeration, Acquisition, and NodeMapInfo examples.

It can also be helpful to familiarize yourself with the ImageFormatControl and Exposure examples. As they are somewhat shorter and simpler, either provides a strong introduction to camera customization.

This example shows the process of configuring, using, and cleaning up a camera for use with both a software and a hardware trigger.

17.28 Trigger_QuickSpin.cpp

@brief Trigger_QuickSpin.cpp shows how to capture images with the

trigger using the QuickSpin API. QuickSpin is a subset of the [Spinnaker](#) library that allows for simpler node access and control.

This example demonstrates how to prepare, execute, and clean up the camera in regards to using both software and hardware triggers. Retrieving and setting node values using QuickSpin is the only portion of the example that differs from Trigger.

A much wider range of topics is covered in the full Spinnaker examples than in the QuickSpin ones. There are only enough QuickSpin examples to demonstrate node access and to get started with the API; please see full Spinnaker examples for further or specific knowledge on a topic.

Index

- [_ClearXMLCache](#)
 - [CNodeMapRefT< TCameraParams >, 679](#)
 - [Spinnaker::GenApi, 363](#)
- [_Connect](#)
 - [CNodeMapRefT< TCameraParams >, 680](#)
 - [Spinnaker::GenApi, 363](#)
- [_CycleDetectAccessMode](#)
 - [Spinnaker::GenApi, 356](#)
- [_Destroy](#)
 - [CNodeMapRefT< TCameraParams >, 680](#)
- [_EAccessMode](#)
 - [Spinnaker::GenApi, 356](#)
- [_ECachingMode](#)
 - [Spinnaker::GenApi, 356](#)
- [_ECallbackType](#)
 - [Spinnaker::GenApi, 357](#)
- [_EDisplayNotation](#)
 - [Spinnaker::GenApi, 357](#)
- [_EEndianess](#)
 - [Spinnaker::GenApi, 357](#)
- [_EGenApiSchemaVersion](#)
 - [Spinnaker::GenApi, 358](#)
- [_EIncMode](#)
 - [Spinnaker::GenApi, 358](#)
- [_EInputDirection](#)
 - [Spinnaker::GenApi, 358](#)
- [_EInterfaceType](#)
 - [Spinnaker::GenApi, 359](#)
- [_ELinkType](#)
 - [Spinnaker::GenApi, 359](#)
- [_ENamespace](#)
 - [Spinnaker::GenApi, 359](#)
- [_ERepresentation](#)
 - [Spinnaker::GenApi, 360](#)
- [_ESign](#)
 - [Spinnaker::GenApi, 360](#)
- [_ESlope](#)
 - [Spinnaker::GenApi, 360](#)
- [_EStandardNamespace](#)
 - [Spinnaker::GenApi, 361](#)
- [_EVisibility](#)
 - [Spinnaker::GenApi, 361](#)
- [_EXMLValidation](#)
 - [Spinnaker::GenApi, 361](#)
- [_EYesNo](#)
 - [Spinnaker::GenApi, 362](#)
- [_GetDeviceName](#)
 - [CNodeMapRefT< TCameraParams >, 680](#)
 - [Spinnaker::GenApi, 363](#)
- [_GetNode](#)
 - [CNodeMapRefT< TCameraParams >, 680](#)
 - [Spinnaker::GenApi, 364](#)
- [_GetNodes](#)
 - [CNodeMapRefT< TCameraParams >, 680](#)
 - [Spinnaker::GenApi, 364](#)
- [_GetSupportedSchemaVersions](#)
 - [CNodeMapRefT< TCameraParams >, 681](#)
 - [Spinnaker::GenApi, 364](#)
- [_Initialize](#)
 - [CGeneric_XMLLoaderParams, 638](#)
- [_InvalidateNodes](#)
 - [CNodeMapRefT< TCameraParams >, 681](#)
 - [Spinnaker::GenApi, 364](#)
- [_LoadXMLFromFile](#)
 - [CNodeMapRefT< TCameraParams >, 681](#)
 - [Spinnaker::GenApi, 364](#)
- [_LoadXMLFromFileInject](#)
 - [CNodeMapRefT< TCameraParams >, 681](#)
 - [Spinnaker::GenApi, 364](#)
- [_LoadXMLFromString](#)
 - [CNodeMapRefT< TCameraParams >, 681](#)
 - [Spinnaker::GenApi, 364](#)
- [_LoadXMLFromStringInject](#)
 - [CNodeMapRefT< TCameraParams >, 682](#)
 - [Spinnaker::GenApi, 365](#)
- [_LoadXMLFromZIPData](#)
 - [CNodeMapRefT< TCameraParams >, 682](#)
 - [Spinnaker::GenApi, 365](#)
- [_LoadXMLFromZIPFile](#)
 - [CNodeMapRefT< TCameraParams >, 682](#)
 - [Spinnaker::GenApi, 365](#)
- [_Poll](#)
 - [CNodeMapRefT< TCameraParams >, 682](#)
 - [Spinnaker::GenApi, 365](#)
- [_Ptr](#)
 - [CNodeMapRefT< TCameraParams >, 683](#)
 - [NodeMap, 1057](#)
- [_TO_STRING](#)
 - [GCUtilities.h, 1273](#)
- [_UndefinedRepresentation](#)
 - [Types.h, 1327](#)
- [_Undefined](#)
 - [Spinnaker::GenApi, 358](#)
- [_UndefinedAccessMode](#)
 - [Spinnaker::GenApi, 356](#)
- [_UndefinedCachingMode](#)
 - [Spinnaker::GenApi, 357](#)
- [_UndefinedEDisplayNotation](#)

- Spinnaker::GenApi, 357
- _UndefinedESlope
 - Spinnaker::GenApi, 361
- _UndefinedEXMLValidation
 - Spinnaker::GenApi, 362
- _UndefinedEndian
 - Spinnaker::GenApi, 358
- _UndefinedNameSpace
 - Spinnaker::GenApi, 360
- _UndefinedRepresentation
 - Spinnaker::GenApi, 360
- _UndefinedSign
 - Spinnaker::GenApi, 360
- _UndefinedStandardNameSpace
 - Spinnaker::GenApi, 361
- _UndefinedVisibility
 - Spinnaker::GenApi, 361
- _UndefinedYesNo
 - Spinnaker::GenApi, 362
- __ERR__
 - GCUtilities.h, 1272
- __LINE_STR__
 - GCUtilities.h, 1273
- __LOCATION__
 - GCUtilities.h, 1273
- __OUTPUT_FORMATER__
 - GCUtilities.h, 1273
- __STDC_CONSTANT_MACROS
 - GCTypes.h, 1269
- __STDC_LIMIT_MACROS
 - GCTypes.h, 1269
- __TODO__
 - GCUtilities.h, 1273
- __WARN__
 - GCUtilities.h, 1273
- _enableDebug
 - FileAccess_QuickSpin.cpp, 1438
- _fileSelector
 - FileAccess_QuickSpin.cpp, 1438
- _npos
 - gcstring, 778
- _pCount
 - double_autovector_t, 728
 - int64_autovector_t, 977
- _pv
 - double_autovector_t, 728
 - int64_autovector_t, 977
- ~AutoLock
 - AutoLock, 403
- ~BasePtr
 - BasePtr< T, B >, 407
- ~BooleanNode
 - BooleanNode, 414
- ~CChunkAdapter
 - CChunkAdapter, 593
- ~CChunkAdapterDcam
 - CChunkAdapterDcam, 597
- ~CChunkAdapterGEV
 - CChunkAdapterGEV, 602
- ~CChunkAdapterGeneric
 - CChunkAdapterGeneric, 599
- ~CChunkAdapterU3V
 - CChunkAdapterU3V, 604
- ~CChunkPort
 - CChunkPort, 606
- ~CEnumerationTRef
 - CEnumerationTRef< EnumT >, 614
- ~CEventAdapter
 - CEventAdapter, 618
- ~CEventAdapter1394
 - CEventAdapter1394, 620
- ~CEventAdapterGEV
 - CEventAdapterGEV, 625
- ~CEventAdapterGeneric
 - CEventAdapterGeneric, 622
- ~CEventAdapterU3V
 - CEventAdapterU3V, 627
- ~CEventPort
 - CEventPort, 630
- ~CFeatureBag
 - CFeatureBag, 634
- ~CGlobalLock
 - CGlobalLock, 640
- ~CGlobalLockUnlocker
 - CGlobalLockUnlocker, 642
- ~CLock
 - CLock, 657, 659
- ~CNodeCallback
 - CNodeCallback, 664
- ~CNodeMapFactory
 - CNodeMapFactory, 668
- ~CNodeMapRefT
 - CNodeMapRefT< TCameraParams >, 679
- ~CPointer
 - CPointer< T, B >, 691
- ~CPortImpl
 - CPortImpl, 696
- ~CPortWriteList
 - CPortWriteList, 699
- ~CRegisterPortImpl
 - CRegisterPortImpl, 703
- ~CSelectorSet
 - CSelectorSet, 706
- ~Camera
 - Camera, 441
- ~CameraBase
 - CameraBase, 568
- ~CameraList
 - CameraList, 583
- ~CategoryNode
 - CategoryNode, 592
- ~ChunkData
 - ChunkData, 646
- ~CommandNode
 - CommandNode, 685
- ~DeviceArrivalEventHandler

- DeviceArrivalEventHandler, [714](#)
- ~DeviceEventHandler
 - DeviceEventHandler, [717](#)
- ~DeviceEventHandlerImpl
 - DeviceEventHandlerImpl, [720](#)
- ~DeviceRemovalEventHandler
 - DeviceRemovalEventHandler, [725](#)
- ~EnumEntryNode
 - EnumEntryNode, [737](#)
- ~EnumNode
 - EnumNode, [741](#)
- ~EventHandler
 - EventHandler, [750](#)
- ~Exception
 - Exception, [756](#)
- ~FileProtocolAdapter
 - FileProtocolAdapter, [760](#)
- ~FloatNode
 - FloatNode, [767](#)
- ~FloatRegNode
 - FloatRegNode, [773](#)
- ~ICameraBase
 - ICameraBase, [800](#)
- ~ICameraList
 - ICameraList, [808](#)
- ~IChunkData
 - IChunkData, [813](#)
- ~IDataStream
 - IDataStream, [822](#)
- ~IDevFileStreamBuf
 - IDevFileStreamBuf< CharType, Traits >, [831](#)
- ~IDeviceArrivalEventHandler
 - IDeviceArrivalEventHandler, [834](#)
- ~IDeviceEventHandler
 - IDeviceEventHandler, [836](#)
- ~IDeviceRemovalEventHandler
 - IDeviceRemovalEventHandler, [838](#)
- ~IImage
 - IImage, [841](#)
- ~IImageEventHandler
 - IImageEventHandler, [853](#)
- ~IImageList
 - IImageList, [855](#)
- ~IImageListEventHandler
 - IImageListEventHandler, [859](#)
- ~IImageProcessor
 - IImageProcessor, [861](#)
- ~IImageStatistics
 - IImageStatistics, [865](#)
- ~IInterface
 - IInterface, [870](#)
- ~IInterfaceArrivalEventHandler
 - IInterfaceArrivalEventHandler, [874](#)
- ~IInterfaceEventHandler
 - IInterfaceEventHandler, [876](#)
- ~IInterfaceList
 - IInterfaceList, [879](#)
- ~IInterfaceRemovalEventHandler
 - IInterfaceRemovalEventHandler, [882](#)
- ~ILoggingEventHandler
 - ILoggingEventHandler, [884](#)
- ~ISystem
 - ISystem, [1012](#)
- ~ISystemEventHandler
 - ISystemEventHandler, [1017](#)
- ~Image
 - Image, [889](#)
- ~ImageEventHandler
 - ImageEventHandler, [914](#)
- ~ImageEventHandlerImpl
 - ImageEventHandlerImpl, [916](#), [917](#)
- ~ImageList
 - ImageList, [921](#)
- ~ImageListEventHandler
 - ImageListEventHandler, [928](#)
- ~ImageProcessor
 - ImageProcessor, [933](#)
- ~ImagePtr
 - ImagePtr, [942](#)
- ~ImageStatistics
 - ImageStatistics, [944](#)
- ~InferenceBoundingBoxResult
 - InferenceBoundingBoxResult, [970](#)
- ~IntRegNode
 - IntRegNode, [1010](#)
- ~IntegerNode
 - IntegerNode, [980](#)
- ~Interface
 - Interface, [985](#)
- ~InterfaceArrivalEventHandler
 - InterfaceArrivalEventHandler, [991](#)
- ~InterfaceEventHandler
 - InterfaceEventHandler, [993](#)
- ~InterfaceEventHandlerImpl
 - InterfaceEventHandlerImpl, [997](#)
- ~InterfaceList
 - InterfaceList, [1000](#)
- ~InterfaceRemovalEventHandler
 - InterfaceRemovalEventHandler, [1007](#)
- ~Lock
 - LockableObject< Object >::Lock, [1023](#)
- ~LoggingEventData
 - LoggingEventData, [1026](#)
- ~LoggingEventHandler
 - LoggingEventHandler, [1032](#)
- ~Node
 - Node, [1041](#)
- ~NodeMap
 - NodeMap, [1052](#)
- ~ODevFileStreamBuf
 - ODevFileStreamBuf< CharType, Traits >, [1062](#)
- ~PortNode
 - PortNode, [1068](#)
- ~PortRecorder
 - PortRecorder, [1073](#)
- ~PortReplay

- PortReplay, 1077
- ~RegisterNode
 - RegisterNode, 1082
- ~SpinVideo
 - SpinVideo, 1089
- ~StringNode
 - StringNode, 1095
- ~StringRegNode
 - StringRegNode, 1099
- ~System
 - System, 1101
- ~SystemEventHandler
 - SystemEventHandler, 1111
- ~SystemEventHandlerImpl
 - SystemEventHandlerImpl, 1113
- ~SystemPtr
 - SystemPtr, 1117
- ~TransportLayerDevice
 - TransportLayerDevice, 1121
- ~TransportLayerInterface
 - TransportLayerInterface, 1132
- ~TransportLayerStream
 - TransportLayerStream, 1143
- ~TransportLayerSystem
 - TransportLayerSystem, 1152
- ~ValueNode
 - ValueNode, 1162
- ~double_autovector_t
 - double_autovector_t, 726
- ~gcstring
 - gcstring, 778
- ~int64_autovector_t
 - int64_autovector_t, 976
- AasRoiEnable
 - Camera, 442
- AasRoiHeight
 - Camera, 442
- AasRoiOffsetX
 - Camera, 442
- AasRoiOffsetY
 - Camera, 443
- AasRoiWidth
 - Camera, 443
- AcquireImages
 - Acquisition.cpp, 1365
 - AcquisitionMultipleThread.cpp, 1412
 - ActionCommand.cpp, 1414
 - BufferHandling.cpp, 1417
 - ChunkData.cpp, 1419
 - Compression.cpp, 1421
 - CounterAndTimer.cpp, 1424
 - DeviceEvents.cpp, 1426
 - Exposure.cpp, 1432
 - Exposure_QuickSpin.cpp, 1434
 - FileAccess_QuickSpin.cpp, 1435
 - GigEVisionPerformance.cpp, 1441
 - ImageEvents.cpp, 1446
 - ImageFormatControl.cpp, 1448
 - ImageFormatControl_QuickSpin.cpp, 1449
 - Inference.cpp, 1452
 - LogicBlock.cpp, 1458
 - LookupTable.cpp, 1460
 - Polarization.cpp, 1467
 - SaveToAvi.cpp, 1471
 - Sequencer.cpp, 1472
 - Trigger.cpp, 1479
 - Trigger_QuickSpin.cpp, 1481
- AcquireImagesAndSaveToFile
 - AcquisitionMultipleCamerasWriteToFile.cpp, 1410
- Acquisition.cpp
 - AcquireImages, 1365
 - ConfigureGVCPHeartbeat, 1365
 - DisableGVCPHeartbeat, 1365
 - main, 1366
 - PrintDeviceInfo, 1366
 - ResetGVCPHeartbeat, 1366
 - RunSingleCamera, 1366
- AcquisitionAbort
 - Camera, 443
- AcquisitionArm
 - Camera, 443
- AcquisitionBurstFrameCount
 - Camera, 444
- AcquisitionFrameCount
 - Camera, 444
- AcquisitionFrameRate
 - Camera, 444
- AcquisitionFrameRateEnable
 - Camera, 444
- AcquisitionLineRate
 - Camera, 444
- AcquisitionMode
 - Camera, 445
- AcquisitionMode_Continuous
 - Spinnaker, 234
- AcquisitionMode_MultiFrame
 - Spinnaker, 234
- AcquisitionMode_SingleFrame
 - Spinnaker, 234
- AcquisitionModeEnums
 - Spinnaker, 233
- AcquisitionMultipleCameraRecovery.cpp
 - cameraGrabInfoMap, 1409
 - ConfigureCamera, 1408
 - ConfigureUserSet1, 1408
 - GetDeviceSerial, 1408
 - globalCamList, 1409
 - main, 1408
 - PrintExampleStatistics, 1408
 - RefreshCameraList, 1409
 - ResetCameraUserSetToDefault, 1409
 - SleepyWrapper, 1409
- AcquisitionMultipleCamerasWriteToFile.cpp
 - AcquireImagesAndSaveToFile, 1410
 - ConfigureCameras, 1410
 - CreateFiles, 1411

- imageInfos, [1411](#)
- k_numImages, [1412](#)
- kDestinationDirectory, [1412](#)
- main, [1411](#)
- PrintDeviceInfo, [1411](#)
- RetrieveImagesFromFiles, [1411](#)
- RunCameras, [1411](#)
- AcquisitionMultipleThread.cpp
 - AcquireImages, [1412](#)
 - ConfigureGVCPHeartbeat, [1412](#)
 - DisableGVCPHeartbeat, [1413](#)
 - main, [1413](#)
 - PrintDeviceInfo, [1413](#)
 - ResetGVCPHeartbeat, [1413](#)
 - RunMultipleCameras, [1413](#)
- AcquisitionResultingFrameRate
 - Camera, [445](#)
- AcquisitionStart
 - Camera, [445](#)
- AcquisitionStatus
 - Camera, [445](#)
- AcquisitionStatusSelector
 - Camera, [445](#)
- AcquisitionStatusSelector_AcquisitionActive
 - Spinnaker, [234](#)
- AcquisitionStatusSelector_AcquisitionTransfer
 - Spinnaker, [234](#)
- AcquisitionStatusSelector_AcquisitionTriggerWait
 - Spinnaker, [234](#)
- AcquisitionStatusSelector_ExposureActive
 - Spinnaker, [234](#)
- AcquisitionStatusSelector_FrameActive
 - Spinnaker, [234](#)
- AcquisitionStatusSelector_FrameTriggerWait
 - Spinnaker, [234](#)
- AcquisitionStatusSelectorEnums
 - Spinnaker, [234](#)
- AcquisitionStop
 - Camera, [445](#)
- ActionCommand
 - TransportLayerInterface, [1133](#)
- ActionCommand.cpp
 - AcquireImages, [1414](#)
 - ConfigureActionControl, [1414](#)
 - ConfigureChunkData, [1414](#)
 - ConfigureIEEE1588, [1415](#)
 - ConfigureInterface, [1415](#)
 - ConfigureOtherNodes, [1415](#)
 - ConfigureTrigger, [1415](#)
 - main, [1415](#)
 - PrintDeviceInfo, [1415](#)
 - RunMultipleCameras, [1415](#)
 - SleepyWrapper, [1416](#)
- ActionCommandResult, [401](#)
 - DeviceAddress, [401](#)
 - Status, [401](#)
- ActionCommandStatus
 - Spinnaker, [234](#)
- ActionDeviceKey
 - Camera, [446](#)
- ActionGroupKey
 - Camera, [446](#)
- ActionGroupMask
 - Camera, [446](#)
- ActionQueueSize
 - Camera, [446](#)
- ActionSelector
 - Camera, [446](#)
- ActionUnconditionalMode
 - Camera, [446](#)
- ActionUnconditionalMode_Off
 - Spinnaker, [235](#)
- ActionUnconditionalMode_On
 - Spinnaker, [235](#)
- ActionUnconditionalModeEnums
 - Spinnaker, [235](#)
- AdaptiveCompressionEnable
 - Camera, [447](#)
- AdcBitDepth
 - Camera, [447](#)
- AdcBitDepth_Bit10
 - Spinnaker, [235](#)
- AdcBitDepth_Bit12
 - Spinnaker, [235](#)
- AdcBitDepth_Bit14
 - Spinnaker, [235](#)
- AdcBitDepth_Bit8
 - Spinnaker, [235](#)
- AdcBitDepthEnums
 - Spinnaker, [235](#)
- Add
 - CameraList, [583](#)
 - ICameraList, [808](#)
 - ImageList, [855](#)
 - InterfaceList, [879](#)
 - ImageList, [922](#)
 - InterfaceList, [1001](#)
- AddInjectionData
 - CNodeMapFactory, [670](#)
- Address
 - Spinnaker::GenApi, [388](#)
- AnnounceImage
 - IDataStream, [822](#)
- aPAUSEMACCtrlFramesReceived
 - Camera, [447](#)
- aPAUSEMACCtrlFramesTransmitted
 - Camera, [447](#)
- Append
 - CameraList, [584](#)
 - ICameraList, [809](#)
 - ImageList, [856](#)
 - InterfaceList, [879](#)
 - ImageList, [922](#)
 - InterfaceList, [1001](#)
 - SpinVideo, [1089](#)
- append

- gcstring, 778
- Application
 - CCMSettings, 611
- ApplyGamma
 - ImageProcessor, 862
 - ImageProcessor, 934
- ApplyStyleSheet
 - CNodeMapFactory, 670
- argBayerRG
 - GigEVisionPerformance.cpp, 1443
- argDuration
 - GigEVisionPerformance.cpp, 1443
- argMaxFrames
 - GigEVisionPerformance.cpp, 1443
- argNumImages
 - GigEVisionPerformance.cpp, 1443
- argPacketDelay
 - GigEVisionPerformance.cpp, 1443
- argPacketSize
 - GigEVisionPerformance.cpp, 1443
- argPrintUsage
 - GigEVisionPerformance.cpp, 1443
- argRelease
 - GigEVisionPerformance.cpp, 1444
- argUserSetFrames
 - GigEVisionPerformance.cpp, 1444
- arrayLabelClassification
 - Inference.cpp, 1455
- arrayLabelDetection
 - Inference.cpp, 1455
- assign
 - gcstring, 778, 779
- attach
 - FileProtocolAdapter, 761
- AttachBuffer
 - CChunkAdapter, 594
 - CChunkAdapterDcam, 597
 - CChunkAdapterGeneric, 600
 - CChunkAdapterGEV, 602
 - CChunkAdapterU3V, 604
 - IDataStream, 822
- AttachChunk
 - CChunkPort, 607
- AttachEvent
 - CEventPort, 630
- AttachNode
 - CEventPort, 630
- AttachNodeMap
 - CChunkAdapter, 594
 - CEventAdapter, 618
- AttachPort
 - CChunkPort, 607
- AttachStatistics_t, 402
 - NumAttachedChunks, 402
 - NumChunkPorts, 402
 - NumChunks, 402
- AutoAlgorithmSelector
 - Camera, 447
- AutoAlgorithmSelector_Ae
 - Spinnaker, 235
- AutoAlgorithmSelector_Awb
 - Spinnaker, 235
- AutoAlgorithmSelectorEnums
 - Spinnaker, 235
- AutoExposureControlLoopDamping
 - Camera, 448
- AutoExposureControlPriority
 - Camera, 448
- AutoExposureControlPriority_ExposureTime
 - Spinnaker, 236
- AutoExposureControlPriority_Gain
 - Spinnaker, 236
- AutoExposureControlPriorityEnums
 - Spinnaker, 235
- AutoExposureEVCompensation
 - Camera, 448
- AutoExposureExposureTimeLowerLimit
 - Camera, 448
- AutoExposureExposureTimeUpperLimit
 - Camera, 449
- AutoExposureGainLowerLimit
 - Camera, 449
- AutoExposureGainUpperLimit
 - Camera, 449
- AutoExposureGreyValueLowerLimit
 - Camera, 449
- AutoExposureGreyValueUpperLimit
 - Camera, 450
- AutoExposureLightingMode
 - Camera, 450
- AutoExposureLightingMode_AutoDetect
 - Spinnaker, 236
- AutoExposureLightingMode_Backlight
 - Spinnaker, 236
- AutoExposureLightingMode_Frontlight
 - Spinnaker, 236
- AutoExposureLightingMode_Normal
 - Spinnaker, 236
- AutoExposureLightingModeEnums
 - Spinnaker, 236
- AutoExposureMeteringMode
 - Camera, 450
- AutoExposureMeteringMode_Average
 - Spinnaker, 237
- AutoExposureMeteringMode_CenterWeighted
 - Spinnaker, 237
- AutoExposureMeteringMode_HistogramPeak
 - Spinnaker, 237
- AutoExposureMeteringMode_Partial
 - Spinnaker, 237
- AutoExposureMeteringMode_Spot
 - Spinnaker, 237
- AutoExposureMeteringModeEnums
 - Spinnaker, 236
- AutoExposureTargetGreyValue
 - Camera, 450

- AutoExposureTargetGreyValueAuto
 - Camera, [451](#)
- AutoExposureTargetGreyValueAuto_Continuous
 - Spinnaker, [237](#)
- AutoExposureTargetGreyValueAuto_Off
 - Spinnaker, [237](#)
- AutoExposureTargetGreyValueAutoEnums
 - Spinnaker, [237](#)
- AutoLock, [403](#)
 - ~AutoLock, [403](#)
 - AutoLock, [403](#)
- Automatic
 - Spinnaker::GenApi, [361](#)
- AutoVector Class, [101](#)
- AVI Recorder Class, [42](#)
- AVIOption, [404](#)
 - AVIOption, [404](#)
 - frameRate, [404](#)
 - height, [405](#)
 - reserved, [405](#)
 - width, [405](#)
- BalanceRatio
 - Camera, [451](#)
- BalanceRatioSelector
 - Camera, [451](#)
- BalanceRatioSelector_Blue
 - Spinnaker, [237](#)
- BalanceRatioSelector_Red
 - Spinnaker, [237](#)
- BalanceRatioSelectorEnums
 - Spinnaker, [237](#)
- BalanceWhiteAuto
 - Camera, [451](#)
- BalanceWhiteAuto_Continuous
 - Spinnaker, [238](#)
- BalanceWhiteAuto_Off
 - Spinnaker, [238](#)
- BalanceWhiteAuto_Once
 - Spinnaker, [238](#)
- BalanceWhiteAutoDamping
 - Camera, [452](#)
- BalanceWhiteAutoEnums
 - Spinnaker, [237](#)
- BalanceWhiteAutoLowerLimit
 - Camera, [452](#)
- BalanceWhiteAutoProfile
 - Camera, [452](#)
- BalanceWhiteAutoProfile_Indoor
 - Spinnaker, [238](#)
- BalanceWhiteAutoProfile_Outdoor
 - Spinnaker, [238](#)
- BalanceWhiteAutoProfileEnums
 - Spinnaker, [238](#)
- BalanceWhiteAutoUpperLimit
 - Camera, [452](#)
- BasePtr
 - BasePtr< T, B >, [407](#)
- BasePtr Class, [43](#)
- BasePtr< T, B >, [405](#)
 - ~BasePtr, [407](#)
 - BasePtr, [407](#)
 - CameraListImpl, [410](#)
 - get, [407](#)
 - InterfaceListImpl, [410](#)
 - IsValid, [407](#)
 - m_pT, [411](#)
 - operator bool, [408](#)
 - operator T*, [408](#)
 - operator!=, [408](#)
 - operator->, [409](#)
 - operator=, [409](#)
 - operator==, [409](#), [410](#)
- BeginAcquisition
 - CameraBase, [568](#)
 - ICameraBase, [800](#)
 - SpinnakerDirectShow.h, [1336](#)
- Beginner
 - Spinnaker::GenApi, [361](#)
- BigEndian
 - Spinnaker::GenApi, [358](#)
- binaryFile
 - PGMOption, [1064](#)
 - PPMOption, [1080](#)
- BinningHorizontal
 - Camera, [453](#)
- BinningHorizontalMode
 - Camera, [453](#)
- BinningHorizontalMode_Average
 - Spinnaker, [238](#)
- BinningHorizontalMode_Sum
 - Spinnaker, [238](#)
- BinningHorizontalModeEnums
 - Spinnaker, [238](#)
- BinningSelector
 - Camera, [453](#)
- BinningSelector_All
 - Spinnaker, [239](#)
- BinningSelector_ISP
 - Spinnaker, [239](#)
- BinningSelector_Sensor
 - Spinnaker, [239](#)
- BinningSelectorEnums
 - Spinnaker, [238](#)
- BinningVertical
 - Camera, [453](#)
- BinningVerticalMode
 - Camera, [454](#)
- BinningVerticalMode_Average
 - Spinnaker, [239](#)
- BinningVerticalMode_Sum
 - Spinnaker, [239](#)
- BinningVerticalModeEnums
 - Spinnaker, [239](#)
- bitrate
 - H264Option, [797](#)
- BlackLevel

- Camera, [454](#)
- BlackLevelAuto
 - Camera, [454](#)
- BlackLevelAuto_Continuous
 - Spinnaker, [240](#)
- BlackLevelAuto_Off
 - Spinnaker, [240](#)
- BlackLevelAuto_Once
 - Spinnaker, [240](#)
- BlackLevelAutoBalance
 - Camera, [454](#)
- BlackLevelAutoBalance_Continuous
 - Spinnaker, [239](#)
- BlackLevelAutoBalance_Off
 - Spinnaker, [239](#)
- BlackLevelAutoBalance_Once
 - Spinnaker, [239](#)
- BlackLevelAutoBalanceEnums
 - Spinnaker, [239](#)
- BlackLevelAutoEnums
 - Spinnaker, [239](#)
- BlackLevelClampingEnable
 - Camera, [454](#)
- BlackLevelRaw
 - Camera, [455](#)
- BlackLevelSelector
 - Camera, [455](#)
- BlackLevelSelector_All
 - Spinnaker, [240](#)
- BlackLevelSelector_Analog
 - Spinnaker, [240](#)
- BlackLevelSelector_Digital
 - Spinnaker, [240](#)
- BlackLevelSelectorEnums
 - Spinnaker, [240](#)
- BlockId
 - GVCP_EVENT_ITEM, [787](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [789](#)
- BlockId64High
 - GVCP_EVENT_ITEM_EXTENDED_ID, [789](#)
- BlockId64Low
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
- BMPOption, [411](#)
 - BMPOption, [411](#)
 - indexedColor_8bit, [411](#)
 - reserved, [412](#)
- Boolean
 - Spinnaker::GenApi, [360](#)
- BooleanGetValue
 - SpinnakerDirectShow.h, [1336](#)
- BooleanNode, [412](#)
 - ~BooleanNode, [414](#)
 - BooleanNode, [413](#), [414](#)
 - GetValue, [414](#)
 - operator=, [414](#)
 - SetReference, [414](#)
 - SetValue, [415](#)
- BooleanNode Class, [104](#)
- BooleanSetValue
 - SpinnakerDirectShow.h, [1338](#)
- bottomRightXCoord
 - InferenceBoxRect, [973](#)
 - InferenceBoxRotatedRect, [974](#)
- bottomRightYCoord
 - InferenceBoxRect, [973](#)
 - InferenceBoxRotatedRect, [974](#)
- boxType
 - InferenceBoundingBox, [968](#)
- BufferHandling.cpp
 - AcquireImages, [1417](#)
 - ConfigureTrigger, [1417](#)
 - GrabNextImageByTrigger, [1417](#)
 - k_numLoops, [1416](#)
 - main, [1417](#)
 - numBuffers, [1417](#)
 - PrintDeviceInfo, [1418](#)
 - ResetTrigger, [1418](#)
 - RunSingleCamera, [1418](#)
 - SleepyWrapper, [1418](#)
 - z_numTriggers, [1417](#)
- BufferOwnership
 - Spinnaker, [240](#)
- bufferSize
 - SpinnakerDirectShow.h, [1351](#)
- build
 - LibraryVersion, [1021](#)
- c_str
 - gcstring, [779](#)
- CacheChunkData
 - PortNode, [1068](#)
 - Spinnaker::GenApi, [365](#)
- CacheUsage_Automatic
 - Spinnaker::GenApi, [362](#)
- CacheUsage_ForceRead
 - Spinnaker::GenApi, [362](#)
- CacheUsage_ForceWrite
 - Spinnaker::GenApi, [362](#)
- CacheUsage_Ignore
 - Spinnaker::GenApi, [362](#)
- CalculateStatistics
 - Image, [841](#)
 - Image, [890](#)
- CallbackHandlerInternal
 - EventHandler, [751](#)
- CallbackHandleType
 - Spinnaker::GenApi, [353](#)
- Camera, [415](#)
 - ~Camera, [441](#)
 - AasRoiEnable, [442](#)
 - AasRoiHeight, [442](#)
 - AasRoiOffsetX, [442](#)
 - AasRoiOffsetY, [443](#)
 - AasRoiWidth, [443](#)
 - AcquisitionAbort, [443](#)
 - AcquisitionArm, [443](#)
 - AcquisitionBurstFrameCount, [444](#)

- AcquisitionFrameCount, [444](#)
- AcquisitionFrameRate, [444](#)
- AcquisitionFrameRateEnable, [444](#)
- AcquisitionLineRate, [444](#)
- AcquisitionMode, [445](#)
- AcquisitionResultingFrameRate, [445](#)
- AcquisitionStart, [445](#)
- AcquisitionStatus, [445](#)
- AcquisitionStatusSelector, [445](#)
- AcquisitionStop, [445](#)
- ActionDeviceKey, [446](#)
- ActionGroupKey, [446](#)
- ActionGroupMask, [446](#)
- ActionQueueSize, [446](#)
- ActionSelector, [446](#)
- ActionUnconditionalMode, [446](#)
- AdaptiveCompressionEnable, [447](#)
- AdcBitDepth, [447](#)
- aPAUSEMACCtrlFramesReceived, [447](#)
- aPAUSEMACCtrlFramesTransmitted, [447](#)
- AutoAlgorithmSelector, [447](#)
- AutoExposureControlLoopDamping, [448](#)
- AutoExposureControlPriority, [448](#)
- AutoExposureEVCompensation, [448](#)
- AutoExposureExposureTimeLowerLimit, [448](#)
- AutoExposureExposureTimeUpperLimit, [449](#)
- AutoExposureGainLowerLimit, [449](#)
- AutoExposureGainUpperLimit, [449](#)
- AutoExposureGreyValueLowerLimit, [449](#)
- AutoExposureGreyValueUpperLimit, [450](#)
- AutoExposureLightingMode, [450](#)
- AutoExposureMeteringMode, [450](#)
- AutoExposureTargetGreyValue, [450](#)
- AutoExposureTargetGreyValueAuto, [451](#)
- BalanceRatio, [451](#)
- BalanceRatioSelector, [451](#)
- BalanceWhiteAuto, [451](#)
- BalanceWhiteAutoDamping, [452](#)
- BalanceWhiteAutoLowerLimit, [452](#)
- BalanceWhiteAutoProfile, [452](#)
- BalanceWhiteAutoUpperLimit, [452](#)
- BinningHorizontal, [453](#)
- BinningHorizontalMode, [453](#)
- BinningSelector, [453](#)
- BinningVertical, [453](#)
- BinningVerticalMode, [454](#)
- BlackLevel, [454](#)
- BlackLevelAuto, [454](#)
- BlackLevelAutoBalance, [454](#)
- BlackLevelClampingEnable, [454](#)
- BlackLevelRaw, [455](#)
- BlackLevelSelector, [455](#)
- Camera, [442](#)
- ChunkBlackLevel, [455](#)
- ChunkBlackLevelSelector, [455](#)
- ChunkCompressionMode, [456](#)
- ChunkCompressionRatio, [456](#)
- ChunkCounterSelector, [456](#)
- ChunkCounterValue, [456](#)
- ChunkCRC, [456](#)
- ChunkEnable, [456](#)
- ChunkEncoderSelector, [457](#)
- ChunkEncoderStatus, [457](#)
- ChunkEncoderValue, [457](#)
- ChunkExposureEndLineStatusAll, [457](#)
- ChunkExposureTime, [457](#)
- ChunkExposureTimeSelector, [457](#)
- ChunkFrameID, [458](#)
- ChunkGain, [458](#)
- ChunkGainSelector, [458](#)
- ChunkHeight, [458](#)
- ChunkImage, [458](#)
- ChunkImageComponent, [458](#)
- ChunkInferenceBoundingBoxResult, [459](#)
- ChunkInferenceConfidence, [459](#)
- ChunkInferenceFrameId, [459](#)
- ChunkInferenceResult, [459](#)
- ChunkLinePitch, [459](#)
- ChunkLineStatusAll, [459](#)
- ChunkModeActive, [460](#)
- ChunkOffsetX, [460](#)
- ChunkOffsetY, [460](#)
- ChunkPartSelector, [460](#)
- ChunkPixelDynamicRangeMax, [460](#)
- ChunkPixelDynamicRangeMin, [460](#)
- ChunkPixelFormat, [461](#)
- ChunkRegionID, [461](#)
- ChunkScan3dAxisMax, [461](#)
- ChunkScan3dAxisMin, [461](#)
- ChunkScan3dCoordinateOffset, [461](#)
- ChunkScan3dCoordinateReferenceSelector, [461](#)
- ChunkScan3dCoordinateReferenceValue, [462](#)
- ChunkScan3dCoordinateScale, [462](#)
- ChunkScan3dCoordinateSelector, [462](#)
- ChunkScan3dCoordinateSystem, [462](#)
- ChunkScan3dCoordinateSystemReference, [462](#)
- ChunkScan3dCoordinateTransformSelector, [462](#)
- ChunkScan3dDistanceUnit, [463](#)
- ChunkScan3dInvalidDataFlag, [463](#)
- ChunkScan3dInvalidDataValue, [463](#)
- ChunkScan3dOutputMode, [463](#)
- ChunkScan3dTransformValue, [463](#)
- ChunkScanLineSelector, [463](#)
- ChunkSelector, [464](#)
- ChunkSequencerSetActive, [464](#)
- ChunkSerialData, [464](#)
- ChunkSerialDataLength, [464](#)
- ChunkSerialReceiveOverflow, [464](#)
- ChunkSourceID, [464](#)
- ChunkStreamChannelID, [465](#)
- ChunkTimerSelector, [465](#)
- ChunkTimerValue, [465](#)
- ChunkTimestamp, [465](#)
- ChunkTimestampLatchValue, [465](#)
- ChunkTransferBlockID, [465](#)
- ChunkTransferQueueCurrentBlockCount, [466](#)

- ChunkTransferStreamID, 466
- ChunkWidth, 466
- CICongfiguration, 466
- CITimeSlotsCount, 466
- ColorTransformationEnable, 466
- ColorTransformationSelector, 467
- ColorTransformationValue, 467
- ColorTransformationValueSelector, 467
- CompressionRatio, 467
- CompressionSaturationPriority, 468
- CounterDelay, 468
- CounterDuration, 468
- CounterEventActivation, 468
- CounterEventSource, 468
- CounterReset, 468
- CounterResetActivation, 469
- CounterResetSource, 469
- CounterSelector, 469
- CounterStatus, 469
- CounterTriggerActivation, 469
- CounterTriggerSource, 469
- CounterValue, 470
- CounterValueAtReset, 470
- CxpConnectionSelector, 470
- CxpConnectionTestErrorCount, 470
- CxpConnectionTestMode, 470
- CxpConnectionTestPacketCount, 470
- CxpLinkConfiguration, 471
- CxpLinkConfigurationPreferred, 471
- CxpLinkConfigurationStatus, 471
- CxpPoCxpAuto, 471
- CxpPoCxpStatus, 471
- CxpPoCxpTripReset, 471
- CxpPoCxpTurnOff, 472
- DecimationHorizontal, 472
- DecimationHorizontalMode, 472
- DecimationSelector, 472
- DecimationVertical, 472
- DecimationVerticalMode, 473
- DefectCorrectionMode, 473
- DefectCorrectStaticEnable, 473
- DefectTableApply, 473
- DefectTableCoordinateX, 474
- DefectTableCoordinateY, 474
- DefectTableFactoryRestore, 474
- DefectTableIndex, 474
- DefectTablePixelCount, 475
- DefectTableSave, 475
- Deinterlacing, 475
- DeviceCharacterSet, 475
- DeviceClockFrequency, 476
- DeviceClockSelector, 476
- DeviceConnectionSelector, 476
- DeviceConnectionSpeed, 476
- DeviceConnectionStatus, 476
- DeviceEventChannelCount, 477
- DeviceFamilyName, 477
- DeviceFeaturePersistenceEnd, 477
- DeviceFeaturePersistenceStart, 477
- DeviceFirmwareVersion, 477
- DeviceGenCPVersionMajor, 478
- DeviceGenCPVersionMinor, 478
- DeviceID, 478
- DeviceIndicatorMode, 478
- DeviceLinkBandwidthReserve, 478
- DeviceLinkCommandTimeout, 478
- DeviceLinkConnectionCount, 479
- DeviceLinkCurrentThroughput, 479
- DeviceLinkHeartbeatMode, 479
- DeviceLinkHeartbeatTimeout, 479
- DeviceLinkSelector, 479
- DeviceLinkSpeed, 479
- DeviceLinkThroughputLimit, 480
- DeviceLinkThroughputLimitMode, 480
- DeviceManifestEntrySelector, 480
- DeviceManifestPrimaryURL, 480
- DeviceManifestSchemaMajorVersion, 480
- DeviceManifestSchemaMinorVersion, 481
- DeviceManifestSecondaryURL, 481
- DeviceManifestXMLMajorVersion, 481
- DeviceManifestXMLMinorVersion, 481
- DeviceManifestXMLSubMinorVersion, 481
- DeviceManufacturerInfo, 481
- DeviceMaxThroughput, 482
- DeviceModelName, 482
- DevicePowerSupplySelector, 482
- DeviceRegistersCheck, 482
- DeviceRegistersEndianness, 482
- DeviceRegistersStreamingEnd, 483
- DeviceRegistersStreamingStart, 483
- DeviceRegistersValid, 483
- DeviceReset, 483
- DeviceScanType, 483
- DeviceSerialNumber, 483
- DeviceSerialPortBaudRate, 484
- DeviceSerialPortSelector, 484
- DeviceSFNCVersionMajor, 484
- DeviceSFNCVersionMinor, 484
- DeviceSFNCVersionSubMinor, 484
- DeviceStreamChannelCount, 485
- DeviceStreamChannelEndianness, 485
- DeviceStreamChannelLink, 485
- DeviceStreamChannelPacketSize, 485
- DeviceStreamChannelSelector, 485
- DeviceStreamChannelType, 486
- DeviceTapGeometry, 486
- DeviceTemperature, 486
- DeviceTemperatureSelector, 486
- DeviceTLType, 486
- DeviceTLVersionMajor, 487
- DeviceTLVersionMinor, 487
- DeviceTLVersionSubMinor, 487
- DeviceType, 487
- DeviceUptime, 487
- DeviceUserID, 488
- DeviceVendorName, 488

- DeviceVersion, [488](#)
- EncoderDivider, [488](#)
- EncoderMode, [488](#)
- EncoderOutputMode, [488](#)
- EncoderReset, [489](#)
- EncoderResetActivation, [489](#)
- EncoderResetSource, [489](#)
- EncoderSelector, [489](#)
- EncoderSourceA, [489](#)
- EncoderSourceB, [489](#)
- EncoderStatus, [490](#)
- EncoderTimeout, [490](#)
- EncoderValue, [490](#)
- EncoderValueAtReset, [490](#)
- EnumerationCount, [490](#)
- EventAcquisitionEnd, [490](#)
- EventAcquisitionEndFrameID, [491](#)
- EventAcquisitionEndTimestamp, [491](#)
- EventAcquisitionError, [491](#)
- EventAcquisitionErrorFrameID, [491](#)
- EventAcquisitionErrorTimestamp, [491](#)
- EventAcquisitionStart, [491](#)
- EventAcquisitionStartFrameID, [492](#)
- EventAcquisitionStartTimestamp, [492](#)
- EventAcquisitionTransferEnd, [492](#)
- EventAcquisitionTransferEndFrameID, [492](#)
- EventAcquisitionTransferEndTimestamp, [492](#)
- EventAcquisitionTransferStart, [492](#)
- EventAcquisitionTransferStartFrameID, [493](#)
- EventAcquisitionTransferStartTimestamp, [493](#)
- EventAcquisitionTrigger, [493](#)
- EventAcquisitionTriggerFrameID, [493](#)
- EventAcquisitionTriggerTimestamp, [493](#)
- EventActionLate, [493](#)
- EventActionLateFrameID, [494](#)
- EventActionLateTimestamp, [494](#)
- EventCounter0End, [494](#)
- EventCounter0EndFrameID, [494](#)
- EventCounter0EndTimestamp, [494](#)
- EventCounter0Start, [494](#)
- EventCounter0StartFrameID, [495](#)
- EventCounter0StartTimestamp, [495](#)
- EventCounter1End, [495](#)
- EventCounter1EndFrameID, [495](#)
- EventCounter1EndTimestamp, [495](#)
- EventCounter1Start, [495](#)
- EventCounter1StartFrameID, [496](#)
- EventCounter1StartTimestamp, [496](#)
- EventEncoder0Restarted, [496](#)
- EventEncoder0RestartedFrameID, [496](#)
- EventEncoder0RestartedTimestamp, [496](#)
- EventEncoder0Stopped, [496](#)
- EventEncoder0StoppedFrameID, [497](#)
- EventEncoder0StoppedTimestamp, [497](#)
- EventEncoder1Restarted, [497](#)
- EventEncoder1RestartedFrameID, [497](#)
- EventEncoder1RestartedTimestamp, [497](#)
- EventEncoder1Stopped, [497](#)
- EventEncoder1StoppedFrameID, [498](#)
- EventEncoder1StoppedTimestamp, [498](#)
- EventError, [498](#)
- EventErrorCode, [498](#)
- EventErrorFrameID, [498](#)
- EventErrorTimestamp, [498](#)
- EventExposureEnd, [499](#)
- EventExposureEndFrameID, [499](#)
- EventExposureEndTimestamp, [499](#)
- EventExposureStart, [499](#)
- EventExposureStartFrameID, [499](#)
- EventExposureStartTimestamp, [499](#)
- EventFrameBurstEnd, [500](#)
- EventFrameBurstEndFrameID, [500](#)
- EventFrameBurstEndTimestamp, [500](#)
- EventFrameBurstStart, [500](#)
- EventFrameBurstStartFrameID, [500](#)
- EventFrameBurstStartTimestamp, [500](#)
- EventFrameEnd, [501](#)
- EventFrameEndFrameID, [501](#)
- EventFrameEndTimestamp, [501](#)
- EventFrameStart, [501](#)
- EventFrameStartFrameID, [501](#)
- EventFrameStartTimestamp, [501](#)
- EventFrameTransferEnd, [502](#)
- EventFrameTransferEndFrameID, [502](#)
- EventFrameTransferEndTimestamp, [502](#)
- EventFrameTransferStart, [502](#)
- EventFrameTransferStartFrameID, [502](#)
- EventFrameTransferStartTimestamp, [502](#)
- EventFrameTrigger, [503](#)
- EventFrameTriggerFrameID, [503](#)
- EventFrameTriggerTimestamp, [503](#)
- EventLine0AnyEdge, [503](#)
- EventLine0AnyEdgeFrameID, [503](#)
- EventLine0AnyEdgeTimestamp, [503](#)
- EventLine0FallingEdge, [504](#)
- EventLine0FallingEdgeFrameID, [504](#)
- EventLine0FallingEdgeTimestamp, [504](#)
- EventLine0RisingEdge, [504](#)
- EventLine0RisingEdgeFrameID, [504](#)
- EventLine0RisingEdgeTimestamp, [504](#)
- EventLine1AnyEdge, [505](#)
- EventLine1AnyEdgeFrameID, [505](#)
- EventLine1AnyEdgeTimestamp, [505](#)
- EventLine1FallingEdge, [505](#)
- EventLine1FallingEdgeFrameID, [505](#)
- EventLine1FallingEdgeTimestamp, [505](#)
- EventLine1RisingEdge, [506](#)
- EventLine1RisingEdgeFrameID, [506](#)
- EventLine1RisingEdgeTimestamp, [506](#)
- EventLinkSpeedChange, [506](#)
- EventLinkSpeedChangeFrameID, [506](#)
- EventLinkSpeedChangeTimestamp, [506](#)
- EventLinkTrigger0, [507](#)
- EventLinkTrigger0FrameID, [507](#)
- EventLinkTrigger0Timestamp, [507](#)
- EventLinkTrigger1, [507](#)

- EventLinkTrigger1FrameID, [507](#)
- EventLinkTrigger1Timestamp, [507](#)
- EventNotification, [508](#)
- EventSelector, [508](#)
- EventSequencerSetChange, [508](#)
- EventSequencerSetChangeFrameID, [508](#)
- EventSequencerSetChangeTimestamp, [508](#)
- EventSerialData, [508](#)
- EventSerialDataLength, [509](#)
- EventSerialPortReceive, [509](#)
- EventSerialPortReceiveTimestamp, [509](#)
- EventSerialReceiveOverflow, [509](#)
- EventStream0TransferBlockEnd, [509](#)
- EventStream0TransferBlockEndFrameID, [509](#)
- EventStream0TransferBlockEndTimestamp, [510](#)
- EventStream0TransferBlockStart, [510](#)
- EventStream0TransferBlockStartFrameID, [510](#)
- EventStream0TransferBlockStartTimestamp, [510](#)
- EventStream0TransferBlockTrigger, [510](#)
- EventStream0TransferBlockTriggerFrameID, [510](#)
- EventStream0TransferBlockTriggerTimestamp, [511](#)
- EventStream0TransferBurstEnd, [511](#)
- EventStream0TransferBurstEndFrameID, [511](#)
- EventStream0TransferBurstEndTimestamp, [511](#)
- EventStream0TransferBurstStart, [511](#)
- EventStream0TransferBurstStartFrameID, [511](#)
- EventStream0TransferBurstStartTimestamp, [512](#)
- EventStream0TransferEnd, [512](#)
- EventStream0TransferEndFrameID, [512](#)
- EventStream0TransferEndTimestamp, [512](#)
- EventStream0TransferOverflow, [512](#)
- EventStream0TransferOverflowFrameID, [512](#)
- EventStream0TransferOverflowTimestamp, [513](#)
- EventStream0TransferPause, [513](#)
- EventStream0TransferPauseFrameID, [513](#)
- EventStream0TransferPauseTimestamp, [513](#)
- EventStream0TransferResume, [513](#)
- EventStream0TransferResumeFrameID, [513](#)
- EventStream0TransferResumeTimestamp, [514](#)
- EventStream0TransferStart, [514](#)
- EventStream0TransferStartFrameID, [514](#)
- EventStream0TransferStartTimestamp, [514](#)
- EventTest, [514](#)
- EventTestTimestamp, [514](#)
- EventTimer0End, [515](#)
- EventTimer0EndFrameID, [515](#)
- EventTimer0EndTimestamp, [515](#)
- EventTimer0Start, [515](#)
- EventTimer0StartFrameID, [515](#)
- EventTimer0StartTimestamp, [515](#)
- EventTimer1End, [516](#)
- EventTimer1EndFrameID, [516](#)
- EventTimer1EndTimestamp, [516](#)
- EventTimer1Start, [516](#)
- EventTimer1StartFrameID, [516](#)
- EventTimer1StartTimestamp, [516](#)
- ExposureActiveMode, [517](#)
- ExposureAuto, [517](#)
- ExposureMode, [517](#)
- ExposureTime, [517](#)
- ExposureTimeMode, [517](#)
- ExposureTimeSelector, [518](#)
- FactoryReset, [518](#)
- FileAccessBuffer, [518](#)
- FileAccessLength, [518](#)
- FileAccessOffset, [518](#)
- FileOpenMode, [518](#)
- FileOperationExecute, [519](#)
- FileOperationResult, [519](#)
- FileOperationSelector, [519](#)
- FileOperationStatus, [519](#)
- FileSelector, [520](#)
- FileSize, [520](#)
- Gain, [520](#)
- GainAuto, [520](#)
- GainAutoBalance, [520](#)
- GainSelector, [521](#)
- Gamma, [521](#)
- GammaEnable, [521](#)
- GevActiveLinkCount, [521](#)
- GevCCP, [521](#)
- GevCurrentDefaultGateway, [521](#)
- GevCurrentIPAddress, [522](#)
- GevCurrentIPConfigurationDHCP, [522](#)
- GevCurrentIPConfigurationLLA, [522](#)
- GevCurrentIPConfigurationPersistentIP, [522](#)
- GevCurrentPhysicalLinkConfiguration, [522](#)
- GevCurrentSubnetMask, [522](#)
- GevDiscoveryAckDelay, [523](#)
- GevFirstURL, [523](#)
- GevGVCPExtendedStatusCodes, [523](#)
- GevGVCPExtendedStatusCodesSelector, [523](#)
- GevGVCPHeartbeatDisable, [523](#)
- GevGVCPPendingAck, [523](#)
- GevGVCPPendingTimeout, [524](#)
- GevGVSPExtendedIDMode, [524](#)
- GevHeartbeatTimeout, [524](#)
- GevIEEE1588, [524](#)
- GevIEEE1588ClockAccuracy, [524](#)
- GevIEEE1588Mode, [524](#)
- GevIEEE1588Status, [525](#)
- GevInterfaceSelector, [525](#)
- GevIPConfigurationStatus, [525](#)
- GevMACAddress, [525](#)
- GevMCDA, [525](#)
- GevMCPHostPort, [525](#)
- GevMCRC, [526](#)
- GevMCSP, [526](#)
- GevMCTT, [526](#)
- GevNumberOfInterfaces, [526](#)
- GevPAUSEFrameReception, [526](#)
- GevPAUSEFrameTransmission, [526](#)
- GevPersistentDefaultGateway, [527](#)
- GevPersistentIPAddress, [527](#)
- GevPersistentSubnetMask, [527](#)
- GevPhysicalLinkConfiguration, [527](#)

- GevPrimaryApplicationIPAddress, [527](#)
- GevPrimaryApplicationSocket, [527](#)
- GevPrimaryApplicationSwitchoverKey, [528](#)
- GevSCCFGAllInTransmission, [528](#)
- GevSCCFGExtendedChunkData, [528](#)
- GevSCCFGPacketResendDestination, [528](#)
- GevSCCFGUnconditionalStreaming, [528](#)
- GevSCDA, [528](#)
- GevSCPD, [529](#)
- GevSCPDirection, [529](#)
- GevSCPHostPort, [529](#)
- GevSCPInterfaceIndex, [529](#)
- GevSCPSBigEndian, [529](#)
- GevSCPSDoNotFragment, [529](#)
- GevSCPSFireTestPacket, [530](#)
- GevSCPSPacketSize, [530](#)
- GevSCSP, [530](#)
- GevSCZoneConfigurationLock, [530](#)
- GevSCZoneCount, [530](#)
- GevSCZoneDirectionAll, [530](#)
- GevSecondURL, [531](#)
- GevStreamChannelSelector, [531](#)
- GevSupportedOption, [531](#)
- GevSupportedOptionSelector, [531](#)
- GevTimestampTickFrequency, [531](#)
- GuiXmlManifestAddress, [531](#)
- Height, [532](#)
- HeightMax, [532](#)
- ImageComponentEnable, [532](#)
- ImageComponentSelector, [532](#)
- ImageCompressionBitrate, [532](#)
- ImageCompressionJPEGFormatOption, [533](#)
- ImageCompressionMode, [533](#)
- ImageCompressionQuality, [533](#)
- ImageCompressionRateOption, [533](#)
- Init, [442](#)
- IspEnable, [533](#)
- LineFilterWidth, [533](#)
- LineFormat, [534](#)
- LineInputFilterSelector, [534](#)
- LineInverter, [534](#)
- LineMode, [534](#)
- LinePitch, [534](#)
- LineSelector, [534](#)
- LineSource, [535](#)
- LineStatus, [535](#)
- LineStatusAll, [535](#)
- LinkErrorCount, [535](#)
- LinkUptime, [535](#)
- LogicBlockLUTInputActivation, [535](#)
- LogicBlockLUTInputSelector, [536](#)
- LogicBlockLUTInputSource, [536](#)
- LogicBlockLUTOutputValue, [536](#)
- LogicBlockLUTOutputValueAll, [536](#)
- LogicBlockLUTRowIndex, [536](#)
- LogicBlockLUTSelector, [536](#)
- LogicBlockSelector, [537](#)
- LUTEnable, [537](#)
- LUTIndex, [537](#)
- LUTSelector, [537](#)
- LUTValue, [537](#)
- LUTValueAll, [538](#)
- MaxDeviceResetTime, [538](#)
- OffsetX, [538](#)
- OffsetY, [538](#)
- PacketResendRequestCount, [538](#)
- PayloadSize, [539](#)
- PixelColorFilter, [539](#)
- PixelDynamicRangeMax, [539](#)
- PixelDynamicRangeMin, [539](#)
- PixelFormat, [539](#)
- PixelFormatInfoID, [540](#)
- PixelFormatInfoSelector, [540](#)
- PixelSize, [540](#)
- PowerSupplyCurrent, [540](#)
- PowerSupplyVoltage, [540](#)
- RegionDestination, [541](#)
- RegionMode, [541](#)
- RegionSelector, [541](#)
- ReverseX, [541](#)
- ReverseY, [541](#)
- RgbTransformLightSource, [542](#)
- Saturation, [542](#)
- SaturationEnable, [542](#)
- Scan3dAxisMax, [542](#)
- Scan3dAxisMin, [542](#)
- Scan3dCoordinateOffset, [543](#)
- Scan3dCoordinateReferenceSelector, [543](#)
- Scan3dCoordinateReferenceValue, [543](#)
- Scan3dCoordinateScale, [543](#)
- Scan3dCoordinateSelector, [543](#)
- Scan3dCoordinateSystem, [543](#)
- Scan3dCoordinateSystemReference, [544](#)
- Scan3dCoordinateTransformSelector, [544](#)
- Scan3dDistanceUnit, [544](#)
- Scan3dInvalidDataFlag, [544](#)
- Scan3dInvalidDataValue, [544](#)
- Scan3dOutputMode, [544](#)
- Scan3dTransformValue, [545](#)
- SensorDescription, [545](#)
- SensorDigitizationTaps, [545](#)
- SensorHeight, [545](#)
- SensorShutterMode, [545](#)
- SensorTaps, [545](#)
- SensorWidth, [546](#)
- SequencerConfigurationMode, [546](#)
- SequencerConfigurationValid, [546](#)
- SequencerFeatureEnable, [546](#)
- SequencerMode, [546](#)
- SequencerPathSelector, [547](#)
- SequencerSetActive, [547](#)
- SequencerSetLoad, [547](#)
- SequencerSetNext, [547](#)
- SequencerSetSave, [547](#)
- SequencerSetSelector, [548](#)
- SequencerSetStart, [548](#)

- SequencerSetValid, [548](#)
- SequencerTriggerActivation, [548](#)
- SequencerTriggerSource, [549](#)
- SerialPortBaudRate, [549](#)
- SerialPortDataBits, [549](#)
- SerialPortParity, [549](#)
- SerialPortSelector, [549](#)
- SerialPortSource, [550](#)
- SerialPortStopBits, [550](#)
- SerialReceiveFramingErrorCount, [550](#)
- SerialReceiveParityErrorCount, [550](#)
- SerialReceiveQueueClear, [550](#)
- SerialReceiveQueueCurrentCharacterCount, [550](#)
- SerialReceiveQueueMaxCharacterCount, [551](#)
- SerialTransmitQueueCurrentCharacterCount, [551](#)
- SerialTransmitQueueMaxCharacterCount, [551](#)
- Sharpening, [551](#)
- SharpeningAuto, [551](#)
- SharpeningEnable, [552](#)
- SharpeningThreshold, [552](#)
- SoftwareSignalPulse, [552](#)
- SoftwareSignalSelector, [552](#)
- SourceCount, [553](#)
- SourceSelector, [553](#)
- Test0001, [553](#)
- TestEventGenerate, [553](#)
- TestPattern, [553](#)
- TestPatternGeneratorSelector, [553](#)
- TestPendingAck, [554](#)
- TimerDelay, [554](#)
- TimerDuration, [554](#)
- TimerReset, [554](#)
- TimerSelector, [554](#)
- TimerStatus, [555](#)
- TimerTriggerActivation, [555](#)
- TimerTriggerSource, [555](#)
- TimerValue, [555](#)
- Timestamp, [555](#)
- TimestampLatch, [555](#)
- TimestampLatchValue, [556](#)
- TimestampReset, [556](#)
- TLParamsLocked, [556](#)
- TransferAbort, [556](#)
- TransferBlockCount, [556](#)
- TransferBurstCount, [556](#)
- TransferComponentSelector, [557](#)
- TransferControlMode, [557](#)
- TransferOperationMode, [557](#)
- TransferPause, [557](#)
- TransferQueueCurrentBlockCount, [557](#)
- TransferQueueMaxBlockCount, [557](#)
- TransferQueueMode, [558](#)
- TransferQueueOverflowCount, [558](#)
- TransferResume, [558](#)
- TransferSelector, [558](#)
- TransferStart, [558](#)
- TransferStatus, [558](#)
- TransferStatusSelector, [559](#)
- TransferStop, [559](#)
- TransferStreamChannel, [559](#)
- TransferTriggerActivation, [559](#)
- TransferTriggerMode, [559](#)
- TransferTriggerSelector, [559](#)
- TransferTriggerSource, [560](#)
- TriggerActivation, [560](#)
- TriggerDelay, [560](#)
- TriggerDivider, [560](#)
- TriggerEventTest, [560](#)
- TriggerMode, [561](#)
- TriggerMultiplier, [561](#)
- TriggerOverlap, [561](#)
- TriggerSelector, [561](#)
- TriggerSoftware, [561](#)
- TriggerSource, [562](#)
- UserOutputSelector, [562](#)
- UserOutputValue, [562](#)
- UserOutputValueAll, [562](#)
- UserOutputValueAllMask, [562](#)
- UserSetDefault, [563](#)
- UserSetFeatureEnable, [563](#)
- UserSetLoad, [563](#)
- UserSetSave, [563](#)
- UserSetSelector, [564](#)
- V3_3Enable, [564](#)
- WhiteClip, [564](#)
- WhiteClipSelector, [564](#)
- Width, [564](#)
- WidthMax, [565](#)
- Camera Base Class, [45](#)
- Camera Base Interface Class, [92](#)
- Camera Class, [44](#)
- Camera List Class, [47](#)
- CameraBase, [565](#)
 - ~CameraBase, [568](#)
 - BeginAcquisition, [568](#)
 - CameraBase, [568](#)
 - DeInit, [569](#)
 - DiscoverMaxPacketSize, [569](#)
 - EndAcquisition, [569](#)
 - ForceIP, [570](#)
 - GetAccessMode, [570](#)
 - GetBufferOwnership, [570](#)
 - GetGuiXml, [571](#)
 - GetNextImage, [571](#)
 - GetNextImageSync, [572](#)
 - GetNodeMap, [572](#)
 - GetNumDataStreams, [573](#)
 - GetNumImagesInUse, [573](#)
 - GetTLDeviceNodeMap, [573](#)
 - GetTLStreamNodeMap, [573](#)
 - GetUniqueID, [574](#)
 - GetUserBufferCount, [574](#)
 - GetUserBufferSize, [574](#)
 - GetUserBufferTotalSize, [575](#)
 - Init, [575](#)
 - InterfaceImpl, [581](#)

- IsInitialized, [576](#)
- IsStreaming, [576](#)
- IsValid, [576](#)
- operator=, [577](#)
- ReadPort, [577](#)
- RegisterEventHandler, [577](#), [578](#)
- SetBufferOwnership, [579](#)
- SetUserBuffers, [579](#), [580](#)
- TransportLayerDevice, [1122](#)
- TransportLayerStream, [1144](#)
- UnregisterEventHandler, [580](#)
- WritePort, [581](#)
- CameraCloseFile
 - Inference.cpp, [1452](#)
- CameraDefs Class, [46](#)
- CameraDeleteFile
 - Inference.cpp, [1452](#)
- cameraGrabInfoMap
 - AcquisitionMultipleCameraRecovery.cpp, [1409](#)
- CameraInternal
 - ICameraBase, [806](#)
 - TransportLayerDevice, [1122](#)
 - TransportLayerStream, [1144](#)
- CameraList, [581](#)
 - ~CameraList, [583](#)
 - Add, [583](#)
 - Append, [584](#)
 - CameraList, [583](#)
 - Clear, [584](#)
 - GetByDeviceID, [584](#)
 - GetByIndex, [585](#)
 - GetBySerial, [585](#)
 - GetSize, [586](#)
 - operator=, [586](#)
 - operator[], [586](#)
 - Remove, [586](#)
 - RemoveByDeviceID, [587](#)
 - RemoveByIndex, [587](#)
 - RemoveBySerial, [587](#)
- CameraListImpl
 - BasePtr< T, B >, [410](#)
 - ICameraList, [811](#)
- CameraOpenFile
 - Inference.cpp, [1452](#)
- CameraPtr, [588](#)
 - CameraPtr, [589](#)
- CameraPtr Class, [48](#)
- CameraWriteToFile
 - Inference.cpp, [1452](#)
- CastToDestroy
 - Spinnaker::GenApi, [365](#)
- CategoryNode, [590](#)
 - ~CategoryNode, [592](#)
 - CategoryNode, [591](#)
 - GetFeatures, [592](#)
 - SetReference, [592](#)
- CategoryNode Class, [105](#)
- causeSpinnakerException
 - ExceptionHandling.cpp, [1431](#)
- causeStandardException
 - ExceptionHandling.cpp, [1431](#)
- CBasePtr
 - Pointer Class, [166](#)
- CBooleanPtr
 - Pointer Class, [166](#)
- CBooleanRef
 - Spinnaker::GenApi, [353](#)
- cbPostInsideLock
 - Spinnaker::GenApi, [357](#)
- cbPostOutsideLock
 - Spinnaker::GenApi, [357](#)
- CCategoryPtr
 - Pointer Class, [167](#)
- CCategoryRef
 - Spinnaker::GenApi, [353](#)
- CChunkAdapter, [592](#)
 - ~CChunkAdapter, [593](#)
 - AttachBuffer, [594](#)
 - AttachNodeMap, [594](#)
 - CChunkAdapter, [593](#)
 - CheckBufferLayout, [594](#)
 - ClearCaches, [594](#)
 - DetachBuffer, [595](#)
 - DetachNodeMap, [595](#)
 - m_pChunkAdapter, [595](#)
 - UpdateBuffer, [595](#)
- CChunkAdapterDcam, [596](#)
 - ~CChunkAdapterDcam, [597](#)
 - AttachBuffer, [597](#)
 - CChunkAdapterDcam, [597](#)
 - CheckBufferLayout, [597](#)
 - CheckCRC, [598](#)
 - HasCRC, [598](#)
- CChunkAdapterGeneric, [598](#)
 - ~CChunkAdapterGeneric, [599](#)
 - AttachBuffer, [600](#)
 - CChunkAdapterGeneric, [599](#)
 - CheckBufferLayout, [600](#)
- CChunkAdapterGEV, [601](#)
 - ~CChunkAdapterGEV, [602](#)
 - AttachBuffer, [602](#)
 - CChunkAdapterGEV, [602](#)
 - CheckBufferLayout, [602](#)
- CChunkAdapterU3V, [603](#)
 - ~CChunkAdapterU3V, [604](#)
 - AttachBuffer, [604](#)
 - CChunkAdapterU3V, [604](#)
 - CheckBufferLayout, [604](#)
- CChunkPort, [605](#)
 - ~CChunkPort, [606](#)
 - AttachChunk, [607](#)
 - AttachPort, [607](#)
 - CChunkPort, [606](#)
 - CheckChunkID, [607](#)
 - ClearCache, [607](#)
 - DetachChunk, [608](#)

- DetachPort, 608
- GetAccessMode, 608
- GetChunkIDLength, 608
- GetPrincipalInterfaceType, 608
- GetSwapEndianness, 608
- InvalidateNode, 609
- m_pChunkPort, 610
- m_pPort, 610
- m_pPortAdapter, 610
- Read, 609
- SetPortImpl, 609
- UpdateBuffer, 609
- Write, 609
- CChunkPortPtr
 - Pointer Class, 167
- CCMAApplication
 - Spinnaker, 240
- CCMColorSpace
 - Spinnaker, 241
- CCMColorTemperature
 - Spinnaker, 241
- CCMSensor
 - Spinnaker, 241
- CCMSettings, 610
 - Application, 611
 - CCMSettings, 610
 - ColorSpace, 611
 - ColorTemperature, 611
 - CustomCCMCode, 611
 - Sensor, 611
 - Type, 611
- CCMType
 - Spinnaker, 241
- CCommandPtr
 - Pointer Class, 167
- CCommandRef
 - Spinnaker::GenApi, 354
- CDeviceInfoPtr
 - Pointer Class, 167
- centerXCoord
 - InferenceBoxCircle, 972
- centerYCoord
 - InferenceBoxCircle, 972
- CEnumEntryPtr
 - Pointer Class, 167
- CEnumEntryRef
 - Spinnaker::GenApi, 354
- CEnumerationPtr
 - Pointer Class, 167
- CEnumerationRef
 - Spinnaker::GenApi, 354
- CEnumerationTRef
 - CEnumerationTRef< EnumT >, 614
- CEnumerationTRef< EnumT >, 612
 - ~CEnumerationTRef, 614
 - CEnumerationTRef, 614
 - GetCurrentEntry, 614
 - GetEntry, 615
 - GetValue, 615
 - operator(), 615
 - operator=, 616
 - SetEnumReference, 616
 - SetNumEnums, 616
 - SetReference, 616
 - SetValue, 617
- CEventAdapter, 617
 - ~CEventAdapter, 618
 - AttachNodeMap, 618
 - CEventAdapter, 618
 - DeliverMessage, 618
 - DetachNodeMap, 619
 - m_pEventAdapter, 619
- CEventAdapter1394, 619
 - ~CEventAdapter1394, 620
 - CEventAdapter1394, 620
 - DeliverEventMessage, 621
 - DeliverMessage, 621
- CEventAdapterGeneric, 621
 - ~CEventAdapterGeneric, 622
 - CEventAdapterGeneric, 622
 - DeliverMessage, 623
- CEventAdapterGEV, 624
 - ~CEventAdapterGEV, 625
 - CEventAdapterGEV, 625
 - DeliverEventMessage, 625
 - DeliverMessage, 625
- CEventAdapterU3V, 626
 - ~CEventAdapterU3V, 627
 - CEventAdapterU3V, 627
 - DeliverEventMessage, 627
 - DeliverMessage, 627
- CEventPort, 628
 - ~CEventPort, 630
 - AttachEvent, 630
 - AttachNode, 630
 - CEventPort, 629
 - CheckEventID, 630
 - DetachEvent, 631
 - DetachNode, 631
 - GetAccessMode, 631
 - GetEventIDLength, 631
 - GetPrincipalInterfaceType, 631
 - GetSwapEndianness, 631
 - InvalidateNode, 632
 - m_pEventPort, 632
 - m_pNode, 633
 - m_pPortAdapter, 633
 - Read, 632
 - SetPortImpl, 632
 - Write, 632
- CFeatureBag, 633
 - ~CFeatureBag, 634
 - CFeatureBag, 634
 - GetFeatureBagHandle, 634
 - LoadFromBag, 634
 - operator==, 635

- PersistFeature, [635](#)
- SetInfo, [635](#)
- StoreToBag, [635](#)
- CFloatPtr, [636](#)
 - CFloatPtr, [637](#)
 - GetEnumAlias, [637](#)
 - GetIntAlias, [637](#)
 - operator=, [637](#)
- CFloatRef
 - Spinnaker::GenApi, [354](#)
- CGeneric_XMLLoaderParams, [638](#)
 - _Initialize, [638](#)
- CGlobalLock, [639](#)
 - ~CGlobalLock, [640](#)
 - CGlobalLock, [639](#), [640](#)
 - IsValid, [640](#)
 - Lock, [640](#)
 - m_DebugCount, [641](#)
 - TryLock, [640](#)
 - Unlock, [641](#)
- CGlobalLockUnlocker, [641](#)
 - ~CGlobalLockUnlocker, [642](#)
 - CGlobalLockUnlocker, [642](#)
 - m_enabled, [643](#)
 - m_Lock, [643](#)
 - UnlockEarly, [642](#)
- ChangeHeightAndGain
 - NodeMapCallback.cpp, [1462](#)
- CheckBufferLayout
 - CChunkAdapter, [594](#)
 - CChunkAdapterDcam, [597](#)
 - CChunkAdapterGeneric, [600](#)
 - CChunkAdapterGEV, [602](#)
 - CChunkAdapterU3V, [604](#)
- CheckChunkID
 - CChunkPort, [607](#)
- CheckCRC
 - CChunkAdapterDcam, [598](#)
 - Image, [841](#)
 - Image, [890](#)
- CheckEventID
 - CEventPort, [630](#)
- CheckGevEnabled
 - EnumerationEvents.cpp, [1430](#)
- chosenChunkData
 - ChunkData.cpp, [1421](#)
- chosenEvent
 - DeviceEvents.cpp, [1428](#)
- chosenException
 - ExceptionHandling.cpp, [1432](#)
- chosenFileUploadPersistence
 - Inference.cpp, [1456](#)
- chosenInferenceNetworkType
 - Inference.cpp, [1456](#)
- chosenRead
 - NodeMapInfo.cpp, [1466](#)
- chosenTrigger
 - Trigger.cpp, [1480](#)
 - Trigger_QuickSpin.cpp, [1482](#)
- chosenVideoType
 - SaveToAvi.cpp, [1471](#)
- Chunk Data Inference Class, [50](#)
- CHUNK_BASE_ADDRESS_REGISTER
 - IChunkPort.h, [1278](#)
- CHUNK_BASE_ADDRESS_REGISTER_LEN
 - IChunkPort.h, [1278](#)
- CHUNK_LENGTH_REGISTER
 - IChunkPort.h, [1278](#)
- CHUNK_LENGTH_REGISTER_LEN
 - IChunkPort.h, [1278](#)
- ChunkAdapter Class, [106](#)
- ChunkAdapterDcam Class, [107](#)
- ChunkAdapterGeneric Class, [108](#)
- ChunkAdapterGEV Class, [109](#)
- ChunkAdapterU3V Class, [188](#)
- ChunkBlackLevel
 - Camera, [455](#)
- ChunkBlackLevelSelector
 - Camera, [455](#)
- ChunkBlackLevelSelector_All
 - Spinnaker, [242](#)
- ChunkBlackLevelSelectorEnums
 - Spinnaker, [242](#)
- ChunkCompressionMode
 - Camera, [456](#)
- ChunkCompressionRatio
 - Camera, [456](#)
- ChunkCounterSelector
 - Camera, [456](#)
- ChunkCounterSelector_Counter0
 - Spinnaker, [242](#)
- ChunkCounterSelector_Counter1
 - Spinnaker, [242](#)
- ChunkCounterSelector_Counter2
 - Spinnaker, [242](#)
- ChunkCounterSelectorEnums
 - Spinnaker, [242](#)
- ChunkCounterValue
 - Camera, [456](#)
- ChunkCRC
 - Camera, [456](#)
- ChunkData, [643](#)
 - ~ChunkData, [646](#)
 - ChunkData, [646](#)
 - ChunkDataImpl, [656](#)
 - GetBlackLevel, [646](#)
 - GetCompressionMode, [646](#)
 - GetCompressionRatio, [646](#)
 - GetCounterValue, [647](#)
 - GetCRC, [647](#)
 - GetEncoderValue, [647](#)
 - GetExposureEndLineStatusAll, [647](#)
 - GetExposureTime, [648](#)
 - GetFrameID, [648](#)
 - GetGain, [648](#)
 - GetHeight, [648](#)

- GetImage, [649](#)
- GetInferenceBoundingBoxResult, [649](#)
- GetInferenceConfidence, [649](#)
- GetInferenceFrameId, [649](#)
- GetInferenceResult, [650](#)
- GetLinePitch, [650](#)
- GetLineStatusAll, [650](#)
- GetOffsetX, [650](#)
- GetOffsetY, [651](#)
- GetPartSelector, [651](#)
- GetPixelDynamicRangeMax, [651](#)
- GetPixelDynamicRangeMin, [651](#)
- GetScan3dAxisMax, [652](#)
- GetScan3dAxisMin, [652](#)
- GetScan3dCoordinateOffset, [652](#)
- GetScan3dCoordinateReferenceValue, [652](#)
- GetScan3dCoordinateScale, [653](#)
- GetScan3dInvalidDataValue, [653](#)
- GetScan3dTransformValue, [653](#)
- GetScanLineSelector, [653](#)
- GetSequencerSetActive, [654](#)
- GetSerialDataLength, [654](#)
- GetStreamChannelId, [654](#)
- GetTimerValue, [654](#)
- GetTimestamp, [655](#)
- GetTimestampLatchValue, [655](#)
- GetTransferBlockId, [655](#)
- GetTransferQueueCurrentBlockCount, [655](#)
- GetWidth, [656](#)
- SetChunks, [656](#)
- ChunkData Class, [49](#)
- ChunkData.cpp
 - AcquireImages, [1419](#)
 - chosenChunkData, [1421](#)
 - chunkDataType, [1419](#)
 - ConfigureChunkData, [1419](#)
 - DisableChunkData, [1420](#)
 - DisplayChunkData, [1420](#)
 - IMAGE, [1419](#)
 - main, [1420](#)
 - NODEMAP, [1419](#)
 - PrintDeviceInfo, [1420](#)
 - RunSingleCamera, [1420](#)
- ChunkDataImpl
 - ChunkData, [656](#)
 - IChunkData, [820](#)
 - InferenceBoundingBoxResult, [972](#)
- chunkDataType
 - ChunkData.cpp, [1419](#)
- ChunkEnable
 - Camera, [456](#)
- ChunkEncoderSelector
 - Camera, [457](#)
- ChunkEncoderSelector_Encoder0
 - Spinnaker, [242](#)
- ChunkEncoderSelector_Encoder1
 - Spinnaker, [243](#)
- ChunkEncoderSelector_Encoder2
 - Spinnaker, [243](#)
- ChunkEncoderSelectorEnums
 - Spinnaker, [242](#)
- ChunkEncoderStatus
 - Camera, [457](#)
- ChunkEncoderStatus_EncoderDown
 - Spinnaker, [243](#)
- ChunkEncoderStatus_EncoderIdle
 - Spinnaker, [243](#)
- ChunkEncoderStatus_EncoderStatic
 - Spinnaker, [243](#)
- ChunkEncoderStatus_EncoderUp
 - Spinnaker, [243](#)
- ChunkEncoderStatusEnums
 - Spinnaker, [243](#)
- ChunkEncoderValue
 - Camera, [457](#)
- ChunkExposureEndLineStatusAll
 - Camera, [457](#)
- ChunkExposureTime
 - Camera, [457](#)
- ChunkExposureTimeSelector
 - Camera, [457](#)
- ChunkExposureTimeSelector_Blue
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Common
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Cyan
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Green
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Infrared
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Magenta
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Red
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Stage1
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Stage2
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Ultraviolet
 - Spinnaker, [243](#)
- ChunkExposureTimeSelector_Yellow
 - Spinnaker, [243](#)
- ChunkExposureTimeSelectorEnums
 - Spinnaker, [243](#)
- ChunkFrameId
 - Camera, [458](#)
- ChunkGain
 - Camera, [458](#)
- ChunkGainSelector
 - Camera, [458](#)
- ChunkGainSelector_All
 - Spinnaker, [244](#)
- ChunkGainSelector_Blue
 - Spinnaker, [244](#)
- ChunkGainSelector_Green
 - Spinnaker, [244](#)

- Spinnaker, [244](#)
- ChunkGainSelector_Red
 - Spinnaker, [244](#)
- ChunkGainSelectorEnums
 - Spinnaker, [243](#)
- ChunkHeight
 - Camera, [458](#)
- ChunkID
 - DCAM_CHUNK_TRAILER, [712](#)
 - GVCP_CHUNK_TRAILER, [786](#)
 - SingleChunkData_t, [1085](#)
 - SingleChunkDataStr_t, [1086](#)
 - U3V_CHUNK_TRAILER, [1157](#)
- ChunkImage
 - Camera, [458](#)
- ChunkImageComponent
 - Camera, [458](#)
- ChunkImageComponent_Color
 - Spinnaker, [244](#)
- ChunkImageComponent_Confidence
 - Spinnaker, [244](#)
- ChunkImageComponent_Disparity
 - Spinnaker, [244](#)
- ChunkImageComponent_Infrared
 - Spinnaker, [244](#)
- ChunkImageComponent_Intensity
 - Spinnaker, [244](#)
- ChunkImageComponent_Range
 - Spinnaker, [244](#)
- ChunkImageComponent_Scatter
 - Spinnaker, [244](#)
- ChunkImageComponent_Ultraviolet
 - Spinnaker, [244](#)
- ChunkImageComponentEnums
 - Spinnaker, [244](#)
- ChunkInferenceBoundingBoxResult
 - Camera, [459](#)
- ChunkInferenceConfidence
 - Camera, [459](#)
- ChunkInferenceFrameld
 - Camera, [459](#)
- ChunkInferenceResult
 - Camera, [459](#)
- ChunkLength
 - DCAM_CHUNK_TRAILER, [712](#)
 - GVCP_CHUNK_TRAILER, [786](#)
 - SingleChunkData_t, [1085](#)
 - SingleChunkDataStr_t, [1086](#)
 - U3V_CHUNK_TRAILER, [1157](#)
- ChunkLinePitch
 - Camera, [459](#)
- ChunkLineStatusAll
 - Camera, [459](#)
- ChunkModeActive
 - Camera, [460](#)
- ChunkOffset
 - SingleChunkData_t, [1085](#)
 - SingleChunkDataStr_t, [1086](#)
- ChunkOffsetX
 - Camera, [460](#)
- ChunkOffsetY
 - Camera, [460](#)
- ChunkPartSelector
 - Camera, [460](#)
- ChunkPixelDynamicRangeMax
 - Camera, [460](#)
- ChunkPixelDynamicRangeMin
 - Camera, [460](#)
- ChunkPixelFormat
 - Camera, [461](#)
- ChunkPixelFormat_BayerBG8
 - Spinnaker, [245](#)
- ChunkPixelFormat_BayerGB8
 - Spinnaker, [245](#)
- ChunkPixelFormat_BayerGR8
 - Spinnaker, [245](#)
- ChunkPixelFormat_BayerRG8
 - Spinnaker, [245](#)
- ChunkPixelFormat_Mono12Packed
 - Spinnaker, [244](#)
- ChunkPixelFormat_Mono16
 - Spinnaker, [244](#)
- ChunkPixelFormat_Mono8
 - Spinnaker, [244](#)
- ChunkPixelFormat_RGB8Packed
 - Spinnaker, [244](#)
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY
 - Spinnaker, [245](#)
- ChunkPixelFormat_YUV422Packed
 - Spinnaker, [245](#)
- ChunkPixelFormatEnums
 - Spinnaker, [244](#)
- ChunkPort Class, [110](#)
- ChunkRegionID
 - Camera, [461](#)
- ChunkRegionID_Region0
 - Spinnaker, [245](#)
- ChunkRegionID_Region1
 - Spinnaker, [245](#)
- ChunkRegionID_Region2
 - Spinnaker, [245](#)
- ChunkRegionIDEnums
 - Spinnaker, [245](#)
- ChunkScan3dAxisMax
 - Camera, [461](#)
- ChunkScan3dAxisMin
 - Camera, [461](#)
- ChunkScan3dCoordinateOffset
 - Camera, [461](#)
- ChunkScan3dCoordinateReferenceSelector
 - Camera, [461](#)
- ChunkScan3dCoordinateReferenceSelector_RotationX
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelector_RotationY
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelector_RotationZ

- Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationX
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationY
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationZ
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceSelectorEnums
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateReferenceValue
 - Camera, [462](#)
- ChunkScan3dCoordinateScale
 - Camera, [462](#)
- ChunkScan3dCoordinateSelector
 - Camera, [462](#)
- ChunkScan3dCoordinateSelector_CoordinateA
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSelector_CoordinateB
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSelector_CoordinateC
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSelectorEnums
 - Spinnaker, [245](#)
- ChunkScan3dCoordinateSystem
 - Camera, [462](#)
- ChunkScan3dCoordinateSystem_Cartesian
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystem_Cylindrical
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystem_Spherical
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystemEnums
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystemReference
 - Camera, [462](#)
- ChunkScan3dCoordinateSystemReference_Anchor
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystemReference_Transformed
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateSystemReferenceEnums
 - Spinnaker, [246](#)
- ChunkScan3dCoordinateTransformSelector
 - Camera, [462](#)
- ChunkScan3dCoordinateTransformSelector_RotationX
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelector_RotationY
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelector_RotationZ
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationX
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationY
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationZ
 - Spinnaker, [247](#)
- ChunkScan3dCoordinateTransformSelectorEnums
 - Spinnaker, [246](#)
- ChunkScan3dDistanceUnit
 - Camera, [463](#)
- ChunkScan3dDistanceUnit_Inch
 - Spinnaker, [247](#)
- ChunkScan3dDistanceUnit_Millimeter
 - Spinnaker, [247](#)
- ChunkScan3dDistanceUnitEnums
 - Spinnaker, [247](#)
- ChunkScan3dInvalidDataFlag
 - Camera, [463](#)
- ChunkScan3dInvalidDataValue
 - Camera, [463](#)
- ChunkScan3dOutputMode
 - Camera, [463](#)
- ChunkScan3dOutputMode_CalibratedABC_Grid
 - Spinnaker, [247](#)
- ChunkScan3dOutputMode_CalibratedABC_PointCloud
 - Spinnaker, [247](#)
- ChunkScan3dOutputMode_CalibratedAC
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_CalibratedAC_Linescan
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_CalibratedC
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_CalibratedC_Linescan
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_DisparityC
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_DisparityC_Linescan
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_RectifiedC
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_RectifiedC_Linescan
 - Spinnaker, [248](#)
- ChunkScan3dOutputMode_UncalibratedC
 - Spinnaker, [247](#)
- ChunkScan3dOutputModeEnums
 - Spinnaker, [247](#)
- ChunkScan3dTransformValue
 - Camera, [463](#)
- ChunkScanLineSelector
 - Camera, [463](#)
- ChunkSelector
 - Camera, [464](#)
- ChunkSelector_BlackLevel
 - Spinnaker, [249](#)
- ChunkSelector_CRC
 - Spinnaker, [248](#)
- ChunkSelector_ExposureEndLineStatusAll
 - Spinnaker, [249](#)
- ChunkSelector_ExposureTime
 - Spinnaker, [248](#)
- ChunkSelector_FrameID
 - Spinnaker, [248](#)
- ChunkSelector_Gain
 - Spinnaker, [249](#)
- ChunkSelector_Height
 - Spinnaker, [248](#)
- ChunkSelector_Image

- Spinnaker, [248](#)
- ChunkSelector_OffsetX
 - Spinnaker, [248](#)
- ChunkSelector_OffsetY
 - Spinnaker, [248](#)
- ChunkSelector_PixelFormat
 - Spinnaker, [249](#)
- ChunkSelector_SequencerSetActive
 - Spinnaker, [249](#)
- ChunkSelector_SerialData
 - Spinnaker, [249](#)
- ChunkSelector_Timestamp
 - Spinnaker, [249](#)
- ChunkSelector_Width
 - Spinnaker, [248](#)
- ChunkSelectorEnums
 - Spinnaker, [248](#)
- ChunkSequencerSetActive
 - Camera, [464](#)
- ChunkSerialData
 - Camera, [464](#)
- ChunkSerialDataLength
 - Camera, [464](#)
- ChunkSerialReceiveOverflow
 - Camera, [464](#)
- ChunkSourceID
 - Camera, [464](#)
- ChunkSourceID_Source0
 - Spinnaker, [249](#)
- ChunkSourceID_Source1
 - Spinnaker, [249](#)
- ChunkSourceID_Source2
 - Spinnaker, [249](#)
- ChunkSourceIDEnums
 - Spinnaker, [249](#)
- ChunkStreamChannelID
 - Camera, [465](#)
- ChunkTimerSelector
 - Camera, [465](#)
- ChunkTimerSelector_Timer0
 - Spinnaker, [249](#)
- ChunkTimerSelector_Timer1
 - Spinnaker, [249](#)
- ChunkTimerSelector_Timer2
 - Spinnaker, [249](#)
- ChunkTimerSelectorEnums
 - Spinnaker, [249](#)
- ChunkTimerValue
 - Camera, [465](#)
- ChunkTimestamp
 - Camera, [465](#)
- ChunkTimestampLatchValue
 - Camera, [465](#)
- ChunkTransferBlockID
 - Camera, [465](#)
- ChunkTransferQueueCurrentBlockCount
 - Camera, [466](#)
- ChunkTransferStreamID
 - Camera, [466](#)
- ChunkTransferStreamID_Stream0
 - Spinnaker, [250](#)
- ChunkTransferStreamID_Stream1
 - Spinnaker, [250](#)
- ChunkTransferStreamID_Stream2
 - Spinnaker, [250](#)
- ChunkTransferStreamID_Stream3
 - Spinnaker, [250](#)
- ChunkTransferStreamIDEnums
 - Spinnaker, [249](#)
- ChunkWidth
 - Camera, [466](#)
- CIntegerPtr
 - Pointer Class, [168](#)
- CIntegerRef
 - Spinnaker::GenApi, [354](#)
- circle
 - InferenceBoundingBox, [968](#)
- CL
 - Spinnaker::GenApi, [361](#)
- classId
 - InferenceBoundingBox, [968](#)
- CLASSIFICATION
 - Inference.cpp, [1452](#)
- CIConfiguration
 - Camera, [466](#)
- CIConfiguration_Base
 - Spinnaker, [250](#)
- CIConfiguration_DualBase
 - Spinnaker, [250](#)
- CIConfiguration_EightyBit
 - Spinnaker, [250](#)
- CIConfiguration_Full
 - Spinnaker, [250](#)
- CIConfiguration_Medium
 - Spinnaker, [250](#)
- CIConfigurationEnums
 - Spinnaker, [250](#)
- CleanUp
 - SerialRxTx.cpp, [1476](#)
- CleanupChunkAdapter
 - IDataStream, [823](#)
- Clear
 - CameraList, [584](#)
 - ICameraList, [809](#)
 - ImageList, [856](#)
 - IInterfaceList, [879](#)
 - ImageList, [922](#)
 - InterfaceList, [1001](#)
- ClearCache
 - CChunkPort, [607](#)
 - CNodeMapFactory, [670](#)
- ClearCaches
 - CChunkAdapter, [594](#)
- ClearXMLCache
 - NodeMap, [1052](#)
- CLock, [657](#), [658](#)

- ~CLOCK, 657, 659
- CLOCK, 657, 659
- Lock, 658, 660
- m_bOwnLock, 660
- m_lock, 661
- NodeMap, 660
- TryLock, 658, 660
- Unlock, 658, 660
- CLOCKEx, 661, 662
 - m_lockEx, 663
- Close
 - SpinVideo, 1090
- close
 - IDevFileStreamBase< CharType, Traits >, 829
 - IDevFileStreamBuf< CharType, Traits >, 832
 - ODevFileStreamBase< CharType, Traits >, 1060
 - ODevFileStreamBuf< CharType, Traits >, 1062
- CloseFile
 - FileAccess_QuickSpin.cpp, 1435
- closeFile
 - FileProtocolAdapter, 762
- CITimeSlotsCount
 - Camera, 466
- CITimeSlotsCount_One
 - Spinnaker, 250
- CITimeSlotsCount_Three
 - Spinnaker, 250
- CITimeSlotsCount_Two
 - Spinnaker, 250
- CITimeSlotsCountEnums
 - Spinnaker, 250
- CNodeCallback, 663
 - ~CNodeCallback, 664
 - CNodeCallback, 664
 - Destroy, 664
 - GetCallbackType, 665
 - GetNode, 665
 - m_CallbackType, 665
 - m_pNode, 665
 - operator(), 665
- CNodeMapDynPtr
 - Pointer Class, 168
- CNodeMapFactory, 666
 - ~CNodeMapFactory, 668
 - AddInjectionData, 670
 - ApplyStyleSheet, 670
 - ClearCache, 670
 - CNodeMapFactory, 668, 669
 - CreateEmptyNodeMap, 671
 - CreateNodeDataFromNodeMap, 671
 - CreateNodeMap, 671
 - ExtractSubtree, 671
 - GetNodeStatistics, 672
 - GetSupportedSchemaVersions, 672
 - IsCameraDescriptionFileDataReleased, 672
 - IsEmpty, 672
 - IsLoaded, 672
 - IsPreprocessed, 673
 - LoadAndInject, 673
 - operator=, 673
 - Preprocess, 673
 - ReleaseCameraDescriptionFileData, 673
 - ToString, 674
 - ToXml, 674
- CNodeMapFactory::NodeStatistics_t, 1058
 - NumLinks, 1058
 - NumNodes, 1058
 - NumProperties, 1058
 - NumStrings, 1058
- CNodeMapPtr
 - Pointer Class, 168
- CNodeMapRef, 674
 - CNodeMapRef, 675, 676
 - operator=, 676
 - Spinnaker::GenApi, 354
- CNodeMapRefT
 - CNodeMapRefT< TCameraParams >, 679
- CNodeMapRefT< TCameraParams >, 677
 - _ClearXMLCache, 679
 - _Connect, 680
 - _Destroy, 680
 - _GetDeviceName, 680
 - _GetNode, 680
 - _GetNodes, 680
 - _GetSupportedSchemaVersions, 681
 - _InvalidateNodes, 681
 - _LoadXMLFromFile, 681
 - _LoadXMLFromFileInject, 681
 - _LoadXMLFromString, 681
 - _LoadXMLFromStringInject, 682
 - _LoadXMLFromZIPData, 682
 - _LoadXMLFromZIPFile, 682
 - _Poll, 682
 - _Ptr, 683
 - ~CNodeMapRefT, 679
 - CNodeMapRefT, 679
 - operator=, 682, 683
- CNodePtr
 - Pointer Class, 168
- CNodeRef
 - Spinnaker::GenApi, 354
- ColorProcessingAlgorithm
 - Spinnaker, 250
- ColorSpace
 - CCMSettings, 611
- ColorTemperature
 - CCMSettings, 611
- ColorTransformationEnable
 - Camera, 466
- ColorTransformationSelector
 - Camera, 467
- ColorTransformationSelector_RGBtoRGB
 - Spinnaker, 251
- ColorTransformationSelector_RGBtoYUV
 - Spinnaker, 251
- ColorTransformationSelectorEnums

- Spinnaker, [251](#)
- ColorTransformationValue
 - Camera, [467](#)
- ColorTransformationValueSelector
 - Camera, [467](#)
- ColorTransformationValueSelector_Gain00
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain01
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain02
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain10
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain11
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain12
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain20
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain21
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Gain22
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Offset0
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Offset1
 - Spinnaker, [252](#)
- ColorTransformationValueSelector_Offset2
 - Spinnaker, [252](#)
- ColorTransformationValueSelectorEnums
 - Spinnaker, [251](#)
- COM_PORT_COUNT_MAX
 - SerialRxTx.cpp, [1474](#)
- Combine
 - INode Interface, [143](#), [144](#)
- Command
 - GVCP_REQUEST_HEADER, [795](#)
- COMMAND_MAGIC
 - Spinnaker::GenApi, [388](#)
- CommandExecute
 - SpinnakerDirectShow.h, [1338](#)
- CommandHeader
 - U3V_EVENT_MESSAGE, [1160](#)
- CommandId
 - U3V_COMMAND_HEADER, [1158](#)
- CommandNode, [683](#)
 - ~CommandNode, [685](#)
 - CommandNode, [684](#), [685](#)
 - Execute, [685](#)
 - IsDone, [685](#)
 - operator(), [686](#)
 - SetReference, [686](#)
- CommandNode Class, [111](#)
- compare
 - gcstring, [779](#)
- Compatibility.h
 - FMT_I64, [1251](#)
- CompressedImageInfo, [686](#)
 - CompressedImageInfo, [686](#)
 - compressedImageSize, [687](#)
 - fileName, [687](#)
 - imageHeight, [687](#)
 - imagePixelFormat, [687](#)
 - imageWidth, [687](#)
 - imageXOffset, [687](#)
 - imageYOffset, [688](#)
- compressedImageSize
 - CompressedImageInfo, [687](#)
- compression
 - TIFFOption, [1118](#)
- Compression.cpp
 - AcquireImages, [1421](#)
 - ConfigureGVCPHeartbeat, [1422](#)
 - DisableGVCPHeartbeat, [1422](#)
 - DisableImageChunkData, [1422](#)
 - DisableImageCompression, [1422](#)
 - enableChunkData, [1423](#)
 - EnableImageChunkData, [1422](#)
 - EnableImageCompression, [1422](#)
 - main, [1423](#)
 - PrintDeviceInfo, [1423](#)
 - ProcessCompressedImagesFromFile, [1423](#)
 - ResetGVCPHeartbeat, [1423](#)
 - RunSingleCamera, [1423](#)
- compressionLevel
 - PNGOption, [1065](#)
- CompressionRatio
 - Camera, [467](#)
- CompressionSaturationPriority
 - Camera, [468](#)
- CompressionSaturationPriority_DropFrame
 - Spinnaker, [252](#)
- CompressionSaturationPriority_ReduceFrameRate
 - Spinnaker, [252](#)
- CompressionSaturationPriorityEnums
 - Spinnaker, [252](#)
- confidence
 - DeviceEventInferenceData, [721](#)
 - InferenceBoundingBox, [969](#)
- ConfigureActionControl
 - ActionCommand.cpp, [1414](#)
- ConfigureCallbacks
 - NodeMapCallback.cpp, [1462](#)
- ConfigureCamera
 - AcquisitionMultipleCameraRecovery.cpp, [1408](#)
- ConfigureCameras
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1410](#)
- ConfigureChunkData
 - ActionCommand.cpp, [1414](#)
 - ChunkData.cpp, [1419](#)
 - Inference.cpp, [1453](#)
- ConfigureCustomImageSettings
 - ImageFormatControl.cpp, [1448](#)
 - ImageFormatControl_QuickSpin.cpp, [1449](#)
- ConfigureDevice
 - SerialRxTx.cpp, [1476](#)

- ConfigureDeviceEvents
 - DeviceEvents.cpp, [1427](#)
- ConfigureDigitalIO
 - CounterAndTimer.cpp, [1424](#)
- ConfigureExposure
 - Exposure.cpp, [1432](#)
 - Exposure_QuickSpin.cpp, [1434](#)
- ConfigureExposureandTrigger
 - CounterAndTimer.cpp, [1424](#)
- ConfigureGVCPHeartbeat
 - Acquisition.cpp, [1365](#)
 - AcquisitionMultipleThread.cpp, [1412](#)
 - Compression.cpp, [1422](#)
- ConfigureIEEE1588
 - ActionCommand.cpp, [1415](#)
- ConfigureImageEvents
 - ImageEvents.cpp, [1446](#)
- ConfigureInference
 - Inference.cpp, [1453](#)
- ConfigureInterface
 - ActionCommand.cpp, [1415](#)
- ConfigureLogicBlock
 - LogicBlock.cpp, [1458](#)
- ConfigureLookupTables
 - LookupTable.cpp, [1460](#)
- ConfigureOtherNodes
 - ActionCommand.cpp, [1415](#)
- ConfigureSequencerPartOne
 - Sequencer.cpp, [1472](#)
- ConfigureSequencerPartTwo
 - Sequencer.cpp, [1472](#)
- ConfigureStream
 - Polarization.cpp, [1467](#)
- ConfigureTestPattern
 - Inference.cpp, [1453](#)
- ConfigureTrigger
 - ActionCommand.cpp, [1415](#)
 - BufferHandling.cpp, [1417](#)
 - Inference.cpp, [1453](#)
 - LogicBlock.cpp, [1458](#)
 - Trigger.cpp, [1479](#)
 - Trigger_QuickSpin.cpp, [1481](#)
- ConfigureUserSet1
 - AcquisitionMultipleCameraRecovery.cpp, [1408](#)
- Connect
 - NodeMap, [1052](#)
 - Spinnaker::GenApi, [366](#)
- Container Class, [112](#)
- ContentType_Xml
 - Spinnaker::GenApi, [363](#)
- ContentType_ZippedXml
 - Spinnaker::GenApi, [363](#)
- Conversion, [189](#)
 - NumToCString, [189](#)
- Convert
 - ImageProcessor, [862](#), [863](#)
 - ImageProcessor, [935–937](#)
- Counter, [688](#)
 - Counter, [688](#)
 - GetValue, [688](#)
 - IsZero, [689](#)
 - operator unsigned int, [689](#)
 - operator++, [689](#)
 - operator--, [689](#)
- Counter Class, [113](#)
- CounterAndTimer.cpp
 - AcquireImages, [1424](#)
 - ConfigureDigitalIO, [1424](#)
 - ConfigureExposureandTrigger, [1424](#)
 - main, [1425](#)
 - PrintDeviceInfo, [1425](#)
 - ResetTrigger, [1425](#)
 - RunSingleCamera, [1425](#)
 - SetupCounterAndTimer, [1425](#)
- CounterDelay
 - Camera, [468](#)
- CounterDuration
 - Camera, [468](#)
- CounterEventActivation
 - Camera, [468](#)
- CounterEventActivation_AnyEdge
 - Spinnaker, [253](#)
- CounterEventActivation_FallingEdge
 - Spinnaker, [253](#)
- CounterEventActivation_LevelHigh
 - Spinnaker, [253](#)
- CounterEventActivation_LevelLow
 - Spinnaker, [253](#)
- CounterEventActivation_RisingEdge
 - Spinnaker, [253](#)
- CounterEventActivationEnums
 - Spinnaker, [252](#)
- CounterEventSource
 - Camera, [468](#)
- CounterEventSource_Counter0End
 - Spinnaker, [253](#)
- CounterEventSource_Counter0Start
 - Spinnaker, [253](#)
- CounterEventSource_Counter1End
 - Spinnaker, [253](#)
- CounterEventSource_Counter1Start
 - Spinnaker, [253](#)
- CounterEventSource_ExposureEnd
 - Spinnaker, [253](#)
- CounterEventSource_ExposureStart
 - Spinnaker, [253](#)
- CounterEventSource_FrameTriggerWait
 - Spinnaker, [253](#)
- CounterEventSource_Line0
 - Spinnaker, [253](#)
- CounterEventSource_Line1
 - Spinnaker, [253](#)
- CounterEventSource_Line2
 - Spinnaker, [253](#)
- CounterEventSource_Line3
 - Spinnaker, [253](#)

- CounterEventSource_LogicBlock0
 - Spinnaker, [253](#)
- CounterEventSource_LogicBlock1
 - Spinnaker, [253](#)
- CounterEventSource_MHzTick
 - Spinnaker, [253](#)
- CounterEventSource_Off
 - Spinnaker, [253](#)
- CounterEventSource_UserOutput0
 - Spinnaker, [253](#)
- CounterEventSource_UserOutput1
 - Spinnaker, [253](#)
- CounterEventSource_UserOutput2
 - Spinnaker, [253](#)
- CounterEventSource_UserOutput3
 - Spinnaker, [253](#)
- CounterEventSourceEnums
 - Spinnaker, [253](#)
- CounterReset
 - Camera, [468](#)
- CounterResetActivation
 - Camera, [469](#)
- CounterResetActivation_AnyEdge
 - Spinnaker, [254](#)
- CounterResetActivation_FallingEdge
 - Spinnaker, [254](#)
- CounterResetActivation_LevelHigh
 - Spinnaker, [254](#)
- CounterResetActivation_LevelLow
 - Spinnaker, [254](#)
- CounterResetActivation_RisingEdge
 - Spinnaker, [254](#)
- CounterResetActivationEnums
 - Spinnaker, [253](#)
- CounterResetSource
 - Camera, [469](#)
- CounterResetSource_Counter0End
 - Spinnaker, [254](#)
- CounterResetSource_Counter0Start
 - Spinnaker, [254](#)
- CounterResetSource_Counter1End
 - Spinnaker, [254](#)
- CounterResetSource_Counter1Start
 - Spinnaker, [254](#)
- CounterResetSource_ExposureEnd
 - Spinnaker, [254](#)
- CounterResetSource_ExposureStart
 - Spinnaker, [254](#)
- CounterResetSource_FrameTriggerWait
 - Spinnaker, [254](#)
- CounterResetSource_Line0
 - Spinnaker, [254](#)
- CounterResetSource_Line1
 - Spinnaker, [254](#)
- CounterResetSource_Line2
 - Spinnaker, [254](#)
- CounterResetSource_Line3
 - Spinnaker, [254](#)
- CounterResetSource_LogicBlock0
 - Spinnaker, [254](#)
- CounterResetSource_LogicBlock1
 - Spinnaker, [254](#)
- CounterResetSource_Off
 - Spinnaker, [254](#)
- CounterResetSource_UserOutput0
 - Spinnaker, [254](#)
- CounterResetSource_UserOutput1
 - Spinnaker, [254](#)
- CounterResetSource_UserOutput2
 - Spinnaker, [254](#)
- CounterResetSource_UserOutput3
 - Spinnaker, [254](#)
- CounterResetSourceEnums
 - Spinnaker, [254](#)
- CounterSelector
 - Camera, [469](#)
- CounterSelector_Counter0
 - Spinnaker, [255](#)
- CounterSelector_Counter1
 - Spinnaker, [255](#)
- CounterSelectorEnums
 - Spinnaker, [254](#)
- CounterStart
 - PerformanceCounter, [191](#)
- CounterStatus
 - Camera, [469](#)
- CounterStatus_CounterActive
 - Spinnaker, [255](#)
- CounterStatus_CounterCompleted
 - Spinnaker, [255](#)
- CounterStatus_CounterIdle
 - Spinnaker, [255](#)
- CounterStatus_CounterOverflow
 - Spinnaker, [255](#)
- CounterStatus_CounterTriggerWait
 - Spinnaker, [255](#)
- CounterStatusEnums
 - Spinnaker, [255](#)
- CounterTriggerActivation
 - Camera, [469](#)
- CounterTriggerActivation_AnyEdge
 - Spinnaker, [255](#)
- CounterTriggerActivation_FallingEdge
 - Spinnaker, [255](#)
- CounterTriggerActivation_LevelHigh
 - Spinnaker, [255](#)
- CounterTriggerActivation_LevelLow
 - Spinnaker, [255](#)
- CounterTriggerActivation_RisingEdge
 - Spinnaker, [255](#)
- CounterTriggerActivationEnums
 - Spinnaker, [255](#)
- CounterTriggerSource
 - Camera, [469](#)
- CounterTriggerSource_Counter0End
 - Spinnaker, [256](#)

- CounterTriggerSource_Counter0Start
 Spinnaker, [256](#)
- CounterTriggerSource_Counter1End
 Spinnaker, [256](#)
- CounterTriggerSource_Counter1Start
 Spinnaker, [256](#)
- CounterTriggerSource_ExposureEnd
 Spinnaker, [256](#)
- CounterTriggerSource_ExposureStart
 Spinnaker, [256](#)
- CounterTriggerSource_FrameTriggerWait
 Spinnaker, [256](#)
- CounterTriggerSource_Line0
 Spinnaker, [256](#)
- CounterTriggerSource_Line1
 Spinnaker, [256](#)
- CounterTriggerSource_Line2
 Spinnaker, [256](#)
- CounterTriggerSource_Line3
 Spinnaker, [256](#)
- CounterTriggerSource_LogicBlock0
 Spinnaker, [256](#)
- CounterTriggerSource_LogicBlock1
 Spinnaker, [256](#)
- CounterTriggerSource_Off
 Spinnaker, [256](#)
- CounterTriggerSource_UserOutput0
 Spinnaker, [256](#)
- CounterTriggerSource_UserOutput1
 Spinnaker, [256](#)
- CounterTriggerSource_UserOutput2
 Spinnaker, [256](#)
- CounterTriggerSource_UserOutput3
 Spinnaker, [256](#)
- CounterTriggerSourceEnums
 Spinnaker, [255](#)
- CounterValue
 Camera, [470](#)
- CounterValueAtReset
 Camera, [470](#)
- CPointer
 CPointer< T, B >, [691](#)
- CPointer< T, B >, [690](#)
 ~CPointer, [691](#)
 CPointer, [691](#)
 IsValid, [691](#)
 m_pT, [694](#)
 operator bool, [691](#)
 operator T*, [692](#)
 operator!=, [692](#)
 operator*, [693](#)
 operator(), [693](#)
 operator->, [693](#)
 operator=, [693](#)
 operator==, [693](#), [694](#)
- CPortConstructPtr
 Pointer Class, [168](#)
- CPortImpl, [694](#)
 ~CPortImpl, [696](#)
 CPortImpl, [695](#)
 GetAccessMode, [696](#)
 GetSwapEndianness, [696](#)
 InvalidateNode, [696](#)
 m_ptrPort, [697](#)
 Read, [696](#)
 Replay, [697](#)
 SetPortImpl, [697](#)
 Write, [697](#)
- CPortPtr
 Pointer Class, [168](#)
- CPortRecorderPtr
 Pointer Class, [169](#)
- CPortRecorderRef
 Spinnaker::GenApi, [354](#)
- CPortRef
 Spinnaker::GenApi, [355](#)
- CPortReplayPtr
 Pointer Class, [169](#)
- CPortWriteList, [698](#)
 ~CPortWriteList, [699](#)
 CPortWriteList, [699](#)
 GetCookie, [699](#)
 GetPortWriteListHandle, [700](#)
 m_pWriteList, [700](#)
 Replay, [700](#)
 SetCookie, [700](#)
 Write, [700](#)
- CPortWriteListPtr
 Pointer Class, [169](#)
- CpuUsageInfo, [701](#)
 dummy, [701](#)
- cpuUsageInfo
 GigEVisionPerformance.cpp, [1444](#)
- CpuUtil, [190](#)
 GetCpuStats, [190](#)
 StartCpuTracing, [190](#)
 StopCpuTracing, [190](#)
- CRCChecksum
 DCAM_CHECKSUM, [712](#)
- Create
 Image, [891](#), [892](#)
- CreateAndSaveAolpDolpImages
 Polarization.cpp, [1468](#)
- CreateAndSaveGlareReducedImage
 Polarization.cpp, [1468](#)
- CreateAndSaveStokesImages
 Polarization.cpp, [1468](#)
- CreateAolp
 ImageUtilityPolarization, [960](#), [961](#)
- CreateColorCorrected
 ImageUtilityCCM, [954](#), [955](#)
- CreateDolp
 ImageUtilityPolarization, [961](#), [962](#)
- CreateEmptyNodeMap
 CNodeMapFactory, [671](#)
- CreateFiles

- AcquisitionMultipleCamerasWriteToFile.cpp, [1411](#)
- CreateGlareReduced
 - ImageUtilityPolarization, [962](#), [963](#)
- CreateHeatmap
 - ImageUtilityHeatmap, [957](#)
- CreateHeatmapImages
 - Polarization.cpp, [1468](#)
- CreateNodeDataFromNodeMap
 - CNodeMapFactory, [671](#)
- CreateNodeMap
 - CNodeMapFactory, [671](#)
- CreateNormalized
 - ImageUtility, [950–953](#)
- CreateNormalizedImage
 - Polarization.cpp, [1468](#)
- CreateScaled
 - ImageUtility, [953](#)
- CreateShared
 - Image, [892](#)
- CreateStokesS0
 - ImageUtilityPolarization, [963](#), [964](#)
- CreateStokesS1
 - ImageUtilityPolarization, [964](#)
- CreateStokesS2
 - ImageUtilityPolarization, [965](#)
- CRegisterPortImpl, [701](#)
 - ~CRegisterPortImpl, [703](#)
 - CRegisterPortImpl, [702](#)
 - GetAccessMode, [703](#)
 - Read, [703](#)
 - ReadRegister, [703](#)
 - SetPortImpl, [704](#)
 - Write, [704](#)
 - WriteRegister, [704](#)
- CRegisterPtr
 - Pointer Class, [169](#)
- CRegisterRef
 - Spinnaker::GenApi, [355](#)
- CSelectorPtr
 - Pointer Class, [169](#)
- CSelectorRef
 - Spinnaker::GenApi, [355](#)
- CSelectorSet, [705](#)
 - ~CSelectorSet, [706](#)
 - CSelectorSet, [706](#)
 - GetSelectorList, [706](#)
 - IsEmpty, [706](#)
 - Restore, [707](#)
 - SetFirst, [707](#)
 - SetNext, [707](#)
 - ToString, [707](#)
- CStringPtr
 - Pointer Class, [169](#)
- CStringRef
 - Spinnaker::GenApi, [355](#)
- ctDependingNodes
 - Spinnaker::GenApi, [359](#)
- CTestPortStruct
 - CTestPortStruct< CDataStruct >, [709](#)
 - CTestPortStruct< CDataStruct >, [708](#)
 - CTestPortStruct, [709](#)
 - GetAccessMode, [709](#)
 - GetNumReads, [709](#)
 - GetNumWrites, [710](#)
 - GetPrincipalInterfaceType, [710](#)
 - m_BaseAddress, [711](#)
 - m_NumReads, [711](#)
 - m_NumWrites, [711](#)
 - MemSet, [710](#)
 - Read, [710](#)
 - ResetStatistics, [710](#)
 - Write, [710](#)
- ctInvalidatingChildren
 - Spinnaker::GenApi, [359](#)
- ctParentNodes
 - Spinnaker::GenApi, [359](#)
- ctReadingChildren
 - Spinnaker::GenApi, [359](#)
- ctTerminalNodes
 - Spinnaker::GenApi, [359](#)
- ctWritingChildren
 - Spinnaker::GenApi, [359](#)
- Custom
 - Spinnaker::GenApi, [360](#)
- CustomCCMCode
 - CCMSettings, [611](#)
- CValuePtr
 - Pointer Class, [170](#)
- CValueRef
 - Spinnaker::GenApi, [355](#)
- CxpConnectionSelector
 - Camera, [470](#)
- CxpConnectionTestErrorCount
 - Camera, [470](#)
- CxpConnectionTestMode
 - Camera, [470](#)
- CxpConnectionTestMode_Mode1
 - Spinnaker, [256](#)
- CxpConnectionTestMode_Off
 - Spinnaker, [256](#)
- CxpConnectionTestModeEnums
 - Spinnaker, [256](#)
- CxpConnectionTestPacketCount
 - Camera, [470](#)
- CxpLinkConfiguration
 - Camera, [471](#)
- CxpLinkConfiguration_Auto
 - Spinnaker, [257](#)
- CxpLinkConfiguration_CXP1_X1
 - Spinnaker, [257](#)
- CxpLinkConfiguration_CXP1_X2
 - Spinnaker, [257](#)
- CxpLinkConfiguration_CXP1_X3
 - Spinnaker, [257](#)
- CxpLinkConfiguration_CXP1_X4
 - Spinnaker, [257](#)

- CxpLinkConfiguration_CXP1_X5
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP1_X6
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X1
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X2
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X3
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X4
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X5
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP2_X6
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X1
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X2
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X3
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X4
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X5
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP3_X6
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X1
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X2
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X3
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X4
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X5
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP5_X6
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X1
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X2
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X3
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X4
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X5
Spinnaker, [257](#)
- CxpLinkConfiguration_CXP6_X6
Spinnaker, [257](#)
- CxpLinkConfigurationEnums
Spinnaker, [256](#)
- CxpLinkConfigurationPreferred
Camera, [471](#)
- CxpLinkConfigurationPreferred_CXP1_X1
Spinnaker, [257](#)
- CxpLinkConfigurationPreferred_CXP1_X2
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X3
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X4
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X5
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X6
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X1
Spinnaker, [257](#)
- CxpLinkConfigurationPreferred_CXP2_X2
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X3
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X4
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X5
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X6
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X1
Spinnaker, [257](#)
- CxpLinkConfigurationPreferred_CXP3_X2
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X3
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X4
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X5
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X6
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X1
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X2
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X3
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X4
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X5
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X6
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X1
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X2
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X3
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X4
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X5
Spinnaker, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X6
Spinnaker, [258](#)

- CxpLinkConfigurationPreferredEnums
 - Spinnaker, [257](#)
- CxpLinkConfigurationStatus
 - Camera, [471](#)
- CxpLinkConfigurationStatus_CXP1_X1
 - Spinnaker, [258](#)
- CxpLinkConfigurationStatus_CXP1_X2
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP1_X3
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP1_X4
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP1_X5
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP1_X6
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP2_X1
 - Spinnaker, [258](#)
- CxpLinkConfigurationStatus_CXP2_X2
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP2_X3
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP2_X4
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP2_X5
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP2_X6
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP3_X1
 - Spinnaker, [258](#)
- CxpLinkConfigurationStatus_CXP3_X2
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP3_X3
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP3_X4
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP3_X5
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP3_X6
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X1
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X2
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X3
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X4
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X5
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP5_X6
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X1
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X2
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X3
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X4
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X5
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_CXP6_X6
 - Spinnaker, [259](#)
- CxpLinkConfigurationStatus_None
 - Spinnaker, [258](#)
- CxpLinkConfigurationStatus_Pending
 - Spinnaker, [258](#)
- CxpLinkConfigurationStatusEnums
 - Spinnaker, [258](#)
- CxpPoCxpAuto
 - Camera, [471](#)
- CxpPoCxpStatus
 - Camera, [471](#)
- CxpPoCxpStatus_Auto
 - Spinnaker, [259](#)
- CxpPoCxpStatus_Off
 - Spinnaker, [259](#)
- CxpPoCxpStatus_Tripped
 - Spinnaker, [259](#)
- CxpPoCxpStatusEnums
 - Spinnaker, [259](#)
- CxpPoCxpTripReset
 - Camera, [471](#)
- CxpPoCxpTurnOff
 - Camera, [472](#)
- Data
 - GVCP_EVENTDATA_REQUEST, [793](#)
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [794](#)
- DATA_BITS
 - SerialRxTx.cpp, [1475](#)
- DCAM_CHECKSUM, [711](#)
 - CRCChecksum, [712](#)
- DCAM_CHUNK_TRAILER, [712](#)
 - ChunkID, [712](#)
 - ChunkLength, [712](#)
 - InverseChunkLength, [712](#)
- DDR
 - Inference.cpp, [1451](#)
- DecimationHorizontal
 - Camera, [472](#)
- DecimationHorizontalMode
 - Camera, [472](#)
- DecimationHorizontalMode_Discard
 - Spinnaker, [260](#)
- DecimationHorizontalModeEnums
 - Spinnaker, [259](#)
- DecimationSelector
 - Camera, [472](#)
- DecimationSelector_All
 - Spinnaker, [260](#)
- DecimationSelector_Sensor
 - Spinnaker, [260](#)
- DecimationSelectorEnums
 - Spinnaker, [260](#)

- DecimationVertical
 - Camera, [472](#)
- DecimationVerticalMode
 - Camera, [473](#)
- DecimationVerticalMode_Discard
 - Spinnaker, [260](#)
- DecimationVerticalModeEnums
 - Spinnaker, [260](#)
- Decreasing
 - Spinnaker::GenApi, [361](#)
- DeepCopy
 - Image, [842](#)
 - Image, [892](#), [893](#)
- DefectCorrectionMode
 - Camera, [473](#)
- DefectCorrectionMode_Average
 - Spinnaker, [261](#)
- DefectCorrectionMode_Highlight
 - Spinnaker, [261](#)
- DefectCorrectionMode_Zero
 - Spinnaker, [261](#)
- DefectCorrectionModeEnums
 - Spinnaker, [260](#)
- DefectCorrectStaticEnable
 - Camera, [473](#)
- DefectTableApply
 - Camera, [473](#)
- DefectTableCoordinateX
 - Camera, [474](#)
- DefectTableCoordinateY
 - Camera, [474](#)
- DefectTableFactoryRestore
 - Camera, [474](#)
- DefectTableIndex
 - Camera, [474](#)
- DefectTablePixelCount
 - Camera, [475](#)
- DefectTableSave
 - Camera, [475](#)
- Delnit
 - CameraBase, [569](#)
 - ICameraBase, [800](#)
- Deinterlacing
 - Camera, [475](#)
- Deinterlacing_LineDuplication
 - Spinnaker, [261](#)
- Deinterlacing_Off
 - Spinnaker, [261](#)
- Deinterlacing_Weave
 - Spinnaker, [261](#)
- DeinterlacingEnums
 - Spinnaker, [261](#)
- deleteFile
 - FileProtocolAdapter, [762](#)
- DeleteFileOnCamera
 - Inference.cpp, [1453](#)
- DeliverEventMessage
 - CEventAdapter1394, [621](#)
 - CEventAdapterGEV, [625](#)
 - CEventAdapterU3V, [627](#)
- DeliverMessage
 - CEventAdapter, [618](#)
 - CEventAdapter1394, [621](#)
 - CEventAdapterGeneric, [623](#)
 - CEventAdapterGEV, [625](#)
 - CEventAdapterU3V, [627](#)
- DEPRECATED_CLASS
 - Spinnaker, [336](#)
- Deregister
 - Spinnaker::GenApi, [366](#)
- DeregisterCallback
 - Node, [1041](#)
 - Spinnaker::GenApi, [366](#)
- Destroy
 - CNodeCallback, [664](#)
 - Function_NodeCallback< Function >, [775](#)
 - Member_NodeCallback< Client, Member >, [1035](#)
 - NodeMap, [1052](#)
- DetachBuffer
 - CChunkAdapter, [595](#)
- DetachChunk
 - CChunkPort, [608](#)
- DetachEvent
 - CEventPort, [631](#)
- DetachNode
 - CEventPort, [631](#)
- DetachNodeMap
 - CChunkAdapter, [595](#)
 - CEventAdapter, [619](#)
- DetachPort
 - CChunkPort, [608](#)
- DETECTION
 - Inference.cpp, [1452](#)
- DeviceAccessStatus
 - TransportLayerDevice, [1122](#)
 - TransportLayerInterface, [1133](#)
- DeviceAccessStatus_Busy
 - Spinnaker, [261](#)
- DeviceAccessStatus_NoAccess
 - Spinnaker, [261](#)
- DeviceAccessStatus_OpenReadOnly
 - Spinnaker, [261](#)
- DeviceAccessStatus_OpenReadWrite
 - Spinnaker, [261](#)
- DeviceAccessStatus_ReadOnly
 - Spinnaker, [261](#)
- DeviceAccessStatus_ReadWrite
 - Spinnaker, [261](#)
- DeviceAccessStatus_Unknown
 - Spinnaker, [261](#)
- DeviceAccessStatusEnum
 - Spinnaker, [261](#)
- DeviceAddress
 - ActionCommandResult, [401](#)
- DeviceArrivalEventHandler, [713](#)
 - ~DeviceArrivalEventHandler, [714](#)

- DeviceArrivalEventHandler, 714
 - OnDeviceArrival, 714
 - operator=, 715
- DeviceArrivalEventHandler Class, 52
- DeviceBootloaderVersion
 - TransportLayerDevice, 1122
- DeviceCharacterSet
 - Camera, 475
- DeviceCharacterSet_ASCII
 - Spinnaker, 262
- DeviceCharacterSet_UTF8
 - Spinnaker, 262
- DeviceCharacterSetEnums
 - Spinnaker, 261
- DeviceClockFrequency
 - Camera, 476
- DeviceClockSelector
 - Camera, 476
- DeviceClockSelector_CameraLink
 - Spinnaker, 262
- DeviceClockSelector_Sensor
 - Spinnaker, 262
- DeviceClockSelector_SensorDigitization
 - Spinnaker, 262
- DeviceClockSelectorEnums
 - Spinnaker, 262
- DeviceConnectionSelector
 - Camera, 476
- DeviceConnectionSpeed
 - Camera, 476
- DeviceConnectionStatus
 - Camera, 476
- DeviceConnectionStatus_Active
 - Spinnaker, 262
- DeviceConnectionStatus_Inactive
 - Spinnaker, 262
- DeviceConnectionStatusEnums
 - Spinnaker, 262
- DeviceCount
 - TransportLayerInterface, 1134
- DeviceCurrentSpeed
 - TransportLayerDevice, 1123
- DeviceCurrentSpeed_FullSpeed
 - Spinnaker, 263
- DeviceCurrentSpeed_HighSpeed
 - Spinnaker, 263
- DeviceCurrentSpeed_LowSpeed
 - Spinnaker, 263
- DeviceCurrentSpeed_SuperSpeed
 - Spinnaker, 263
- DeviceCurrentSpeed_UnknownSpeed
 - Spinnaker, 263
- DeviceCurrentSpeedEnum
 - Spinnaker, 262
- DeviceDisplayName
 - TransportLayerDevice, 1123
- DeviceDriverVersion
 - TransportLayerDevice, 1123
- DeviceEndiannessMechanism
 - TransportLayerDevice, 1123
- DeviceEndiannessMechanism_Legacy
 - Spinnaker, 263
- DeviceEndiannessMechanism_Standard
 - Spinnaker, 263
- DeviceEndiannessMechanismEnum
 - Spinnaker, 263
- DeviceEventChannelCount
 - Camera, 477
- DeviceEventExposureEndData, 715
 - frameID, 715
- DeviceEventHandler, 716
 - ~DeviceEventHandler, 717
 - DeviceEventHandler, 717
 - GetDeviceEventId, 717
 - GetDeviceEventName, 718
 - OnDeviceEvent, 718
 - operator=, 718
- DeviceEventHandler Class, 53
- DeviceEventHandlerImpl, 719
 - ~DeviceEventHandlerImpl, 720
 - DeviceEventHandlerImpl, 720
 - OnDeviceEvent, 721
- DeviceEventInferenceData, 721
 - confidence, 721
 - frameID, 721
 - result, 722
- DeviceEvents.cpp
 - AcquireImages, 1426
 - chosenEvent, 1428
 - ConfigureDeviceEvents, 1427
 - eventType, 1426
 - GENERIC, 1426
 - InferenceAvailable, 1427
 - main, 1427
 - PrintDeviceInfo, 1427
 - ResetDeviceEvents, 1427
 - RunSingleCamera, 1427
 - SPECIFIC, 1426
- DeviceEventUtility, 722
 - ParseDeviceEventExposureEnd, 722
 - ParseDeviceEventInference, 723
- DeviceFamilyName
 - Camera, 477
- DeviceFeaturePersistenceEnd
 - Camera, 477
- DeviceFeaturePersistenceStart
 - Camera, 477
- DeviceFirmwareVersion
 - Camera, 477
- DeviceGenCPVersionMajor
 - Camera, 478
- DeviceGenCPVersionMinor
 - Camera, 478
- DeviceID
 - Camera, 478
 - TransportLayerDevice, 1123

- TransportLayerInterface, [1134](#)
- DeviceIndicatorMode
 - Camera, [478](#)
- DeviceIndicatorMode_Active
 - Spinnaker, [263](#)
- DeviceIndicatorMode_ErrorStatus
 - Spinnaker, [263](#)
- DeviceIndicatorMode_Inactive
 - Spinnaker, [263](#)
- DeviceIndicatorModeEnums
 - Spinnaker, [263](#)
- DeviceInstanceId
 - TransportLayerDevice, [1123](#)
- DevicesUpdater
 - TransportLayerDevice, [1124](#)
- DeviceLinkBandwidthReserve
 - Camera, [478](#)
- DeviceLinkCommandTimeout
 - Camera, [478](#)
- DeviceLinkConnectionCount
 - Camera, [479](#)
- DeviceLinkCurrentThroughput
 - Camera, [479](#)
- DeviceLinkHeartbeatMode
 - Camera, [479](#)
- DeviceLinkHeartbeatMode_Off
 - Spinnaker, [264](#)
- DeviceLinkHeartbeatMode_On
 - Spinnaker, [264](#)
- DeviceLinkHeartbeatModeEnums
 - Spinnaker, [263](#)
- DeviceLinkHeartbeatTimeout
 - Camera, [479](#)
- DeviceLinkSelector
 - Camera, [479](#)
- DeviceLinkSpeed
 - Camera, [479](#)
 - TransportLayerDevice, [1124](#)
- DeviceLinkThroughputLimit
 - Camera, [480](#)
- DeviceLinkThroughputLimitMode
 - Camera, [480](#)
- DeviceLinkThroughputLimitMode_Off
 - Spinnaker, [264](#)
- DeviceLinkThroughputLimitMode_On
 - Spinnaker, [264](#)
- DeviceLinkThroughputLimitModeEnums
 - Spinnaker, [264](#)
- DeviceLocation
 - TransportLayerDevice, [1124](#)
- DeviceManifestEntrySelector
 - Camera, [480](#)
- DeviceManifestPrimaryURL
 - Camera, [480](#)
- DeviceManifestSchemaMajorVersion
 - Camera, [480](#)
- DeviceManifestSchemaMinorVersion
 - Camera, [481](#)
- DeviceManifestSecondaryURL
 - Camera, [481](#)
- DeviceManifestXMLMajorVersion
 - Camera, [481](#)
- DeviceManifestXMLMinorVersion
 - Camera, [481](#)
- DeviceManifestXMLSubMinorVersion
 - Camera, [481](#)
- DeviceManufacturerInfo
 - Camera, [481](#)
- DeviceMaxThroughput
 - Camera, [482](#)
- DeviceModelName
 - Camera, [482](#)
 - TransportLayerDevice, [1124](#)
 - TransportLayerInterface, [1134](#)
- DeviceMulticastMonitorMode
 - TransportLayerDevice, [1124](#)
- DevicePortId
 - TransportLayerDevice, [1124](#)
- DevicePowerSupplySelector
 - Camera, [482](#)
- DevicePowerSupplySelector_External
 - Spinnaker, [264](#)
- DevicePowerSupplySelectorEnums
 - Spinnaker, [264](#)
- DeviceRegistersCheck
 - Camera, [482](#)
- DeviceRegistersEndianness
 - Camera, [482](#)
- DeviceRegistersEndianness_Big
 - Spinnaker, [264](#)
- DeviceRegistersEndianness_Little
 - Spinnaker, [264](#)
- DeviceRegistersEndiannessEnums
 - Spinnaker, [264](#)
- DeviceRegistersStreamingEnd
 - Camera, [483](#)
- DeviceRegistersStreamingStart
 - Camera, [483](#)
- DeviceRegistersValid
 - Camera, [483](#)
- DeviceRemovalEventHandler, [723](#)
 - ~DeviceRemovalEventHandler, [725](#)
 - DeviceRemovalEventHandler, [724](#)
 - OnDeviceRemoval, [725](#)
 - operator=, [725](#)
- DeviceRemovalEventHandler Class, [54](#)
- DeviceReset
 - Camera, [483](#)
 - TransportLayerDevice, [1125](#)
- DeviceScanType
 - Camera, [483](#)
- DeviceScanType_Areascan
 - Spinnaker, [265](#)
- DeviceScanTypeEnums
 - Spinnaker, [265](#)
- DeviceSelector

- TransportLayerInterface, [1134](#)
- DeviceSerialNumber
 - Camera, [483](#)
 - TransportLayerDevice, [1125](#)
 - TransportLayerInterface, [1134](#)
- DeviceSerialPortBaudRate
 - Camera, [484](#)
- DeviceSerialPortBaudRate_Baud115200
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud19200
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud230400
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud38400
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud460800
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud57600
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud921600
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRate_Baud9600
 - Spinnaker, [265](#)
- DeviceSerialPortBaudRateEnums
 - Spinnaker, [265](#)
- DeviceSerialPortSelector
 - Camera, [484](#)
- DeviceSerialPortSelector_CameraLink
 - Spinnaker, [265](#)
- DeviceSerialPortSelectorEnums
 - Spinnaker, [265](#)
- DeviceSFNCVersionMajor
 - Camera, [484](#)
- DeviceSFNCVersionMinor
 - Camera, [484](#)
- DeviceSFNCVersionSubMinor
 - Camera, [484](#)
- DeviceStreamChannelCount
 - Camera, [485](#)
- DeviceStreamChannelEndianness
 - Camera, [485](#)
- DeviceStreamChannelEndianness_Big
 - Spinnaker, [266](#)
- DeviceStreamChannelEndianness_Little
 - Spinnaker, [266](#)
- DeviceStreamChannelEndiannessEnums
 - Spinnaker, [265](#)
- DeviceStreamChannelLink
 - Camera, [485](#)
- DeviceStreamChannelPacketSize
 - Camera, [485](#)
- DeviceStreamChannelSelector
 - Camera, [485](#)
- DeviceStreamChannelType
 - Camera, [486](#)
- DeviceStreamChannelType_Receiver
 - Spinnaker, [266](#)
- DeviceStreamChannelType_Transmitter
 - Spinnaker, [266](#)
- DeviceStreamChannelTypeEnums
 - Spinnaker, [266](#)
- DeviceTapGeometry
 - Camera, [486](#)
- DeviceTapGeometry_Geometry_10X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_10X_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X10
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X10_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X2
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X2_1Y
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_1X2_1Y2
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_1X2_2YE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X3
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X3_1Y
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_1X4
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X4_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X8
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X8_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_1X_1Y
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_1X_1Y2
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_1X_2YE
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2E
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2E_2YE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X2M
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2X_1Y
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y
 - Spinnaker, [266](#)

- Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2X_2YE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2XE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2XE_1Y2
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2XE_2YE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2XM
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_2XM_1Y
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2XM_1Y2
 - Spinnaker, [266](#)
- DeviceTapGeometry_Geometry_2XM_2YE
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_3X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_3X_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X2
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X2_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X2E
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X2E_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_4X_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_8X
 - Spinnaker, [267](#)
- DeviceTapGeometry_Geometry_8X_1Y
 - Spinnaker, [267](#)
- DeviceTapGeometryEnums
 - Spinnaker, [266](#)
- DeviceTemperature
 - Camera, [486](#)
- DeviceTemperatureSelector
 - Camera, [486](#)
- DeviceTemperatureSelector_Sensor
 - Spinnaker, [268](#)
- DeviceTemperatureSelectorEnums
 - Spinnaker, [267](#)
- DeviceTLType
 - Camera, [486](#)
- DeviceTLType_CameraLink
 - Spinnaker, [268](#)
- DeviceTLType_CameraLinkHS
 - Spinnaker, [268](#)
- DeviceTLType_CoaXPress
 - Spinnaker, [268](#)
- DeviceTLType_Custom
 - Spinnaker, [268](#)
- DeviceTLType_GigEVision
 - Spinnaker, [268](#)
- DeviceTLType_USB3Vision
 - Spinnaker, [268](#)
- DeviceTLTypeEnums
 - Spinnaker, [268](#)
- DeviceTLVersionMajor
 - Camera, [487](#)
- DeviceTLVersionMinor
 - Camera, [487](#)
- DeviceTLVersionSubMinor
 - Camera, [487](#)
- DeviceType
 - Camera, [487](#)
 - TransportLayerDevice, [1125](#)
- DeviceType_CameraLink
 - Spinnaker, [268](#)
- DeviceType_CameraLinkHS
 - Spinnaker, [268](#)
- DeviceType_CoaXPress
 - Spinnaker, [268](#)
- DeviceType_Custom
 - Spinnaker, [268](#)
- DeviceType_GigEVision
 - Spinnaker, [268](#)
- DeviceType_Peripheral
 - Spinnaker, [270](#)
- DeviceType_Receiver
 - Spinnaker, [270](#)
- DeviceType_Transceiver
 - Spinnaker, [270](#)
- DeviceType_Transmitter
 - Spinnaker, [270](#)
- DeviceType_USB3Vision
 - Spinnaker, [268](#)
- DeviceTypeEnum
 - Spinnaker, [268](#)
- DeviceTypeEnums
 - Spinnaker, [268](#)
- DeviceU3VProtocol
 - TransportLayerDevice, [1125](#)
- DeviceUnlock
 - TransportLayerInterface, [1134](#)
- DeviceUpdateList
 - TransportLayerInterface, [1135](#)
- DeviceUptime
 - Camera, [487](#)
- DeviceUserID
 - Camera, [488](#)
 - TransportLayerDevice, [1125](#)
- DeviceVendorName
 - Camera, [488](#)
 - TransportLayerDevice, [1125](#)
 - TransportLayerInterface, [1135](#)
- DeviceVersion
 - Camera, [488](#)
 - TransportLayerDevice, [1126](#)
- DisableAll
 - ImageStatistics, [866](#)

- ImageStatistics, 945
- DisableChunkData
 - ChunkData.cpp, 1420
 - Inference.cpp, 1453
- DisableGVCPHeartbeat
 - Acquisition.cpp, 1365
 - AcquisitionMultipleThread.cpp, 1413
 - Compression.cpp, 1422
- DisableImageChunkData
 - Compression.cpp, 1422
- DisableImageCompression
 - Compression.cpp, 1422
- DisableTrigger
 - Inference.cpp, 1454
- DiscoverMaxPacketSize
 - CameraBase, 569
 - ICameraBase, 801
- DisplayChunkData
 - ChunkData.cpp, 1420
 - Inference.cpp, 1454
- doc/spindocs/C++/GettingStarted.dox, 1167
- doc/spindocs/C++/ProgrammerGuide.dox, 1167
- doc/spindocs/shared/Benefits.dox, 1167
- doc/spindocs/shared/FlyCapture2Comparison.dox, 1167
- doc/spindocs/shared/GenICamGenTL.dox, 1167
- doc/spindocs/shared/Licensing.dox, 1167
- doc/spindocs/shared/Maintenance.dox, 1167
- DoesEnvironmentVariableExist
 - Spinnaker::GenICam, 396
- double_autovector_t, 725
 - _pCount, 728
 - _pv, 728
 - ~double_autovector_t, 726
 - double_autovector_t, 726
 - operator delete, 727
 - operator new, 727
 - operator=, 727
 - operator[], 727
 - size, 727
- DownloadImage
 - FileAccess_QuickSpin.cpp, 1436
- dummy
 - CpuUsageInfo, 701
- EAccessModeClass, 728
 - FromString, 728
 - ToString, 729
- EatComments
 - Spinnaker::GenApi, 366
- ECacheUsage_t
 - Spinnaker::GenApi, 362
- ECachingModeClass, 729
 - FromString, 730
 - ToString, 730
- EContentType_t
 - Spinnaker::GenApi, 363
- EDisplayNotationClass, 730
 - FromString, 731
 - ToString, 731
- EEndianessClass, 731
 - FromString, 732
 - ToString, 732
- EGenApiSchemaVersionClass, 733
 - FromString, 733
 - ToString, 733
- EInputDirectionClass, 734
 - FromString, 734
 - ToString, 734
- empty
 - gcstring, 779
- EnableAll
 - ImageStatistics, 866
 - ImageStatistics, 945
- enableChunkData
 - Compression.cpp, 1423
- EnableGreyOnly
 - ImageStatistics, 866
 - ImageStatistics, 945
- EnableHSLOnly
 - ImageStatistics, 866
 - ImageStatistics, 945
- EnableImageChunkData
 - Compression.cpp, 1422
- EnableImageCompression
 - Compression.cpp, 1422
- EnableManualFramerate
 - GigEVisionPerformance.cpp, 1441
- EnableRGBOnly
 - ImageStatistics, 866
 - ImageStatistics, 946
- ENamespaceClass, 735
 - FromString, 735
 - ToString, 735
- EncoderDivider
 - Camera, 488
- EncoderMode
 - Camera, 488
- EncoderMode_FourPhase
 - Spinnaker, 270
- EncoderMode_HighResolution
 - Spinnaker, 270
- EncoderModeEnums
 - Spinnaker, 270
- EncoderOutputMode
 - Camera, 488
- EncoderOutputMode_DirectionDown
 - Spinnaker, 270
- EncoderOutputMode_DirectionUp
 - Spinnaker, 270
- EncoderOutputMode_Motion
 - Spinnaker, 270
- EncoderOutputMode_Off
 - Spinnaker, 270
- EncoderOutputMode_PositionDown
 - Spinnaker, 270
- EncoderOutputMode_PositionUp

Spinnaker, [270](#)
 EncoderOutputModeEnums
 Spinnaker, [270](#)
 EncoderReset
 Camera, [489](#)
 EncoderResetActivation
 Camera, [489](#)
 EncoderResetActivation_AnyEdge
 Spinnaker, [271](#)
 EncoderResetActivation_FallingEdge
 Spinnaker, [271](#)
 EncoderResetActivation_LevelHigh
 Spinnaker, [271](#)
 EncoderResetActivation_LevelLow
 Spinnaker, [271](#)
 EncoderResetActivation_RisingEdge
 Spinnaker, [271](#)
 EncoderResetActivationEnums
 Spinnaker, [271](#)
 EncoderResetSource
 Camera, [489](#)
 EncoderResetSource_AcquisitionEnd
 Spinnaker, [271](#)
 EncoderResetSource_AcquisitionStart
 Spinnaker, [271](#)
 EncoderResetSource_AcquisitionTrigger
 Spinnaker, [271](#)
 EncoderResetSource_Action0
 Spinnaker, [272](#)
 EncoderResetSource_Action1
 Spinnaker, [272](#)
 EncoderResetSource_Action2
 Spinnaker, [272](#)
 EncoderResetSource_Counter0End
 Spinnaker, [271](#)
 EncoderResetSource_Counter0Start
 Spinnaker, [271](#)
 EncoderResetSource_Counter1End
 Spinnaker, [271](#)
 EncoderResetSource_Counter1Start
 Spinnaker, [271](#)
 EncoderResetSource_Counter2End
 Spinnaker, [271](#)
 EncoderResetSource_Counter2Start
 Spinnaker, [271](#)
 EncoderResetSource_ExposureEnd
 Spinnaker, [271](#)
 EncoderResetSource_ExposureStart
 Spinnaker, [271](#)
 EncoderResetSource_FrameEnd
 Spinnaker, [271](#)
 EncoderResetSource_FrameStart
 Spinnaker, [271](#)
 EncoderResetSource_FrameTrigger
 Spinnaker, [271](#)
 EncoderResetSource_Line0
 Spinnaker, [271](#)
 EncoderResetSource_Line1

Spinnaker, [271](#)
 EncoderResetSource_Line2
 Spinnaker, [271](#)
 EncoderResetSource_LinkTrigger0
 Spinnaker, [272](#)
 EncoderResetSource_LinkTrigger1
 Spinnaker, [272](#)
 EncoderResetSource_LinkTrigger2
 Spinnaker, [272](#)
 EncoderResetSource_Off
 Spinnaker, [271](#)
 EncoderResetSource_SoftwareSignal0
 Spinnaker, [272](#)
 EncoderResetSource_SoftwareSignal1
 Spinnaker, [272](#)
 EncoderResetSource_SoftwareSignal2
 Spinnaker, [272](#)
 EncoderResetSource_Timer0End
 Spinnaker, [272](#)
 EncoderResetSource_Timer0Start
 Spinnaker, [271](#)
 EncoderResetSource_Timer1End
 Spinnaker, [272](#)
 EncoderResetSource_Timer1Start
 Spinnaker, [271](#)
 EncoderResetSource_Timer2End
 Spinnaker, [272](#)
 EncoderResetSource_Timer2Start
 Spinnaker, [271](#)
 EncoderResetSource_UserOutput0
 Spinnaker, [272](#)
 EncoderResetSource_UserOutput1
 Spinnaker, [272](#)
 EncoderResetSource_UserOutput2
 Spinnaker, [272](#)
 EncoderResetSourceEnums
 Spinnaker, [271](#)
 EncoderSelector
 Camera, [489](#)
 EncoderSelector_Encoder0
 Spinnaker, [272](#)
 EncoderSelector_Encoder1
 Spinnaker, [272](#)
 EncoderSelector_Encoder2
 Spinnaker, [272](#)
 EncoderSelectorEnums
 Spinnaker, [272](#)
 EncoderSourceA
 Camera, [489](#)
 EncoderSourceA_Line0
 Spinnaker, [273](#)
 EncoderSourceA_Line1
 Spinnaker, [273](#)
 EncoderSourceA_Line2
 Spinnaker, [273](#)
 EncoderSourceA_Off
 Spinnaker, [273](#)
 EncoderSourceAEnums

- Spinnaker, [272](#)
- EncoderSourceB
 - Camera, [489](#)
- EncoderSourceB_Line0
 - Spinnaker, [273](#)
- EncoderSourceB_Line1
 - Spinnaker, [273](#)
- EncoderSourceB_Line2
 - Spinnaker, [273](#)
- EncoderSourceB_Off
 - Spinnaker, [273](#)
- EncoderSourceBEnums
 - Spinnaker, [273](#)
- EncoderStatus
 - Camera, [490](#)
- EncoderStatus_EncoderDown
 - Spinnaker, [273](#)
- EncoderStatus_EncoderIdle
 - Spinnaker, [273](#)
- EncoderStatus_EncoderStatic
 - Spinnaker, [273](#)
- EncoderStatus_EncoderUp
 - Spinnaker, [273](#)
- EncoderStatusEnums
 - Spinnaker, [273](#)
- EncoderTimeout
 - Camera, [490](#)
- EncoderValue
 - Camera, [490](#)
- EncoderValueAtReset
 - Camera, [490](#)
- EndAcquisition
 - CameraBase, [569](#)
 - ICameraBase, [801](#)
 - SpinnakerDirectShow.h, [1338](#)
- endTime
 - SecondsCounter, [192](#)
- entryExist
 - SpinnakerDirectShow.h, [1351](#)
- entryIndex
 - SpinnakerDirectShow.h, [1351](#)
- EnumClasses Class, [114](#)
- EnumEntryNode, [736](#)
 - ~EnumEntryNode, [737](#)
 - EnumEntryNode, [737](#)
 - GetNumericValue, [737](#)
 - GetSymbolic, [738](#)
 - GetValue, [738](#)
 - IsSelfClearing, [738](#)
 - SetReference, [738](#)
- EnumEntryNode Class, [115](#)
- EnumerateGen2Cameras
 - TransportLayerSystem, [1152](#)
- EnumerateGEVInterfaces
 - TransportLayerSystem, [1153](#)
- EnumerateUSBInterfaces
 - TransportLayerSystem, [1153](#)
- Enumeration.cpp
 - main, [1428](#)
 - QueryInterface, [1428](#)
- Enumeration_QuickSpin.cpp
 - main, [1429](#)
 - QueryInterface, [1429](#)
- EnumerationCount
 - Camera, [490](#)
- enumerationEntryName
 - SpinnakerDirectShow.h, [1351](#)
- EnumerationEvents.cpp
 - CheckGevEnabled, [1430](#)
 - main, [1430](#)
- EnumerationGetEntry
 - SpinnakerDirectShow.h, [1339](#)
- EnumerationGetNumEntries
 - SpinnakerDirectShow.h, [1339](#)
- enumerationName
 - SpinnakerDirectShow.h, [1351](#)
- EnumerationSetEntry
 - SpinnakerDirectShow.h, [1340](#)
- EnumNode, [739](#)
 - ~EnumNode, [741](#)
 - EnumNode, [741](#)
 - GetCurrentEntry, [741](#)
 - GetEntries, [741](#)
 - GetEntry, [742](#)
 - GetEntryByName, [742](#)
 - GetIntValue, [742](#)
 - GetSymbolics, [742](#)
 - m_pEnumeration, [744](#)
 - operator*, [743](#)
 - operator=, [743](#)
 - SetIntValue, [743](#)
 - SetReference, [743](#)
- EnumNode Class, [116](#)
- EnumNodeT Class, [117](#)
- ERepresentationClass, [744](#)
 - FromString, [744](#)
 - ToString, [745](#)
- Error
 - Spinnaker, [273](#)
- ESignClass, [745](#)
 - FromString, [746](#)
 - ToString, [746](#)
- ESlopeClass, [746](#)
 - FromString, [747](#)
 - ToString, [747](#)
- EStandardNameSpaceClass, [747](#)
 - FromString, [748](#)
 - ToString, [748](#)
- Event
 - GVCP_EVENTDATA_REQUEST, [793](#)
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [794](#)
- EVENT_TIMEOUT_INFINITE
 - Spinnaker, [338](#)
- EVENT_TIMEOUT_NONE
 - Spinnaker, [338](#)

- EventAcquisitionEnd
 - Camera, [490](#)
- EventAcquisitionEndFrameID
 - Camera, [491](#)
- EventAcquisitionEndTimestamp
 - Camera, [491](#)
- EventAcquisitionError
 - Camera, [491](#)
- EventAcquisitionErrorFrameID
 - Camera, [491](#)
- EventAcquisitionErrorTimestamp
 - Camera, [491](#)
- EventAcquisitionStart
 - Camera, [491](#)
- EventAcquisitionStartFrameID
 - Camera, [492](#)
- EventAcquisitionStartTimestamp
 - Camera, [492](#)
- EventAcquisitionTransferEnd
 - Camera, [492](#)
- EventAcquisitionTransferEndFrameID
 - Camera, [492](#)
- EventAcquisitionTransferEndTimestamp
 - Camera, [492](#)
- EventAcquisitionTransferStart
 - Camera, [492](#)
- EventAcquisitionTransferStartFrameID
 - Camera, [493](#)
- EventAcquisitionTransferStartTimestamp
 - Camera, [493](#)
- EventAcquisitionTrigger
 - Camera, [493](#)
- EventAcquisitionTriggerFrameID
 - Camera, [493](#)
- EventAcquisitionTriggerTimestamp
 - Camera, [493](#)
- EventActionLate
 - Camera, [493](#)
- EventActionLateFrameID
 - Camera, [494](#)
- EventActionLateTimestamp
 - Camera, [494](#)
- EventAdapter Class, [118](#)
- EventAdapter1394 Class, [119](#)
- EventAdapterGeneric Class, [120](#)
- EventAdapterGEV Class, [121](#)
- EventAdapterU3V Class, [122](#)
- EventCallbackInfo
 - ImageListEventHandler, [860](#)
- EventCounter0End
 - Camera, [494](#)
- EventCounter0EndFrameID
 - Camera, [494](#)
- EventCounter0EndTimestamp
 - Camera, [494](#)
- EventCounter0Start
 - Camera, [494](#)
- EventCounter0StartFrameID
 - Camera, [495](#)
- EventCounter0StartTimestamp
 - Camera, [495](#)
- EventCounter1End
 - Camera, [495](#)
- EventCounter1EndFrameID
 - Camera, [495](#)
- EventCounter1EndTimestamp
 - Camera, [495](#)
- EventCounter1Start
 - Camera, [495](#)
- EventCounter1StartFrameID
 - Camera, [496](#)
- EventCounter1StartTimestamp
 - Camera, [496](#)
- EventData
 - U3V_EVENT_MESSAGE, [1160](#)
- EventEncoder0Restarted
 - Camera, [496](#)
- EventEncoder0RestartedFrameID
 - Camera, [496](#)
- EventEncoder0RestartedTimestamp
 - Camera, [496](#)
- EventEncoder0Stopped
 - Camera, [496](#)
- EventEncoder0StoppedFrameID
 - Camera, [497](#)
- EventEncoder0StoppedTimestamp
 - Camera, [497](#)
- EventEncoder1Restarted
 - Camera, [497](#)
- EventEncoder1RestartedFrameID
 - Camera, [497](#)
- EventEncoder1RestartedTimestamp
 - Camera, [497](#)
- EventEncoder1Stopped
 - Camera, [497](#)
- EventEncoder1StoppedFrameID
 - Camera, [498](#)
- EventEncoder1StoppedTimestamp
 - Camera, [498](#)
- EventError
 - Camera, [498](#)
- EventErrorCode
 - Camera, [498](#)
- EventErrorFrameID
 - Camera, [498](#)
- EventErrorTimestamp
 - Camera, [498](#)
- EventExposureEnd
 - Camera, [499](#)
- EventExposureEndFrameID
 - Camera, [499](#)
- EventExposureEndTimestamp
 - Camera, [499](#)
- EventExposureStart
 - Camera, [499](#)
- EventExposureStartFrameID
 - Camera, [499](#)

- Camera, [499](#)
- EventExposureStartTimestamp
 - Camera, [499](#)
- EventFrameBurstEnd
 - Camera, [500](#)
- EventFrameBurstEndFrameID
 - Camera, [500](#)
- EventFrameBurstEndTimestamp
 - Camera, [500](#)
- EventFrameBurstStart
 - Camera, [500](#)
- EventFrameBurstStartFrameID
 - Camera, [500](#)
- EventFrameBurstStartTimestamp
 - Camera, [500](#)
- EventFrameEnd
 - Camera, [501](#)
- EventFrameEndFrameID
 - Camera, [501](#)
- EventFrameEndTimestamp
 - Camera, [501](#)
- EventFrameStart
 - Camera, [501](#)
- EventFrameStartFrameID
 - Camera, [501](#)
- EventFrameStartTimestamp
 - Camera, [501](#)
- EventFrameTransferEnd
 - Camera, [502](#)
- EventFrameTransferEndFrameID
 - Camera, [502](#)
- EventFrameTransferEndTimestamp
 - Camera, [502](#)
- EventFrameTransferStart
 - Camera, [502](#)
- EventFrameTransferStartFrameID
 - Camera, [502](#)
- EventFrameTransferStartTimestamp
 - Camera, [502](#)
- EventFrameTrigger
 - Camera, [503](#)
- EventFrameTriggerFrameID
 - Camera, [503](#)
- EventFrameTriggerTimestamp
 - Camera, [503](#)
- EventHandler, [749](#)
 - ~EventHandler, [750](#)
 - CallbackHandlerInternal, [751](#)
 - EventHandler, [750](#)
 - EventProcessor, [751](#)
 - GetEventPayloadData, [750](#)
 - GetEventPayloadDataSize, [750](#)
 - GetEventType, [750](#)
 - IDataStream, [752](#)
 - ImageSetInspector, [752](#)
 - m_pEventData, [752](#)
 - operator=, [751](#)
 - SetEventPayload, [751](#)
 - SetEventType, [751](#)
 - Stream, [752](#)
- EventHandler Class, [55](#)
- EventId
 - GVCP_EVENT_ITEM, [787](#)
 - GVCP_EVENT_ITEM_BASIC, [788](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
 - U3V_EVENT_DATA, [1159](#)
- EventLine0AnyEdge
 - Camera, [503](#)
- EventLine0AnyEdgeFrameID
 - Camera, [503](#)
- EventLine0AnyEdgeTimestamp
 - Camera, [503](#)
- EventLine0FallingEdge
 - Camera, [504](#)
- EventLine0FallingEdgeFrameID
 - Camera, [504](#)
- EventLine0FallingEdgeTimestamp
 - Camera, [504](#)
- EventLine0RisingEdge
 - Camera, [504](#)
- EventLine0RisingEdgeFrameID
 - Camera, [504](#)
- EventLine0RisingEdgeTimestamp
 - Camera, [504](#)
- EventLine1AnyEdge
 - Camera, [505](#)
- EventLine1AnyEdgeFrameID
 - Camera, [505](#)
- EventLine1AnyEdgeTimestamp
 - Camera, [505](#)
- EventLine1FallingEdge
 - Camera, [505](#)
- EventLine1FallingEdgeFrameID
 - Camera, [505](#)
- EventLine1FallingEdgeTimestamp
 - Camera, [505](#)
- EventLine1RisingEdge
 - Camera, [506](#)
- EventLine1RisingEdgeFrameID
 - Camera, [506](#)
- EventLine1RisingEdgeTimestamp
 - Camera, [506](#)
- EventLinkSpeedChange
 - Camera, [506](#)
- EventLinkSpeedChangeFrameID
 - Camera, [506](#)
- EventLinkSpeedChangeTimestamp
 - Camera, [506](#)
- EventLinkTrigger0
 - Camera, [507](#)
- EventLinkTrigger0FrameID
 - Camera, [507](#)
- EventLinkTrigger0Timestamp
 - Camera, [507](#)
- EventLinkTrigger1
 - Camera, [507](#)

- EventLinkTrigger1FrameID
 - Camera, [507](#)
- EventLinkTrigger1Timestamp
 - Camera, [507](#)
- EventNotification
 - Camera, [508](#)
- EventNotification_Off
 - Spinnaker, [275](#)
- EventNotification_On
 - Spinnaker, [275](#)
- EventNotificationEnums
 - Spinnaker, [274](#)
- EventPort Class, [123](#)
- EventProcessor
 - EventHandler, [751](#)
- EventSelector
 - Camera, [508](#)
- EventSelector_Error
 - Spinnaker, [275](#)
- EventSelector_ExposureEnd
 - Spinnaker, [275](#)
- EventSelector_SerialPortReceive
 - Spinnaker, [275](#)
- EventSelectorEnums
 - Spinnaker, [275](#)
- EventSequencerSetChange
 - Camera, [508](#)
- EventSequencerSetChangeFrameID
 - Camera, [508](#)
- EventSequencerSetChangeTimestamp
 - Camera, [508](#)
- EventSerialData
 - Camera, [508](#)
- EventSerialDataLength
 - Camera, [509](#)
- EventSerialPortReceive
 - Camera, [509](#)
- EventSerialPortReceiveTimestamp
 - Camera, [509](#)
- EventSerialReceiveOverflow
 - Camera, [509](#)
- EventStream0TransferBlockEnd
 - Camera, [509](#)
- EventStream0TransferBlockEndFrameID
 - Camera, [509](#)
- EventStream0TransferBlockEndTimestamp
 - Camera, [510](#)
- EventStream0TransferBlockStart
 - Camera, [510](#)
- EventStream0TransferBlockStartFrameID
 - Camera, [510](#)
- EventStream0TransferBlockStartTimestamp
 - Camera, [510](#)
- EventStream0TransferBlockTrigger
 - Camera, [510](#)
- EventStream0TransferBlockTriggerFrameID
 - Camera, [510](#)
- EventStream0TransferBlockTriggerTimestamp
 - Camera, [511](#)
- EventStream0TransferBurstEnd
 - Camera, [511](#)
- EventStream0TransferBurstEndFrameID
 - Camera, [511](#)
- EventStream0TransferBurstEndTimestamp
 - Camera, [511](#)
- EventStream0TransferBurstStart
 - Camera, [511](#)
- EventStream0TransferBurstStartFrameID
 - Camera, [511](#)
- EventStream0TransferBurstStartTimestamp
 - Camera, [512](#)
- EventStream0TransferEnd
 - Camera, [512](#)
- EventStream0TransferEndFrameID
 - Camera, [512](#)
- EventStream0TransferEndTimestamp
 - Camera, [512](#)
- EventStream0TransferOverflow
 - Camera, [512](#)
- EventStream0TransferOverflowFrameID
 - Camera, [512](#)
- EventStream0TransferOverflowTimestamp
 - Camera, [513](#)
- EventStream0TransferPause
 - Camera, [513](#)
- EventStream0TransferPauseFrameID
 - Camera, [513](#)
- EventStream0TransferPauseTimestamp
 - Camera, [513](#)
- EventStream0TransferResume
 - Camera, [513](#)
- EventStream0TransferResumeFrameID
 - Camera, [513](#)
- EventStream0TransferResumeTimestamp
 - Camera, [514](#)
- EventStream0TransferStart
 - Camera, [514](#)
- EventStream0TransferStartFrameID
 - Camera, [514](#)
- EventStream0TransferStartTimestamp
 - Camera, [514](#)
- EventTest
 - Camera, [514](#)
- EventTestTimestamp
 - Camera, [514](#)
- EventTimer0End
 - Camera, [515](#)
- EventTimer0EndFrameID
 - Camera, [515](#)
- EventTimer0EndTimestamp
 - Camera, [515](#)
- EventTimer0Start
 - Camera, [515](#)
- EventTimer0StartFrameID
 - Camera, [515](#)
- EventTimer0StartTimestamp
 - Camera, [515](#)

- Camera, [515](#)
- EventTimer1End
 - Camera, [516](#)
- EventTimer1EndFrameID
 - Camera, [516](#)
- EventTimer1EndTimestamp
 - Camera, [516](#)
- EventTimer1Start
 - Camera, [516](#)
- EventTimer1StartFrameID
 - Camera, [516](#)
- EventTimer1StartTimestamp
 - Camera, [516](#)
- EventType
 - Spinnaker, [275](#)
- eventType
 - DeviceEvents.cpp, [1426](#)
- EVisibilityClass, [752](#)
 - FromString, [753](#)
 - ToString, [753](#)
- Exception, [754](#)
 - ~Exception, [756](#)
 - Exception, [755](#), [756](#)
 - GetBuildDate, [757](#)
 - GetBuildTime, [757](#)
 - GetError, [757](#)
 - GetErrorMessage, [757](#)
 - GetFileName, [757](#)
 - GetFullErrorMessage, [757](#)
 - GetFunctionName, [757](#)
 - GetLineNumber, [758](#)
 - operator!=, [758](#)
 - operator=, [758](#)
 - operator==, [758](#)
 - what, [758](#)
- Exception Class, [56](#)
- ExceptionHandling.cpp
 - causeSpinnakerException, [1431](#)
 - causeStandardException, [1431](#)
 - chosenException, [1432](#)
 - exceptionType, [1431](#)
 - main, [1431](#)
 - SPINNAKER_EXCEPTION, [1431](#)
 - STANDARD_CAST_TO_SPINNAKER, [1431](#)
 - STANDARD_EXCEPTION, [1431](#)
- exceptionType
 - ExceptionHandling.cpp, [1431](#)
- Execute
 - CommandNode, [685](#)
- ExecuteDeleteCommand
 - FileAccess_QuickSpin.cpp, [1436](#)
- ExecuteReadCommand
 - FileAccess_QuickSpin.cpp, [1436](#)
- ExecuteWriteCommand
 - FileAccess_QuickSpin.cpp, [1436](#)
- EXPAND_TO_STRINGISE
 - GCUilities.h, [1273](#)
- Expert
 - Spinnaker::GenApi, [361](#)
- Exposure.cpp
 - AcquireImages, [1432](#)
 - ConfigureExposure, [1432](#)
 - main, [1432](#)
 - PrintDeviceInfo, [1433](#)
 - ResetExposure, [1433](#)
 - RunSingleCamera, [1433](#)
- Exposure_QuickSpin.cpp
 - AcquireImages, [1434](#)
 - ConfigureExposure, [1434](#)
 - main, [1434](#)
 - PrintDeviceInfo, [1434](#)
 - ResetExposure, [1434](#)
 - RunSingleCamera, [1434](#)
- ExposureActiveMode
 - Camera, [517](#)
- ExposureActiveMode_AllPixels
 - Spinnaker, [276](#)
- ExposureActiveMode_AnyPixels
 - Spinnaker, [276](#)
- ExposureActiveMode_Line1
 - Spinnaker, [276](#)
- ExposureActiveModeEnums
 - Spinnaker, [276](#)
- ExposureAuto
 - Camera, [517](#)
- ExposureAuto_Continuous
 - Spinnaker, [276](#)
- ExposureAuto_Off
 - Spinnaker, [276](#)
- ExposureAuto_Once
 - Spinnaker, [276](#)
- ExposureAutoEnums
 - Spinnaker, [276](#)
- ExposureMode
 - Camera, [517](#)
- ExposureMode_Timed
 - Spinnaker, [276](#)
- ExposureMode_TriggerWidth
 - Spinnaker, [276](#)
- ExposureModeEnums
 - Spinnaker, [276](#)
- ExposureTime
 - Camera, [517](#)
- ExposureTimeMode
 - Camera, [517](#)
- ExposureTimeMode_Common
 - Spinnaker, [277](#)
- ExposureTimeMode_Individual
 - Spinnaker, [277](#)
- ExposureTimeModeEnums
 - Spinnaker, [277](#)
- ExposureTimeSelector
 - Camera, [518](#)
- ExposureTimeSelector_Blue
 - Spinnaker, [277](#)
- ExposureTimeSelector_Common

- Spinnaker, [277](#)
- ExposureTimeSelector_Cyan
 - Spinnaker, [277](#)
- ExposureTimeSelector_Green
 - Spinnaker, [277](#)
- ExposureTimeSelector_Infrared
 - Spinnaker, [277](#)
- ExposureTimeSelector_Magenta
 - Spinnaker, [277](#)
- ExposureTimeSelector_Red
 - Spinnaker, [277](#)
- ExposureTimeSelector_Stage1
 - Spinnaker, [277](#)
- ExposureTimeSelector_Stage2
 - Spinnaker, [277](#)
- ExposureTimeSelector_Ultraviolet
 - Spinnaker, [277](#)
- ExposureTimeSelector_Yellow
 - Spinnaker, [277](#)
- ExposureTimeSelectorEnums
 - Spinnaker, [277](#)
- ExtractAndSavePolarQuadImages
 - Polarization.cpp, [1468](#)
- ExtractIndependentSubtree
 - Spinnaker::GenApi, [367](#)
- ExtractPolarQuadrant
 - ImageUtilityPolarization, [967](#)
- ExtractSubtree
 - CNodeMapFactory, [671](#)
- EYesNoClass, [759](#)
 - FromString, [759](#)
 - ToString, [759](#)
- FactoryReset
 - Camera, [518](#)
- FileAccess_QuickSpin.cpp
 - _enableDebug, [1438](#)
 - _fileSelector, [1438](#)
 - AcquireImages, [1435](#)
 - CloseFile, [1435](#)
 - DownloadImage, [1436](#)
 - ExecuteDeleteCommand, [1436](#)
 - ExecuteReadCommand, [1436](#)
 - ExecuteWriteCommand, [1436](#)
 - InitializeSystem, [1436](#)
 - main, [1436](#)
 - OpenFileToRead, [1436](#)
 - OpenFileToWrite, [1437](#)
 - PrintDebugMessage, [1437](#)
 - PrintDeviceInfo, [1437](#)
 - PrintResultMessage, [1437](#)
 - PrintUsage, [1437](#)
 - UploadImage, [1437](#)
- FileAccessBuffer
 - Camera, [518](#)
- FileAccessLength
 - Camera, [518](#)
- FileAccessOffset
 - Camera, [518](#)
- filebuf_type
 - IDevFileStreamBase< CharType, Traits >, [829](#)
 - ODevFileStreamBase< CharType, Traits >, [1060](#)
- fileName
 - CompressedImageInfo, [687](#)
- FileOpenMode
 - Camera, [518](#)
- FileOpenMode_Read
 - Spinnaker, [279](#)
- FileOpenMode_ReadWrite
 - Spinnaker, [279](#)
- FileOpenMode_Write
 - Spinnaker, [279](#)
- FileOpenModeEnums
 - Spinnaker, [277](#)
- FileOperationExecute
 - Camera, [519](#)
- FileOperationResult
 - Camera, [519](#)
- FileOperationSelector
 - Camera, [519](#)
- FileOperationSelector_Close
 - Spinnaker, [279](#)
- FileOperationSelector_Delete
 - Spinnaker, [279](#)
- FileOperationSelector_Open
 - Spinnaker, [279](#)
- FileOperationSelector_Read
 - Spinnaker, [279](#)
- FileOperationSelector_Write
 - Spinnaker, [279](#)
- FileOperationSelectorEnums
 - Spinnaker, [279](#)
- FileOperationStatus
 - Camera, [519](#)
- FileOperationStatus_Failure
 - Spinnaker, [279](#)
- FileOperationStatus_Overflow
 - Spinnaker, [279](#)
- FileOperationStatus_Success
 - Spinnaker, [279](#)
- FileOperationStatusEnums
 - Spinnaker, [279](#)
- FileProtocolAdapter, [760](#)
 - ~FileProtocolAdapter, [760](#)
 - attach, [761](#)
 - closeFile, [762](#)
 - deleteFile, [762](#)
 - FileProtocolAdapter, [760](#)
 - getBufSize, [762](#)
 - openFile, [763](#)
 - read, [763](#)
 - write, [764](#)
- FileSelector
 - Camera, [520](#)
- FileSelector_SerialPort0
 - Spinnaker, [280](#)
- FileSelector_UserFile1

- Spinnaker, [280](#)
- FileSelector_UserSet0
 - Spinnaker, [280](#)
- FileSelector_UserSet1
 - Spinnaker, [280](#)
- FileSelector_UserSetDefault
 - Spinnaker, [280](#)
- FileSelectorEnums
 - Spinnaker, [279](#)
- FileSize
 - Camera, [520](#)
- Filestream Class, [124](#)
- FileUploadPersistence
 - Inference.cpp, [1451](#)
- FilterDriverStatus
 - TransportLayerInterface, [1135](#)
- FilterDriverStatus_Disabled
 - Spinnaker, [280](#)
- FilterDriverStatus_Enabled
 - Spinnaker, [280](#)
- FilterDriverStatus_NotSupported
 - Spinnaker, [280](#)
- FilterDriverStatusEnum
 - Spinnaker, [280](#)
- find
 - gcstring, [779](#), [780](#)
- find_first_not_of
 - gcstring, [780](#)
- find_first_of
 - gcstring, [780](#)
- fixedIncrement
 - Spinnaker::GenApi, [358](#)
- Flags
 - GVCP_REQUEST_HEADER, [795](#)
 - U3V_COMMAND_HEADER, [1158](#)
- FLASH
 - Inference.cpp, [1451](#)
- FLIR_SPINNAKER_VERSION_BUILD
 - System.h, [1358](#)
- FLIR_SPINNAKER_VERSION_MAJOR
 - System.h, [1358](#)
- FLIR_SPINNAKER_VERSION_MINOR
 - System.h, [1359](#)
- FLIR_SPINNAKER_VERSION_TYPE
 - System.h, [1359](#)
- float32_t
 - GTypes Class, [129](#)
- float64_t
 - GTypes Class, [129](#)
- FloatGetInc
 - SpinnakerDirectShow.h, [1340](#)
- FloatGetIncMode
 - SpinnakerDirectShow.h, [1340](#)
- FloatGetMax
 - SpinnakerDirectShow.h, [1341](#)
- FloatGetMin
 - SpinnakerDirectShow.h, [1341](#)
- FloatGetValue
 - SpinnakerDirectShow.h, [1341](#)
- FloatNode, [764](#)
 - ~FloatNode, [767](#)
- FloatNode, [767](#)
 - GetDisplayNotation, [767](#)
 - GetDisplayPrecision, [767](#)
 - GetEnumAlias, [767](#)
 - GetInc, [767](#)
 - GetIncMode, [768](#)
 - GetIntAlias, [768](#)
 - GetListOfValidValues, [768](#)
 - GetMax, [768](#)
 - GetMin, [768](#)
 - GetRepresentation, [768](#)
 - GetUnit, [769](#)
 - GetValue, [769](#)
 - HasInc, [769](#)
 - ImposeMax, [769](#)
 - ImposeMin, [769](#)
 - operator*, [770](#)
 - operator(), [770](#)
 - operator=, [770](#)
 - SetReference, [770](#)
 - SetValue, [770](#)
- FloatNode Class, [125](#)
- FloatRegNode, [771](#)
 - ~FloatRegNode, [773](#)
- FloatRegNode, [772](#), [773](#)
 - SetReference, [773](#)
- FloatRegNode Class, [126](#)
- FloatSetValue
 - SpinnakerDirectShow.h, [1342](#)
- FlushQueueAllDiscard
 - IDataStream, [823](#)
- FMT_I64
 - Compatibility.h, [1251](#)
- fnAutomatic
 - Spinnaker::GenApi, [357](#)
- fnFixed
 - Spinnaker::GenApi, [357](#)
- fnScientific
 - Spinnaker::GenApi, [357](#)
- ForcelP
 - CameraBase, [570](#)
 - ICameraBase, [801](#)
- frameID
 - DeviceEventExposureEndData, [715](#)
 - DeviceEventInferenceData, [721](#)
- frameRate
 - AVIOption, [404](#)
 - H264Option, [797](#)
 - MJPGOption, [1037](#)
- FromString
 - EAccessModeClass, [728](#)
 - ECachingModeClass, [730](#)
 - EDisplayNotationClass, [731](#)
 - EEndianessClass, [732](#)
 - EGenApiSchemaVersionClass, [733](#)

- ElInputDirectionClass, [734](#)
- ENamespaceClass, [735](#)
- ERepresentationClass, [744](#)
- ESignClass, [746](#)
- ESlopeClass, [747](#)
- EStandardNamespaceClass, [748](#)
- EVisibilityClass, [753](#)
- EYesNoClass, [759](#)
- Spinnaker::GenApi, [367](#)
- ValueNode, [1163](#)
- Function_NodeCallback
 - Function_NodeCallback< Function >, [775](#)
- Function_NodeCallback< Function >, [774](#)
 - Destroy, [775](#)
 - Function_NodeCallback, [775](#)
 - operator(), [775](#)
- Gain
 - Camera, [520](#)
- GainAuto
 - Camera, [520](#)
- GainAuto_Continuous
 - Spinnaker, [281](#)
- GainAuto_Off
 - Spinnaker, [281](#)
- GainAuto_Once
 - Spinnaker, [281](#)
- GainAutoBalance
 - Camera, [520](#)
- GainAutoBalance_Continuous
 - Spinnaker, [280](#)
- GainAutoBalance_Off
 - Spinnaker, [280](#)
- GainAutoBalance_Once
 - Spinnaker, [280](#)
- GainAutoBalanceEnums
 - Spinnaker, [280](#)
- GainAutoEnums
 - Spinnaker, [280](#)
- GainSelector
 - Camera, [521](#)
- GainSelector_All
 - Spinnaker, [281](#)
- GainSelectorEnums
 - Spinnaker, [281](#)
- Gamma
 - Camera, [521](#)
- GammaEnable
 - Camera, [521](#)
- GC_COUNTOF
 - GCUtilities.h, [1274](#)
- GC_INT32_MAX
 - GCTypes.h, [1269](#)
- GC_INT32_MIN
 - GCTypes.h, [1269](#)
- GC_INT64_MAX
 - GCTypes.h, [1269](#)
- GC_INT64_MIN
 - GCTypes.h, [1269](#)
- GC_INT8_MAX
 - GCTypes.h, [1270](#)
- GC_INT8_MIN
 - GCTypes.h, [1270](#)
- GC_UINT32_MAX
 - GCTypes.h, [1270](#)
- GC_UINT64_MAX
 - GCTypes.h, [1270](#)
- GC_UINT8_MAX
 - GCTypes.h, [1270](#)
- gcstring, [776](#)
 - _npos, [778](#)
 - ~gcstring, [778](#)
 - append, [778](#)
 - assign, [778](#), [779](#)
 - c_str, [779](#)
 - compare, [779](#)
 - empty, [779](#)
 - find, [779](#), [780](#)
 - find_first_not_of, [780](#)
 - find_first_of, [780](#)
 - gcstring, [777](#)
 - length, [781](#)
 - max_size, [781](#)
 - npos, [785](#)
 - operator const char *, [781](#)
 - operator delete, [781](#)
 - operator new, [781](#)
 - operator!=, [782](#)
 - operator<, [783](#)
 - operator>, [783](#)
 - operator+, [784](#)
 - operator+=", [782](#)
 - operator=, [783](#)
 - operator==, [783](#)
 - resize, [783](#)
 - size, [783](#)
 - substr, [784](#)
 - swap, [784](#)
- GCString Class, [127](#)
- GCString.h
 - GCSTRING_NPOS, [1265](#)
 - operator<<, [1265](#)
 - operator>>, [1266](#)
- GCSTRING_NPOS
 - GCString.h, [1265](#)
- GCSynch Class, [128](#)
- GCTypes Class, [129](#)
 - float32_t, [129](#)
 - float64_t, [129](#)
- GCTypes.h
 - __STDC_CONSTANT_MACROS, [1269](#)
 - __STDC_LIMIT_MACROS, [1269](#)
 - GC_INT32_MAX, [1269](#)
 - GC_INT32_MIN, [1269](#)
 - GC_INT64_MAX, [1269](#)
 - GC_INT64_MIN, [1269](#)
 - GC_INT8_MAX, [1270](#)

- GC_INT8_MIN, [1270](#)
- GC_UINT32_MAX, [1270](#)
- GC_UINT64_MAX, [1270](#)
- GC_UINT8_MAX, [1270](#)
- GCUtilities Utility, [131](#)
- GCUtilities.h
 - _TO_STRING, [1273](#)
 - __ERR__, [1272](#)
 - __LINE_STR__, [1273](#)
 - __LOCATION__, [1273](#)
 - __OUTPUT_FORMATER__, [1273](#)
 - __TODO__, [1273](#)
 - __WARN__, [1273](#)
 - EXPAND_TO_STRINGISE, [1273](#)
 - GC_COUNTOF, [1274](#)
 - GENICAM_DEPRECATED, [1274](#)
 - GENICAM_UNUSED, [1274](#)
 - USE_TEMP_CACHE_FILE, [1274](#)
- GENCP_COMMAND_HEADER_SIZE
 - Spinnaker::GenApi, [388](#)
- GENCP_EVENT_BASIC_SIZE
 - Spinnaker::GenApi, [388](#)
- GENCP_EVENT_CMD_ID
 - Spinnaker::GenApi, [388](#)
- GENERIC
 - DeviceEvents.cpp, [1426](#)
- GENICAM_DEPRECATED
 - GCUtilities.h, [1274](#)
- GENICAM_UNUSED
 - GCUtilities.h, [1274](#)
- GenICamXMLLocation
 - TransportLayerDevice, [1126](#)
- GenICamXMLLocation_Device
 - Spinnaker, [281](#)
- GenICamXMLLocation_Host
 - Spinnaker, [281](#)
- GenICamXMLLocationEnum
 - Spinnaker, [281](#)
- GenICamXMLPath
 - TransportLayerDevice, [1126](#)
- GenTLSFNCVersionMajor
 - TransportLayerSystem, [1153](#)
- GenTLSFNCVersionMinor
 - TransportLayerSystem, [1153](#)
- GenTLSFNCVersionSubMinor
 - TransportLayerSystem, [1153](#)
- GenTLVersionMajor
 - TransportLayerSystem, [1153](#)
- GenTLVersionMinor
 - TransportLayerSystem, [1154](#)
- Get
 - RegisterNode, [1082](#)
 - Spinnaker::GenApi, [367](#)
- get
 - BasePtr< T, B >, [407](#)
- GetAccessMode
 - CameraBase, [570](#)
 - CChunkPort, [608](#)
- CEventPort, [631](#)
- CPortImpl, [696](#)
- CRegisterPortImpl, [703](#)
- CTestPortStruct< CDataStruct >, [709](#)
- ICameraBase, [801](#)
- Node, [1041](#)
- PortRecorder, [1073](#)
- PortReplay, [1077](#)
- GetAddress
 - RegisterNode, [1084](#)
 - Spinnaker::GenApi, [368](#)
- GetAlias
 - Node, [1042](#)
 - Spinnaker::GenApi, [368](#)
- GetBitsPerPixel
 - Image, [842](#)
 - Image, [893](#)
- GetBlackLevel
 - ChunkData, [646](#)
 - IChunkData, [813](#)
- GetBoxAt
 - InferenceBoundingBoxResult, [971](#)
- GetBoxCount
 - InferenceBoundingBoxResult, [971](#)
- GetBoxSize
 - InferenceBoundingBoxResult, [971](#)
- GetBufferChunkData
 - IDataStream, [823](#)
- GetBufferInfoBool8Type
 - IDataStream, [823](#)
- GetBufferInfoPtrType
 - IDataStream, [823](#)
- GetBufferInfoSizeType
 - IDataStream, [823](#)
- GetBufferInfoUInt64Type
 - IDataStream, [824](#)
- GetBufferOwnership
 - CameraBase, [570](#)
 - ICameraBase, [801](#)
- GetBufferSize
 - Image, [842](#)
 - Image, [893](#)
- getBufSize
 - FileProtocolAdapter, [762](#)
- GetBuildDate
 - Exception, [757](#)
- GetBuildTime
 - Exception, [757](#)
- GetByDeviceID
 - CameraList, [584](#)
 - ICameraList, [809](#)
- GetByIndex
 - CameraList, [585](#)
 - ICameraList, [809](#)
 - ImageList, [856](#)
 - InterfaceList, [880](#)
 - ImageList, [923](#)
 - InterfaceList, [1002](#)

- GetByInterfaceID
 - InterfaceList, [1002](#)
- GetByPixelFormat
 - ImageList, [856](#)
 - ImageList, [923](#)
- GetBySerial
 - CameraList, [585](#)
 - ICameraList, [809](#)
- GetCachingMode
 - Node, [1042](#)
 - Spinnaker::GenApi, [368](#)
- GetCallbackType
 - CNodeCallback, [665](#)
- getCameraCategory
 - GigEVisionPerformance.cpp, [1441](#)
- GetCameraInfo
 - SpinnakerDirectShow.h, [1342](#)
- GetCameras
 - IInterface, [870](#)
 - Interface, [985](#)
 - ISystem, [1012](#)
 - System, [1102](#)
- GetCastAlias
 - Node, [1042](#)
 - Spinnaker::GenApi, [368](#)
- GetCategoryName
 - LoggingEventData, [1026](#)
- GetChannelStatus
 - ImageStatistics, [866](#)
 - ImageStatistics, [946](#)
- GetChildren
 - Node, [1042](#)
 - Spinnaker::GenApi, [368](#)
- GetChunkData
 - Image, [842](#)
 - Image, [893](#)
- GetChunkID
 - PortNode, [1068](#)
- GetChunkIDLength
 - CChunkPort, [608](#)
- GetChunkLayoutId
 - Image, [842](#)
 - Image, [894](#)
- GetColorProcessing
 - Image, [842](#)
 - ImageProcessor, [863](#)
 - Image, [894](#)
 - ImageProcessor, [938](#)
- GetCompressionMode
 - ChunkData, [646](#)
 - ICChunkData, [813](#)
- GetCompressionRatio
 - ChunkData, [646](#)
 - ICChunkData, [813](#)
- GetCookie
 - CPortWriteList, [699](#)
 - Spinnaker::GenApi, [369](#)
- GetCounterValue
 - ChunkData, [647](#)
 - ICChunkData, [814](#)
- GetCpuStats
 - CpuUtil, [190](#)
- GetCRC
 - ChunkData, [647](#)
 - ICChunkData, [814](#)
- GetCurrentEntry
 - CEnumerationTRef< EnumT >, [614](#)
 - EnumNode, [741](#)
 - Spinnaker::GenApi, [369](#)
- GetData
 - Image, [843](#)
 - Image, [894](#)
- GetDataAbsoluteMax
 - Image, [843](#)
 - Image, [895](#)
- GetDataAbsoluteMin
 - Image, [843](#)
 - Image, [895](#)
- GetDescription
 - Node, [1043](#)
 - Spinnaker::GenApi, [369](#)
- GetDeviceEventId
 - DeviceEventHandler, [717](#)
 - IDeviceEventHandler, [836](#)
- GetDeviceEventName
 - DeviceEventHandler, [718](#)
 - IDeviceEventHandler, [836](#)
- GetDeviceName
 - Node, [1043](#)
 - NodeMap, [1053](#)
 - Spinnaker::GenApi, [369](#)
- GetDeviceNodeMap
 - IDataStream, [824](#)
- GetDeviceSerial
 - AcquisitionMultipleCameraRecovery.cpp, [1408](#)
- GetDeviceVersion
 - NodeMap, [1053](#)
 - Spinnaker::GenApi, [369](#)
- GetDisplayName
 - Node, [1043](#)
 - Spinnaker::GenApi, [369](#)
- GetDisplayNotation
 - FloatNode, [767](#)
 - Spinnaker::GenApi, [370](#)
- GetDisplayPrecision
 - FloatNode, [767](#)
 - Spinnaker::GenApi, [370](#)
- GetDocuURL
 - Node, [1043](#)
 - Spinnaker::GenApi, [370](#)
- GetEncoderValue
 - ChunkData, [647](#)
 - ICChunkData, [814](#)
- GetEntries
 - EnumNode, [741](#)
 - Spinnaker::GenApi, [370](#)

- GetEntry
 - CEnumerationTRef< EnumT >, 615
 - EnumNode, 742
 - Spinnaker::GenApi, 370
- GetEntryByName
 - EnumNode, 742
 - Spinnaker::GenApi, 371
- GetEnumAlias
 - CFloatPtr, 637
 - FloatNode, 767
- GetError
 - Exception, 757
- GetErrorMessage
 - Exception, 757
 - SpinUpdate.h, 1354
- GetEventID
 - Node, 1043
 - Spinnaker::GenApi, 371
- GetEventIDLength
 - CEventPort, 631
- GetEventPayloadData
 - EventHandler, 750
- GetEventPayloadDataSize
 - EventHandler, 750
- GetEventType
 - EventHandler, 750
- GetExposureEndLineStatusAll
 - ChunkData, 647
 - ICChunkData, 814
- GetExposureTime
 - ChunkData, 648
 - ICChunkData, 814
- GetFeatureBagHandle
 - CFeatureBag, 634
- GetFeatures
 - CategoryNode, 592
- GetFileName
 - Exception, 757
- GetFiles
 - Spinnaker::GenICam, 396
- GetFloatAlias
 - IntegerNode, 980
- GetFrameID
 - ChunkData, 648
 - ICChunkData, 814
 - Image, 843
 - Image, 895
- GetFullErrorMessage
 - Exception, 757
- GetFunctionName
 - Exception, 757
- GetGain
 - ChunkData, 648
 - ICChunkData, 815
- GetGenApiVersion
 - NodeMap, 1053
 - Spinnaker::GenApi, 371
- GetGenICamCacheFolder
 - Spinnaker::GenICam, 396
- GetGenICamCLProtocolFolder
 - Spinnaker::GenICam, 397
- GetGenICamLogConfig
 - Spinnaker::GenICam, 397
- GetGuiXml
 - CameraBase, 571
 - ICameraBase, 801
- GetHeatmapColorGradient
 - ImageUtilityHeatmap, 958
- GetHeatmapRange
 - ImageUtilityHeatmap, 958
- GetHeight
 - ChunkData, 648
 - ICChunkData, 815
 - Image, 843
 - Image, 896
- GetHistogram
 - ImageStatistics, 867
 - ImageStatistics, 946
- GetID
 - Image, 843
 - Image, 896
- GetImage
 - ChunkData, 649
 - ICChunkData, 815
- getImageCount
 - ImageEventHandlerImpl, 917
- GetImageData
 - Image, 844
 - Image, 896
- GetImageSize
 - Image, 844
 - Image, 897
- GetImageStatus
 - Image, 844
 - Image, 897
- GetImageStatusDescription
 - Image, 897
- GetInc
 - FloatNode, 767
 - IntegerNode, 981
 - Spinnaker::GenApi, 371
- GetIncMode
 - FloatNode, 768
 - IntegerNode, 981
 - Spinnaker::GenApi, 371
- GetInferenceBoundingBoxResult
 - ChunkData, 649
 - ICChunkData, 815
- GetInferenceConfidence
 - ChunkData, 649
 - ICChunkData, 815
- GetInferenceFrameId
 - ChunkData, 649
 - ICChunkData, 815
- GetInferenceResult
 - ChunkData, 650

- ICChunkData, [816](#)
- GetInstance
 - System, [1102](#)
- GetIntAlias
 - CFloatPtr, [637](#)
 - FloatNode, [768](#)
- GetInterfaceId
 - InterfaceEventHandlerImpl, [997](#)
- GetInterfaceName
 - Pointer Class, [170](#)
- GetInterfaces
 - ISystem, [1013](#)
 - System, [1103](#)
- GetIntValue
 - EnumNode, [742](#)
 - Spinnaker::GenApi, [371](#)
- GetLength
 - RegisterNode, [1084](#)
 - Spinnaker::GenApi, [372](#)
- GetLibraryVersion
 - ISystem, [1013](#)
 - System, [1103](#)
- getline
 - Spinnaker::GenICam, [397](#)
- GetLineNumber
 - Exception, [758](#)
- GetLinePitch
 - ChunkData, [650](#)
 - ICChunkData, [816](#)
- GetLineStatusAll
 - ChunkData, [650](#)
 - ICChunkData, [816](#)
- GetListOfValidValues
 - FloatNode, [768](#)
 - IntegerNode, [981](#)
 - Spinnaker::GenApi, [372](#)
- GetLock
 - LockableObject< Object >, [1024](#)
 - NodeMap, [1053](#)
 - Spinnaker::GenApi, [372](#)
- GetLoggingEventPriorityLevel
 - ISystem, [1013](#)
 - System, [1104](#)
- GetLogMessage
 - LoggingEventData, [1026](#)
- GetMax
 - FloatNode, [768](#)
 - IntegerNode, [981](#)
 - Spinnaker::GenApi, [372](#)
- getMaxImages
 - ImageEventHandlerImpl, [917](#)
- GetMaxLength
 - Spinnaker::GenApi, [372](#)
 - StringNode, [1095](#)
- GetMean
 - ImageStatistics, [867](#)
 - ImageStatistics, [947](#)
- GetMin
 - FloatNode, [768](#)
 - IntegerNode, [981](#)
 - Spinnaker::GenApi, [373](#)
- GetModelName
 - NodeMap, [1053](#)
- GetModulePathFromFunction
 - Spinnaker::GenICam, [397](#)
- GetName
 - Node, [1043](#)
- GetNameSpace
 - Node, [1043](#)
 - Spinnaker::GenApi, [373](#)
- GetNDC
 - LoggingEventData, [1026](#)
- GetNextImage
 - CameraBase, [571](#)
 - ICameraBase, [802](#)
 - IDataStream, [824](#)
- GetNextImageInternal
 - IDataStream, [824](#)
- GetNextImageSync
 - CameraBase, [572](#)
- GetNode
 - CNodeCallback, [665](#)
 - NodeMap, [1053](#)
 - Spinnaker::GenApi, [373](#)
 - ValueNode, [1163](#)
- GetNodeHandle
 - Node, [1044](#)
- GetNodeMap
 - CameraBase, [572](#)
 - ICameraBase, [802](#)
 - IDataStream, [824](#)
 - Node, [1044](#)
 - Spinnaker::GenApi, [373](#)
- GetNodeMapHandle
 - NodeMap, [1054](#)
- GetNodes
 - NodeMap, [1054](#)
- GetNodeStatistics
 - CNodeMapFactory, [672](#)
- GetNumChannels
 - Image, [844](#)
 - Image, [898](#)
- GetNumDataStreams
 - CameraBase, [573](#)
 - ICameraBase, [802](#)
- GetNumDecompressionThreads
 - ImageProcessor, [863](#)
 - ImageProcessor, [938](#)
- GetNumericValue
 - EnumEntryNode, [737](#)
 - Spinnaker::GenApi, [373](#)
- GetNumImagesInUse
 - CameraBase, [573](#)
 - ICameraBase, [802](#)
 - IDataStream, [824](#)
- GetNumNodes

- NodeMap, [1054](#)
- Spinnaker::GenApi, [373](#)
- GetNumPixelValues
 - ImageStatistics, [867](#)
 - ImageStatistics, [947](#)
- GetNumReads
 - CTestPortStruct< CDataStruct >, [709](#)
- GetNumWrites
 - CTestPortStruct< CDataStruct >, [710](#)
- GetOffsetX
 - ChunkData, [650](#)
 - IChunkData, [816](#)
- GetOffsetY
 - ChunkData, [651](#)
 - IChunkData, [816](#)
- GetParents
 - Node, [1044](#)
 - Spinnaker::GenApi, [373](#)
- GetPartSelector
 - ChunkData, [651](#)
 - IChunkData, [816](#)
- GetPayloadType
 - Image, [844](#)
 - Image, [898](#)
- GetPerformanceCounter
 - PerformanceCounter, [191](#)
- GetPixelDynamicRangeMax
 - ChunkData, [651](#)
 - IChunkData, [817](#)
- GetPixelDynamicRangeMin
 - ChunkData, [651](#)
 - IChunkData, [817](#)
- GetPixelFormat
 - Image, [844](#)
 - Image, [898](#)
- GetPixelFormatIntType
 - Image, [845](#)
 - Image, [899](#)
- GetPixelFormatName
 - Image, [845](#)
 - Image, [899](#)
- GetPixelValueRange
 - ImageStatistics, [867](#)
 - ImageStatistics, [947](#)
- GetPollingTime
 - Node, [1044](#)
 - Spinnaker::GenApi, [374](#)
- GetPort
 - IDataStream, [824](#)
- GetPortHandle
 - PortNode, [1069](#)
- GetPortReplayHandle
 - PortReplay, [1078](#)
- GetPortWriteListHandle
 - CPortWriteList, [700](#)
- GetPrincipalInterfaceType
 - CChunkPort, [608](#)
 - CEventPort, [631](#)
 - CTestPortStruct< CDataStruct >, [710](#)
 - Node, [1044](#)
 - Spinnaker::GenApi, [374](#)
- GetPriority
 - LoggingEventData, [1027](#)
- GetPriorityName
 - LoggingEventData, [1027](#)
- GetPrivateData
 - Image, [845](#)
 - Image, [899](#)
- GetProductGuid
 - NodeMap, [1054](#)
 - Spinnaker::GenApi, [374](#)
- GetProperty
 - Node, [1045](#)
 - Spinnaker::GenApi, [374](#)
- GetPropertyNames
 - Node, [1045](#)
 - Spinnaker::GenApi, [374](#)
- GetQuadFileNameAppendage
 - Polarization.cpp, [1469](#)
- GetRange
 - ImageStatistics, [867](#)
 - ImageStatistics, [948](#)
- GetRepresentation
 - FloatNode, [768](#)
 - IntegerNode, [981](#)
 - Spinnaker::GenApi, [374](#)
- GetScan3dAxisMax
 - ChunkData, [652](#)
 - IChunkData, [817](#)
- GetScan3dAxisMin
 - ChunkData, [652](#)
 - IChunkData, [817](#)
- GetScan3dCoordinateOffset
 - ChunkData, [652](#)
 - IChunkData, [817](#)
- GetScan3dCoordinateReferenceValue
 - ChunkData, [652](#)
 - IChunkData, [817](#)
- GetScan3dCoordinateScale
 - ChunkData, [653](#)
 - IChunkData, [818](#)
- GetScan3dInvalidDataValue
 - ChunkData, [653](#)
 - IChunkData, [818](#)
- GetScan3dTransformValue
 - ChunkData, [653](#)
 - IChunkData, [818](#)
- GetScanLineSelector
 - ChunkData, [653](#)
 - IChunkData, [818](#)
- GetSchemaVersion
 - NodeMap, [1054](#)
 - Spinnaker::GenApi, [375](#)
- GetSecondsCounter
 - SecondsCounter, [192](#)
- GetSelectedCameraIndex

- SpinnakerDirectShow.h, [1343](#)
- GetSelectedFeatures
 - Node, [1045](#)
 - Spinnaker::GenApi, [375](#)
- GetSelectingFeatures
 - Node, [1045](#)
 - Spinnaker::GenApi, [375](#)
- GetSelectorList
 - CSelectorSet, [706](#)
 - Spinnaker::GenApi, [375](#)
- GetSequencerSetActive
 - ChunkData, [654](#)
 - ICChunkData, [818](#)
- GetSerialDataLength
 - ChunkData, [654](#)
 - ICChunkData, [818](#)
- GetSize
 - CameraList, [586](#)
 - ICameraList, [809](#)
 - ImageList, [856](#)
 - IInterfaceList, [880](#)
 - ImageList, [923](#)
 - InterfaceList, [1002](#)
- GetStandardNameSpace
 - NodeMap, [1054](#)
 - Spinnaker::GenApi, [375](#)
- GetStatistics
 - ImageStatistics, [868](#)
 - ImageStatistics, [948](#)
- GetStreamChannelID
 - ChunkData, [654](#)
 - ICChunkData, [819](#)
- GetStreamID
 - IDataStream, [825](#)
- GetStreamIndex
 - IDataStream, [825](#)
 - Image, [845](#)
 - Image, [900](#)
- GetStreamInfoBool8Type
 - IDataStream, [825](#)
- GetStreamInfoSizeType
 - IDataStream, [825](#)
- GetStreamType
 - IDataStream, [825](#)
- GetStride
 - Image, [845](#)
 - Image, [900](#)
- GetSupportedSchemaVersions
 - CNodeMapFactory, [672](#)
 - NodeMap, [1055](#)
 - Spinnaker::GenApi, [376](#)
- GetSwapEndianness
 - CChunkPort, [608](#)
 - CEventPort, [631](#)
 - CPortImpl, [696](#)
 - PortNode, [1069](#)
 - Spinnaker::GenApi, [376](#)
- GetSymbolic
 - EnumEntryNode, [738](#)
 - Spinnaker::GenApi, [376](#)
- GetSymbolics
 - EnumNode, [742](#)
- GetSystemBuffer
 - IDataStream, [825](#)
- GetThreadName
 - LoggingEventData, [1027](#)
- GetTimerValue
 - ChunkData, [654](#)
 - ICChunkData, [819](#)
- GetTimeStamp
 - Image, [845](#)
 - Image, [900](#)
- GetTimestamp
 - ChunkData, [655](#)
 - ICChunkData, [819](#)
 - LoggingEventData, [1027](#)
- GetTimestampLatchValue
 - ChunkData, [655](#)
 - ICChunkData, [819](#)
- GetTLDeviceNodeMap
 - CameraBase, [573](#)
 - ICameraBase, [802](#)
- GetTLNodeMap
 - IInterface, [871](#)
 - Interface, [986](#)
 - ISystem, [1013](#)
 - System, [1104](#)
- GetTLPayloadType
 - Image, [846](#)
 - Image, [901](#)
- GetTLPixelFormat
 - Image, [846](#)
 - Image, [901](#)
- GetTLPixelFormatNamespace
 - Image, [846](#)
 - Image, [901](#)
- GetTLStreamNodeMap
 - CameraBase, [573](#)
 - ICameraBase, [802](#)
- GetToolTip
 - Node, [1045](#)
 - NodeMap, [1055](#)
 - Spinnaker::GenApi, [376](#)
- GetTransferBlockID
 - ChunkData, [655](#)
 - ICChunkData, [819](#)
- GetTransferQueueCurrentBlockCount
 - ChunkData, [655](#)
 - ICChunkData, [819](#)
- GetUniqueID
 - CameraBase, [574](#)
 - ICameraBase, [803](#)
- GetUnit
 - FloatNode, [769](#)
 - IntegerNode, [982](#)
 - Spinnaker::GenApi, [376](#)

- GetUserBufferCount
 - CameraBase, [574](#)
 - ICameraBase, [803](#)
- GetUserBufferSize
 - CameraBase, [574](#)
 - ICameraBase, [803](#)
- GetUserBufferTotalSize
 - CameraBase, [575](#)
 - ICameraBase, [803](#)
- GetValidPayloadSize
 - Image, [846](#)
 - Image, [902](#)
- GetValue
 - BooleanNode, [414](#)
 - CEnumerationTRef< EnumT >, [615](#)
 - Counter, [688](#)
 - EnumEntryNode, [738](#)
 - FloatNode, [769](#)
 - IntegerNode, [982](#)
 - Spinnaker::GenApi, [376](#)
 - StringNode, [1095](#)
- GetValueOfEnvironmentVariable
 - Spinnaker::GenICam, [398](#)
- GetVendorName
 - NodeMap, [1055](#)
 - Spinnaker::GenApi, [377](#)
- GetVersion
 - InferenceBoundingBoxResult, [971](#)
- GetVersionGuid
 - NodeMap, [1055](#)
 - Spinnaker::GenApi, [377](#)
- GetVisibility
 - Node, [1046](#)
 - Spinnaker::GenApi, [377](#)
- GetWidth
 - ChunkData, [656](#)
 - ICChunkData, [820](#)
 - Image, [846](#)
 - Image, [902](#)
- GetXOffset
 - Image, [846](#)
 - Image, [902](#)
- GetXPadding
 - Image, [847](#)
 - Image, [903](#)
- GetYOffset
 - Image, [847](#)
 - Image, [903](#)
- GetYPadding
 - Image, [847](#)
 - Image, [903](#)
- GEV
 - Spinnaker::GenApi, [361](#)
- GevActionDeviceKey
 - TransportLayerInterface, [1135](#)
- GevActionGroupKey
 - TransportLayerInterface, [1135](#)
- GevActionGroupMask
 - TransportLayerInterface, [1135](#)
- GevActionTime
 - TransportLayerInterface, [1136](#)
- GevActiveLinkCount
 - Camera, [521](#)
- GevCCP
 - Camera, [521](#)
 - TransportLayerDevice, [1126](#)
- GevCCP_ControlAccess
 - Spinnaker, [282](#)
- GevCCP_EnumEntry_GevCCP_ControlAccess
 - Spinnaker, [282](#)
- GevCCP_EnumEntry_GevCCP_ExclusiveAccess
 - Spinnaker, [282](#)
- GevCCP_EnumEntry_GevCCP_OpenAccess
 - Spinnaker, [282](#)
- GevCCP_ExclusiveAccess
 - Spinnaker, [282](#)
- GevCCP_OpenAccess
 - Spinnaker, [282](#)
- GevCCPEnum
 - Spinnaker, [281](#)
- GevCCPEnums
 - Spinnaker, [282](#)
- GevCurrentDefaultGateway
 - Camera, [521](#)
- GevCurrentIPAddress
 - Camera, [522](#)
- GevCurrentIPConfigurationDHCP
 - Camera, [522](#)
- GevCurrentIPConfigurationLLA
 - Camera, [522](#)
- GevCurrentIPConfigurationPersistentIP
 - Camera, [522](#)
- GevCurrentPhysicalLinkConfiguration
 - Camera, [522](#)
- GevCurrentPhysicalLinkConfiguration_DynamicLAG
 - Spinnaker, [282](#)
- GevCurrentPhysicalLinkConfiguration_MultiLink
 - Spinnaker, [282](#)
- GevCurrentPhysicalLinkConfiguration_SingleLink
 - Spinnaker, [282](#)
- GevCurrentPhysicalLinkConfiguration_StaticLAG
 - Spinnaker, [282](#)
- GevCurrentPhysicalLinkConfigurationEnums
 - Spinnaker, [282](#)
- GevCurrentSubnetMask
 - Camera, [522](#)
- GevDeviceAutoForceIP
 - TransportLayerDevice, [1126](#)
 - TransportLayerInterface, [1136](#)
- GevDeviceDiscoverMaximumPacketSize
 - TransportLayerDevice, [1126](#)
- GevDeviceForceGateway
 - TransportLayerDevice, [1127](#)
 - TransportLayerInterface, [1136](#)
- GevDeviceForceIP
 - TransportLayerDevice, [1127](#)

- TransportLayerInterface, [1136](#)
- GevDeviceForceIPAddress
 - TransportLayerDevice, [1127](#)
 - TransportLayerInterface, [1136](#)
- GevDeviceForceSubnetMask
 - TransportLayerDevice, [1127](#)
 - TransportLayerInterface, [1136](#)
- GevDeviceGateway
 - TransportLayerDevice, [1127](#)
 - TransportLayerInterface, [1137](#)
- GevDeviceIPAddress
 - TransportLayerDevice, [1127](#)
 - TransportLayerInterface, [1137](#)
- GevDeviceIsWrongSubnet
 - TransportLayerDevice, [1128](#)
- GevDeviceMACAddress
 - TransportLayerDevice, [1128](#)
 - TransportLayerInterface, [1137](#)
- GevDeviceMaximumPacketSize
 - TransportLayerDevice, [1128](#)
- GevDeviceMaximumRetryCount
 - TransportLayerDevice, [1128](#)
- GevDeviceModelsBigEndian
 - TransportLayerDevice, [1128](#)
- GevDevicePort
 - TransportLayerDevice, [1128](#)
- GevDeviceReadAndWriteTimeout
 - TransportLayerDevice, [1129](#)
- GevDeviceSubnetMask
 - TransportLayerDevice, [1129](#)
 - TransportLayerInterface, [1137](#)
- GevDiscoveryAckDelay
 - Camera, [523](#)
- GevFirstURL
 - Camera, [523](#)
- GevGVCPExtendedStatusCodes
 - Camera, [523](#)
- GevGVCPExtendedStatusCodesSelector
 - Camera, [523](#)
- GevGVCPExtendedStatusCodesSelector_Version1_1
 - Spinnaker, [283](#)
- GevGVCPExtendedStatusCodesSelector_Version2_0
 - Spinnaker, [283](#)
- GevGVCPExtendedStatusCodesSelectorEnums
 - Spinnaker, [282](#)
- GevGVCPHeartbeatDisable
 - Camera, [523](#)
- GevGVCPPendingAck
 - Camera, [523](#)
- GevGVCPPendingTimeout
 - Camera, [524](#)
- GevGVSPExtendedIDMode
 - Camera, [524](#)
- GevGVSPExtendedIDMode_Off
 - Spinnaker, [283](#)
- GevGVSPExtendedIDMode_On
 - Spinnaker, [283](#)
- GevGVSPExtendedIDModeEnums
 - Spinnaker, [283](#)
- GevHeartbeatTimeout
 - Camera, [524](#)
- GevIEEE1588
 - Camera, [524](#)
- GevIEEE1588ClockAccuracy
 - Camera, [524](#)
- GevIEEE1588ClockAccuracy_Unknown
 - Spinnaker, [283](#)
- GevIEEE1588ClockAccuracyEnums
 - Spinnaker, [283](#)
- GevIEEE1588Mode
 - Camera, [524](#)
- GevIEEE1588Mode_Auto
 - Spinnaker, [283](#)
- GevIEEE1588Mode_SlaveOnly
 - Spinnaker, [283](#)
- GevIEEE1588ModeEnums
 - Spinnaker, [283](#)
- GevIEEE1588Status
 - Camera, [525](#)
- GevIEEE1588Status_Disabled
 - Spinnaker, [284](#)
- GevIEEE1588Status_Faulty
 - Spinnaker, [284](#)
- GevIEEE1588Status_Initializing
 - Spinnaker, [284](#)
- GevIEEE1588Status_Listening
 - Spinnaker, [284](#)
- GevIEEE1588Status_Master
 - Spinnaker, [284](#)
- GevIEEE1588Status_Passive
 - Spinnaker, [284](#)
- GevIEEE1588Status_PreMaster
 - Spinnaker, [284](#)
- GevIEEE1588Status_Slave
 - Spinnaker, [284](#)
- GevIEEE1588Status_Uncalibrated
 - Spinnaker, [284](#)
- GevIEEE1588StatusEnums
 - Spinnaker, [284](#)
- GevInterfaceDefaultGateway
 - TransportLayerSystem, [1154](#)
- GevInterfaceDefaultIPAddress
 - TransportLayerSystem, [1154](#)
- GevInterfaceDefaultSubnetMask
 - TransportLayerSystem, [1154](#)
- GevInterfaceGateway
 - TransportLayerInterface, [1137](#)
- GevInterfaceGatewaySelector
 - TransportLayerInterface, [1137](#)
- GevInterfaceMACAddress
 - TransportLayerInterface, [1138](#)
 - TransportLayerSystem, [1154](#)
- GevInterfaceMTU
 - TransportLayerInterface, [1138](#)
- GevInterfaceReceiveLinkSpeed
 - TransportLayerInterface, [1138](#)

- GevInterfaceSelector
 - Camera, [525](#)
- GevInterfaceSubnetIPAddress
 - TransportLayerInterface, [1138](#)
- GevInterfaceSubnetMask
 - TransportLayerInterface, [1138](#)
- GevInterfaceSubnetSelector
 - TransportLayerInterface, [1138](#)
- GevInterfaceTransmitLinkSpeed
 - TransportLayerInterface, [1139](#)
- GevIPConfigurationStatus
 - Camera, [525](#)
- GevIPConfigurationStatus_DHCP
 - Spinnaker, [284](#)
- GevIPConfigurationStatus_ForceIP
 - Spinnaker, [284](#)
- GevIPConfigurationStatus_LLA
 - Spinnaker, [284](#)
- GevIPConfigurationStatus_None
 - Spinnaker, [284](#)
- GevIPConfigurationStatus_PersistentIP
 - Spinnaker, [284](#)
- GevIPConfigurationStatusEnums
 - Spinnaker, [284](#)
- GevMACAddress
 - Camera, [525](#)
- GevMCDA
 - Camera, [525](#)
- GevMCPHostPort
 - Camera, [525](#)
- GevMCRC
 - Camera, [526](#)
- GevMCSP
 - Camera, [526](#)
- GevMCTT
 - Camera, [526](#)
- GevNumberOfInterfaces
 - Camera, [526](#)
- GevPAUSEFrameReception
 - Camera, [526](#)
- GevPAUSEFrameTransmission
 - Camera, [526](#)
- GevPersistentDefaultGateway
 - Camera, [527](#)
- GevPersistentIPAddress
 - Camera, [527](#)
- GevPersistentSubnetMask
 - Camera, [527](#)
- GevPhysicalLinkConfiguration
 - Camera, [527](#)
- GevPhysicalLinkConfiguration_DynamicLAG
 - Spinnaker, [285](#)
- GevPhysicalLinkConfiguration_MultiLink
 - Spinnaker, [285](#)
- GevPhysicalLinkConfiguration_SingleLink
 - Spinnaker, [285](#)
- GevPhysicalLinkConfiguration_StaticLAG
 - Spinnaker, [285](#)
- GevPhysicalLinkConfigurationEnums
 - Spinnaker, [284](#)
- GevPrimaryApplicationIPAddress
 - Camera, [527](#)
- GevPrimaryApplicationSocket
 - Camera, [527](#)
- GevPrimaryApplicationSwitchoverKey
 - Camera, [528](#)
- GevSCCFGAllInTransmission
 - Camera, [528](#)
- GevSCCFGExtendedChunkData
 - Camera, [528](#)
- GevSCCFGPacketResendDestination
 - Camera, [528](#)
- GevSCCFGUnconditionalStreaming
 - Camera, [528](#)
- GevSCDA
 - Camera, [528](#)
- GevSCPD
 - Camera, [529](#)
- GevSCPDirection
 - Camera, [529](#)
- GevSCPHostPort
 - Camera, [529](#)
- GevSCPInterfaceIndex
 - Camera, [529](#)
- GevSCPSBigEndian
 - Camera, [529](#)
- GevSCPSDoNotFragment
 - Camera, [529](#)
- GevSCPSFireTestPacket
 - Camera, [530](#)
- GevSCPSPacketSize
 - Camera, [530](#)
- GevSCSP
 - Camera, [530](#)
- GevSCZoneConfigurationLock
 - Camera, [530](#)
- GevSCZoneCount
 - Camera, [530](#)
- GevSCZoneDirectionAll
 - Camera, [530](#)
- GevSecondURL
 - Camera, [531](#)
- GevStreamChannelSelector
 - Camera, [531](#)
- GevSupportedOption
 - Camera, [531](#)
- GevSupportedOptionSelector
 - Camera, [531](#)
- GevSupportedOptionSelector_Action
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_CCPApplicationSocket
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_CommandsConcatenation
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_DiscoveryAckDelay
 - Spinnaker, [285](#)

- GevSupportedOptionSelector_DiscoveryAckDelayWritable
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_Event
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_EventData
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_ExtendedStatusCodes
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_HeartbeatDisable
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_IPConfigurationDHCP
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_IPConfigurationLLA
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_IPConfigurationPersistentIP
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_LinkSpeed
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_ManifestTable
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_MessageChannelSourceSocket
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_PacketResend
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_PendingAck
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_SerialNumber
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_StreamChannelSourceSocket
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_TestData
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_UserDefinedName
 - Spinnaker, [285](#)
- GevSupportedOptionSelector_WriteMem
 - Spinnaker, [285](#)
- GevSupportedOptionSelectorEnums
 - Spinnaker, [285](#)
- GevTimestampTickFrequency
 - Camera, [531](#)
- GevVersionMajor
 - TransportLayerDevice, [1129](#)
 - TransportLayerSystem, [1154](#)
- GevVersionMinor
 - TransportLayerDevice, [1129](#)
 - TransportLayerSystem, [1155](#)
- GigEVisionPerformance.cpp
 - AcquireImages, [1441](#)
 - argBayerRG, [1443](#)
 - argDuration, [1443](#)
 - argMaxFrames, [1443](#)
 - argNumImages, [1443](#)
 - argPacketDelay, [1443](#)
 - argPacketSize, [1443](#)
 - argPrintUsage, [1443](#)
 - argRelease, [1444](#)
 - argUserSetFrames, [1444](#)
 - cpuUsageInfo, [1444](#)
 - EnableManualFramerate, [1441](#)
 - getCameraCategory, [1441](#)
 - IsRelease, [1444](#)
 - main, [1441](#)
 - NumImagesToGrab, [1444](#)
 - PacketDelayToSet, [1444](#)
 - PacketSizeToSet, [1444](#)
 - ParseArguments, [1441](#)
 - PixelFormatToSet, [1444](#)
 - PrintAllNodes, [1442](#)
 - PrintCPUUsage, [1442](#)
 - PrintDataStreamInfo, [1442](#)
 - PrintDeviceInfo, [1442](#)
 - PrintUsage, [1442](#)
 - RunSingleCamera, [1442](#)
 - SetFrameRate, [1442](#)
 - TestDuration, [1445](#)
 - UseDuration, [1445](#)
 - UseMaxFramerate, [1445](#)
 - UserSetFramerate, [1445](#)
- GlobalCamList
 - AcquisitionMultipleCameraRecovery.cpp, [1409](#)
- GrabInfo, [785](#)
 - GrabInfo, [785](#)
 - imageEventHandler, [785](#)
 - numImagesGrabbed, [785](#)
 - numIncompleteImages, [786](#)
 - numRemovals, [786](#)
- GrabNextImageByTrigger
 - BufferHandling.cpp, [1417](#)
 - Trigger.cpp, [1479](#)
 - Trigger_QuickSpin.cpp, [1481](#)
- GrabTwoImages
 - LogicBlock.cpp, [1459](#)
- GUIXMLLocation
 - TransportLayerDevice, [1129](#)
- GUIXMLLocation_Device
 - Spinnaker, [287](#)
- GUIXMLLocation_Host
 - Spinnaker, [287](#)
- GUIXMLLocationEnum
 - Spinnaker, [285](#)
- GuiXmlManifestAddress
 - Camera, [531](#)
- GUIXMLPath
 - TransportLayerDevice, [1129](#)
- Guru
 - Spinnaker::GenApi, [361](#)
- GVCP_CHUNK_TRAILER, [786](#)
 - ChunkID, [786](#)
 - ChunkLength, [786](#)
- GVCP_EVENT_ITEM, [787](#)
 - BlockId, [787](#)
 - EventId, [787](#)
 - ReservedOrEventSize, [787](#)
 - StreamChannelId, [788](#)
 - TimestampHigh, [788](#)
 - TimestampLow, [788](#)

- GVCP_EVENT_ITEM_BASIC, 788
 - EventId, 788
 - ReservedOrEventSize, 789
- GVCP_EVENT_ITEM_EXTENDED_ID, 789
 - BlockId, 789
 - BlockId64High, 789
 - BlockId64Low, 790
 - EventId, 790
 - ReservedOrEventSize, 790
 - StreamChannelId, 790
 - TimestampHigh, 790
 - TimestampLow, 790
- GVCP_EVENT_REQUEST, 791
 - Header, 791
 - Items, 791
- GVCP_EVENT_REQUEST_EXTENDED_ID, 792
 - Header, 792
 - Items, 792
- GVCP_EVENTDATA_REQUEST, 793
 - Data, 793
 - Event, 793
 - Header, 793
- GVCP_EVENTDATA_REQUEST_EXTENDED_ID, 794
 - Data, 794
 - Event, 794
 - Header, 795
- GVCP_MESSAGE_TAGS
 - Spinnaker::GenApi, 363
- GVCP_REQUEST_HEADER, 795
 - Command, 795
 - Flags, 795
 - Length, 795
 - Magic, 796
 - ReqId, 796
- H264
 - SaveToAvi.cpp, 1470
- H264Option, 796
 - bitrate, 797
 - frameRate, 797
 - H264Option, 797
 - height, 797
 - reserved, 797
 - width, 797
- HARDWARE
 - Trigger.cpp, 1479
 - Trigger_QuickSpin.cpp, 1481
- HasChunkData
 - Image, 847
 - Image, 904
- HasCRC
 - CChunkAdapterDcam, 598
 - Image, 847
 - Image, 904
- HasInc
 - FloatNode, 769
 - Spinnaker::GenApi, 377
- Header
 - GVCP_EVENT_REQUEST, 791
 - GVCP_EVENT_REQUEST_EXTENDED_ID, 792
 - GVCP_EVENTDATA_REQUEST, 793
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, 795
- HeatmapColor
 - Spinnaker, 287
- Height
 - Camera, 532
- height
 - AVIOption, 405
 - H264Option, 797
 - MJPGOption, 1037
- HeightMax
 - Camera, 532
- HexNumber
 - Spinnaker::GenApi, 360
- HostAdapterDriverVersion
 - TransportLayerInterface, 1139
- HostAdapterName
 - TransportLayerInterface, 1139
- HostAdapterVendor
 - TransportLayerInterface, 1139
- IBase
 - Spinnaker::GenApi, 389
- IBase Interface, 103
- IBoolean
 - Spinnaker::GenApi, 389
- IBoolean Interface, 132
- ICameraBase, 798
 - ~ICameraBase, 800
 - BeginAcquisition, 800
 - CameraInternal, 806
 - DelInit, 800
 - DiscoverMaxPacketSize, 801
 - EndAcquisition, 801
 - ForceIP, 801
 - GetAccessMode, 801
 - GetBufferOwnership, 801
 - GetGuiXml, 801
 - GetNextImage, 802
 - GetNodeMap, 802
 - GetNumDataStreams, 802
 - GetNumImagesInUse, 802
 - GetTLDeviceNodeMap, 802
 - GetTLStreamNodeMap, 802
 - GetUniqueID, 803
 - GetUserBufferCount, 803
 - GetUserBufferSize, 803
 - GetUserBufferTotalSize, 803
- ICameraBase, 800
- Init, 803
- InterfaceImpl, 806
- IsInitialized, 803
- IsStreaming, 804
- IsValid, 804
- m_pCameraBaseData, 806
- operator=, 804
- ReadPort, 804

- RegisterEventHandler, [804](#), [805](#)
- SetBufferOwnership, [805](#)
- SetUserBuffers, [805](#)
- TLDevice, [806](#)
- TLStream, [807](#)
- TransportLayerDevice, [1122](#)
- TransportLayerStream, [1144](#)
- UnregisterEventHandler, [805](#)
- WritePort, [806](#)
- ICameraList, [807](#)
 - ~ICameraList, [808](#)
 - Add, [808](#)
 - Append, [809](#)
 - CameraListImpl, [811](#)
 - Clear, [809](#)
 - GetByDeviceID, [809](#)
 - GetByIndex, [809](#)
 - GetBySerial, [809](#)
 - GetSize, [809](#)
 - ICameraList, [808](#)
 - InterfacImpl, [811](#)
 - m_pCameraListData, [811](#)
 - operator=, [810](#)
 - operator[], [810](#)
 - Remove, [810](#)
 - RemoveByDeviceID, [810](#)
 - RemoveByIndex, [810](#)
 - RemoveBySerial, [810](#)
- ICategory
 - Spinnaker::GenApi, [389](#)
- ICategory Interfaces, [133](#)
- IChunkData, [811](#)
 - ~IChunkData, [813](#)
 - ChunkDataImpl, [820](#)
 - GetBlackLevel, [813](#)
 - GetCompressionMode, [813](#)
 - GetCompressionRatio, [813](#)
 - GetCounterValue, [814](#)
 - GetCRC, [814](#)
 - GetEncoderValue, [814](#)
 - GetExposureEndLineStatusAll, [814](#)
 - GetExposureTime, [814](#)
 - GetFrameID, [814](#)
 - GetGain, [815](#)
 - GetHeight, [815](#)
 - GetImage, [815](#)
 - GetInferenceBoundingBoxResult, [815](#)
 - GetInferenceConfidence, [815](#)
 - GetInferenceFrameId, [815](#)
 - GetInferenceResult, [816](#)
 - GetLinePitch, [816](#)
 - GetLineStatusAll, [816](#)
 - GetOffsetX, [816](#)
 - GetOffsetY, [816](#)
 - GetPartSelector, [816](#)
 - GetPixelDynamicRangeMax, [817](#)
 - GetPixelDynamicRangeMin, [817](#)
 - GetScan3dAxisMax, [817](#)
 - GetScan3dAxisMin, [817](#)
 - GetScan3dCoordinateOffset, [817](#)
 - GetScan3dCoordinateReferenceValue, [817](#)
 - GetScan3dCoordinateScale, [818](#)
 - GetScan3dInvalidDataValue, [818](#)
 - GetScan3dTransformValue, [818](#)
 - GetScanLineSelector, [818](#)
 - GetSequencerSetActive, [818](#)
 - GetSerialDataLength, [818](#)
 - GetStreamChannelID, [819](#)
 - GetTimerValue, [819](#)
 - GetTimestamp, [819](#)
 - GetTimestampLatchValue, [819](#)
 - GetTransferBlockID, [819](#)
 - GetTransferQueueCurrentBlockCount, [819](#)
 - GetWidth, [820](#)
 - IChunkData, [813](#)
 - ImageImpl, [820](#)
 - m_pChunkDataData, [820](#)
 - SetChunks, [820](#)
- IChunkData Class, [93](#)
- IChunkPort
 - Spinnaker::GenApi, [389](#)
- IChunkPort Interface, [134](#)
- IChunkPort.h
 - CHUNK_BASE_ADDRESS_REGISTER, [1278](#)
 - CHUNK_BASE_ADDRESS_REGISTER_LEN, [1278](#)
 - CHUNK_LENGTH_REGISTER, [1278](#)
 - CHUNK_LENGTH_REGISTER_LEN, [1278](#)
- ICommand
 - Spinnaker::GenApi, [389](#)
- ICommand Interface, [135](#)
- IDataStream, [821](#)
 - ~IDataStream, [822](#)
 - AnnounceImage, [822](#)
 - AttachBuffer, [822](#)
 - CleanupChunkAdapter, [823](#)
 - EventHandler, [752](#)
 - FlushQueueAllDiscard, [823](#)
 - GetBufferChunkData, [823](#)
 - GetBufferInfoBool8Type, [823](#)
 - GetBufferInfoPtrType, [823](#)
 - GetBufferInfoSizeType, [823](#)
 - GetBufferInfoUInt64Type, [824](#)
 - GetDeviceNodeMap, [824](#)
 - GetNextImage, [824](#)
 - GetNextImageInternal, [824](#)
 - GetNodeMap, [824](#)
 - GetNumImagesInUse, [824](#)
 - GetPort, [824](#)
 - GetStreamID, [825](#)
 - GetStreamIndex, [825](#)
 - GetStreamInfoBool8Type, [825](#)
 - GetStreamInfoSizeType, [825](#)
 - GetStreamType, [825](#)
 - GetSystemBuffer, [825](#)
 - IDataStream, [822](#)

- Image, 911
- InitChunkAdapter, 825
- IsAutoPollingEventHandlerRegistered, 826
- IsCRCCheckEnabled, 826
- IsEventHandlerRegistered, 826
- IsImageBufferInUse, 826
- IsStreaming, 826
- KillBufferEvent, 826
- RegisterImageEventHandler, 826
- ReleaseImageBuffer, 827
- ResetStreamHandle, 827
- RevokeImages, 827
- StartStream, 827
- StopStream, 827
- TransportLayerStreamInfo, 827
- UnregisterImageEventHandler, 827
- WaitOnImageEvent, 828
- IDestroy
 - Spinnaker::GenApi, 389
- IDestroy Interface, 136
- IDevFileStream
 - Spinnaker::GenApi, 355
- IDevFileStreamBase< CharType, Traits >, 828
 - close, 829
 - filebuf_type, 829
 - ios_type, 829
 - is_open, 829
 - istream_type, 829
 - open, 830
 - rdbuf, 830
- IDevFileStreamBuf
 - IDevFileStreamBuf< CharType, Traits >, 831
- IDevFileStreamBuf< CharType, Traits >, 830
 - ~IDevFileStreamBuf, 831
 - close, 832
 - IDevFileStreamBuf, 831
 - is_open, 832
 - open, 832
 - pbackfail, 832
 - underflow, 832
- IDeviceArrivalEventHandler, 833
 - ~IDeviceArrivalEventHandler, 834
 - IDeviceArrivalEventHandler, 834
 - OnDeviceArrival, 834
 - operator=, 834
- IDeviceEventHandler, 835
 - ~IDeviceEventHandler, 836
 - GetDeviceEventId, 836
 - GetDeviceEventName, 836
 - IDeviceEventHandler, 836
 - OnDeviceEvent, 837
 - operator=, 837
- IDeviceInfo
 - Spinnaker::GenApi, 390
- IDeviceInfo Interface, 137
- IDeviceRemovalEventHandler, 837
 - ~IDeviceRemovalEventHandler, 838
 - IDeviceRemovalEventHandler, 838
- OnDeviceRemoval, 839
 - operator=, 839
- idFrom
 - Spinnaker::GenApi, 358
- idNone
 - Spinnaker::GenApi, 358
- idTo
 - Spinnaker::GenApi, 358
- IEnumEntry
 - Spinnaker::GenApi, 390
- IEnumEntry Interface, 138
- IEnumeration
 - Spinnaker::GenApi, 390
- IEnumeration Interface, 139
- IEnumerationT
 - Spinnaker::GenApi, 390
- IEnumerationT Interface, 140
- IEnumReference
 - Spinnaker::GenApi, 390
- IFloat
 - Spinnaker::GenApi, 391
- IFloat Interface, 141
- IID_ISpinnakerInterface
 - SpinnakerDirectShow.h, 1352
- IIDC
 - Spinnaker::GenApi, 361
- Image, 839
 - ~Image, 841
 - CalculateStatistics, 841
 - CheckCRC, 841
 - DeepCopy, 842
 - GetBitsPerPixel, 842
 - GetBufferSize, 842
 - GetChunkData, 842
 - GetChunkLayoutId, 842
 - GetColorProcessing, 842
 - GetData, 843
 - GetDataAbsoluteMax, 843
 - GetDataAbsoluteMin, 843
 - GetFrameId, 843
 - GetHeight, 843
 - GetId, 843
 - GetImageData, 844
 - GetImageSize, 844
 - GetImageStatus, 844
 - GetNumChannels, 844
 - GetPayloadType, 844
 - GetPixelFormat, 844
 - GetPixelFormatIntType, 845
 - GetPixelFormatName, 845
 - GetPrivateData, 845
 - GetStreamIndex, 845
 - GetStride, 845
 - GetTimeStamp, 845
 - GetTLPayloadType, 846
 - GetTLPixelFormat, 846
 - GetTLPixelFormatNamespace, 846
 - GetValidPayloadSize, 846

- GetWidth, 846
- GetXOffset, 846
- GetXPadding, 847
- GetYOffset, 847
- GetYPadding, 847
- HasChunkData, 847
- HasCRC, 847
- IImage, 841
- ImageImpl, 851
- ImageListImpl, 851
- IsCompressed, 847
- IsIncomplete, 848
- IsInUse, 848
- Release, 848
- ResetImage, 848, 849
- Save, 849–851
- Stream, 851
- IImage Class, 94
- IImageEventHandler, 852
 - ~IImageEventHandler, 853
 - IImageEventHandler, 853
 - OnImageEvent, 853
 - operator=, 853
 - Stream, 854
- IImageList, 854
 - ~IImageList, 855
 - Add, 855
 - Append, 856
 - Clear, 856
 - GetByIndex, 856
 - GetByPixelFormat, 856
 - GetSize, 856
 - IImageList, 855
 - ImageListImpl, 858
 - m_pImageListData, 858
 - operator=, 856
 - operator[], 857
 - Release, 857
 - RemoveByIndex, 857
 - RemoveByPixelFormat, 857
 - Save, 857
- IImageListEventHandler, 858
 - ~IImageListEventHandler, 859
 - EventCallbackInfo, 860
 - IImageListEventHandler, 859
 - OnImageListEvent, 860
 - operator=, 860
- IImageProcessor, 860
 - ~IImageProcessor, 861
 - ApplyGamma, 862
 - Convert, 862, 863
 - GetColorProcessing, 863
 - GetNumDecompressionThreads, 863
 - IImageProcessor, 861
 - m_pImageProcessorData, 864
 - operator=, 863
 - SetColorProcessing, 863
 - SetNumDecompressionThreads, 863
- IImageStatistics, 864
 - ~IImageStatistics, 865
 - DisableAll, 866
 - EnableAll, 866
 - EnableGreyOnly, 866
 - EnableHSLOnly, 866
 - EnableRGBOnly, 866
 - GetChannelStatus, 866
 - GetHistogram, 867
 - GetMean, 867
 - GetNumPixelValues, 867
 - GetPixelValueRange, 867
 - GetRange, 867
 - GetStatistics, 868
 - IImageStatistics, 865
 - SetChannelStatus, 868
- IImageStatistics Class, 96
- IInteger
 - Spinnaker::GenApi, 391
- IInteger Interface, 142
- IInterface, 868
 - ~IInterface, 870
 - GetCameras, 870
 - GetTLNodeMap, 871
 - IInterface, 870
 - InterfaceInternal, 872
 - IsInUse, 871
 - IsValid, 871
 - m_pInterfaceData, 872
 - operator=, 871
 - ProducerImpl, 872
 - RegisterEventHandler, 871
 - SendActionCommand, 871
 - TLInterface, 873
 - TransportLayerInterface, 1133
 - UnregisterEventHandler, 872
 - UpdateCameras, 872
- IInterface Class, 97
- IInterfaceArrivalEventHandler, 873
 - ~IInterfaceArrivalEventHandler, 874
 - IInterfaceArrivalEventHandler, 874
 - OnInterfaceArrival, 875
 - operator=, 875
- IInterfaceEventHandler, 875
 - ~IInterfaceEventHandler, 876
 - IInterfaceEventHandler, 876
 - OnDeviceArrival, 877
 - OnDeviceRemoval, 877
 - operator=, 877
- IInterfaceList, 878
 - ~IInterfaceList, 879
 - Add, 879
 - Append, 879
 - Clear, 879
 - GetByIndex, 880
 - GetSize, 880
 - IInterfaceList, 879
 - InterfaceListImpl, 881

- [m_pInterfaceListData](#), 881
 - [operator=](#), 880
 - [operator\[\]](#), 880
 - [Remove](#), 880
- [IInterfaceList Class](#), 98
- [IInterfaceRemovalEventHandler](#), 881
 - [~IInterfaceRemovalEventHandler](#), 882
 - [IInterfaceRemovalEventHandler](#), 882
 - [OnInterfaceRemoval](#), 883
 - [operator=](#), 883
- [ILoggingEventHandler](#), 883
 - [~ILoggingEventHandler](#), 884
 - [ILoggingEventHandler](#), 884
 - [OnLogEvent](#), 885
 - [operator=](#), 885
- [IMAGE](#)
 - [ChunkData.cpp](#), 1419
- [Image](#), 885
 - [~Image](#), 889
 - [CalculateStatistics](#), 890
 - [CheckCRC](#), 890
 - [Create](#), 891, 892
 - [CreateShared](#), 892
 - [DeepCopy](#), 892, 893
 - [GetBitsPerPixel](#), 893
 - [GetBufferSize](#), 893
 - [GetChunkData](#), 893
 - [GetChunkLayoutId](#), 894
 - [GetColorProcessing](#), 894
 - [GetData](#), 894
 - [GetDataAbsoluteMax](#), 895
 - [GetDataAbsoluteMin](#), 895
 - [GetFrameID](#), 895
 - [GetHeight](#), 896
 - [GetID](#), 896
 - [GetImageData](#), 896
 - [GetImageSize](#), 897
 - [GetImageStatus](#), 897
 - [GetImageStatusDescription](#), 897
 - [GetNumChannels](#), 898
 - [GetPayloadType](#), 898
 - [GetPixelFormat](#), 898
 - [GetPixelFormatIntType](#), 899
 - [GetPixelFormatName](#), 899
 - [GetPrivateData](#), 899
 - [GetStreamIndex](#), 900
 - [GetStride](#), 900
 - [GetTimeStamp](#), 900
 - [GetTLPayloadType](#), 901
 - [GetTLPixelFormat](#), 901
 - [GetTLPixelFormatNamespace](#), 901
 - [GetValidPayloadSize](#), 902
 - [GetWidth](#), 902
 - [GetXOffset](#), 902
 - [GetXPadding](#), 903
 - [GetYOffset](#), 903
 - [GetYPadding](#), 903
 - [HasChunkData](#), 904
 - [HasCRC](#), 904
 - [IDataStream](#), 911
 - [Image](#), 889, 890
 - [ImageConverter](#), 911
 - [ImageConverterIpp](#), 911
 - [ImageFiler](#), 911
 - [ImageImpl](#), 911
 - [ImageListImpl](#), 911
 - [ImageProcessorImpl](#), 911
 - [ImageStatsCalculator](#), 911
 - [ImageUtilityImpl](#), 912
 - [ImageUtilityPlanarImpl](#), 912
 - [ImageUtilityPolarizationImpl](#), 912
 - [IsCompressed](#), 904
 - [IsIncomplete](#), 905
 - [IsInUse](#), 905
 - [Load](#), 905
 - [Release](#), 906
 - [ResetImage](#), 906, 907
 - [Save](#), 907–910
 - [Stream](#), 912
- [Image Class](#), 57
- [Image List Class](#), 95
- [Image Processor Class](#), 60
- [Image Utility CCM Class](#), 64
- [Image Utility Class](#), 63
- [Image Utility Heatmap Class](#), 65
- [Image Utility Polarization Class](#), 66
- [ImageComponentEnable](#)
 - [Camera](#), 532
- [ImageComponentSelector](#)
 - [Camera](#), 532
- [ImageComponentSelector_Color](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Confidence](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Disparity](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Infrared](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Intensity](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Range](#)
 - [Spinnaker](#), 287
- [ImageComponentSelector_Scatter](#)
 - [Spinnaker](#), 288
- [ImageComponentSelector_Ultraviolet](#)
 - [Spinnaker](#), 287
- [ImageComponentSelectorEnums](#)
 - [Spinnaker](#), 287
- [ImageCompressionBitrate](#)
 - [Camera](#), 532
- [ImageCompressionJPEGFormatOption](#)
 - [Camera](#), 533
- [ImageCompressionJPEGFormatOption_BaselineOptimized](#)
 - [Spinnaker](#), 288
- [ImageCompressionJPEGFormatOption_BaselineStandard](#)
 - [Spinnaker](#), 288

- ImageCompressionJPEGFormatOption_Lossless
 - Spinnaker, [288](#)
- ImageCompressionJPEGFormatOption_Progressive
 - Spinnaker, [288](#)
- ImageCompressionJPEGFormatOptionEnums
 - Spinnaker, [288](#)
- ImageCompressionMode
 - Camera, [533](#)
- ImageCompressionMode_Lossless
 - Spinnaker, [288](#)
- ImageCompressionMode_Off
 - Spinnaker, [288](#)
- ImageCompressionModeEnums
 - Spinnaker, [288](#)
- ImageCompressionQuality
 - Camera, [533](#)
- ImageCompressionRateOption
 - Camera, [533](#)
- ImageCompressionRateOption_FixBitrate
 - Spinnaker, [289](#)
- ImageCompressionRateOption_FixQuality
 - Spinnaker, [289](#)
- ImageCompressionRateOptionEnums
 - Spinnaker, [288](#)
- ImageConverter
 - Image, [911](#)
- ImageConverterIpp
 - Image, [911](#)
- ImageEventHandler, [913](#)
 - ~ImageEventHandler, [914](#)
 - ImageEventHandler, [914](#)
 - OnImageEvent, [914](#)
 - operator=, [915](#)
- imageEventHandler
 - GrabInfo, [785](#)
- ImageEventHandler Class, [58](#)
- ImageEventHandlerImpl, [915](#)
 - ~ImageEventHandlerImpl, [916](#), [917](#)
 - getImageCount, [917](#)
 - getMaxImages, [917](#)
 - ImageEventHandlerImpl, [916](#), [917](#)
 - OnImageEvent, [917](#), [918](#)
- ImageEvents.cpp
 - AcquireImages, [1446](#)
 - ConfigureImageEvents, [1446](#)
 - main, [1446](#)
 - PrintDeviceInfo, [1447](#)
 - ResetImageEvents, [1447](#)
 - RunSingleCamera, [1447](#)
 - SleepyWrapper, [1447](#)
 - WaitForImages, [1447](#)
- imageFile
 - ImageInfo, [918](#)
- ImageFileFormat
 - Spinnaker, [289](#)
- imageFileName
 - ImageInfo, [919](#)
- ImageFiler
 - Image, [911](#)
 - ImageProcessor, [940](#)
- ImageFormatControl.cpp
 - AcquireImages, [1448](#)
 - ConfigureCustomImageSettings, [1448](#)
 - main, [1448](#)
 - PrintDeviceInfo, [1448](#)
 - RunSingleCamera, [1449](#)
- ImageFormatControl_QuickSpin.cpp
 - AcquireImages, [1449](#)
 - ConfigureCustomImageSettings, [1449](#)
 - main, [1450](#)
 - PrintDeviceInfo, [1450](#)
 - RunSingleCamera, [1450](#)
- imageHeight
 - CompressedImageInfo, [687](#)
 - ImageInfo, [919](#)
- ImageImpl
 - IChunkData, [820](#)
 - Image, [851](#)
 - Image, [911](#)
- ImageInfo, [918](#)
 - imageFile, [918](#)
 - imageFileName, [919](#)
 - imageHeight, [919](#)
 - ImageInfo, [918](#)
 - imageWidth, [919](#)
 - pixelFormat, [919](#)
- imageInfos
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1411](#)
- ImageList, [920](#)
 - ~ImageList, [921](#)
 - Add, [922](#)
 - Append, [922](#)
 - Clear, [922](#)
 - GetByIndex, [923](#)
 - GetByPixelFormat, [923](#)
 - GetSize, [923](#)
 - ImageList, [921](#)
 - Load, [924](#)
 - operator=, [924](#)
 - operator[], [924](#)
 - Release, [925](#)
 - RemoveByIndex, [925](#)
 - RemoveByPixelFormat, [925](#)
 - Save, [925](#)
- ImageListEventHandler, [926](#)
 - ~ImageListEventHandler, [928](#)
 - ImageListEventHandler, [927](#)
 - OnImageListEvent, [928](#)
 - operator=, [928](#)
- ImageListEventHandler Class, [59](#)
- ImageListImpl
 - Image, [851](#)
 - ImageList, [858](#)
 - Image, [911](#)
- ImagePayloadType
 - Spinnaker, [289](#)

- imagePixelFormat
 - CompressedImageInfo, 687
- ImageProcessor, 929
 - ~ImageProcessor, 933
 - ApplyGamma, 934
 - Convert, 935–937
 - GetColorProcessing, 938
 - GetNumDecompressionThreads, 938
 - ImageFiler, 940
 - ImageProcessor, 933
 - operator=, 939
 - SetColorProcessing, 939
 - SetNumDecompressionThreads, 939
- ImageProcessorImpl
 - Image, 911
- ImagePtr, 940
 - ~ImagePtr, 942
 - ImagePtr, 941, 942
 - operator=, 942
- ImagePtr Class, 61
- ImageScalingAlgorithm
 - Spinnaker, 290
- ImageSetInspector
 - EventHandler, 752
- ImageStatistics, 943
 - ~ImageStatistics, 944
 - DisableAll, 945
 - EnableAll, 945
 - EnableGreyOnly, 945
 - EnableHSLOnly, 945
 - EnableRGBOnly, 946
 - GetChannelStatus, 946
 - GetHistogram, 946
 - GetMean, 947
 - GetNumPixelValues, 947
 - GetPixelValueRange, 947
 - GetRange, 948
 - GetStatistics, 948
 - ImageStatistics, 944, 945
 - ImageStatsCalculator, 950
 - operator=, 949
 - SetChannelStatus, 949
- ImageStatistics Class, 62
- ImageStatsCalculator
 - Image, 911
 - ImageStatistics, 950
- ImageStatus
 - Spinnaker, 290
- ImageUtility, 950
 - CreateNormalized, 950–953
 - CreateScaled, 953
- ImageUtilityCCM, 954
 - CreateColorCorrected, 954, 955
- ImageUtilityHeatmap, 956
 - CreateHeatmap, 957
 - GetHeatmapColorGradient, 958
 - GetHeatmapRange, 958
 - SetHeatmapColorGradient, 958
 - SetHeatmapRange, 959
- ImageUtilityImpl
 - Image, 912
- ImageUtilityPlanarImpl
 - Image, 912
- ImageUtilityPolarization, 959
 - CreateAolp, 960, 961
 - CreateDolp, 961, 962
 - CreateGlareReduced, 962, 963
 - CreateStokesS0, 963, 964
 - CreateStokesS1, 964
 - CreateStokesS2, 965
 - ExtractPolarQuadrant, 967
- ImageUtilityPolarizationImpl
 - Image, 912
- imageWidth
 - CompressedImageInfo, 687
 - ImageInfo, 919
- imageXOffset
 - CompressedImageInfo, 687
- imageYOffset
 - CompressedImageInfo, 688
- ImposeAccessMode
 - Node, 1046
 - Spinnaker::GenApi, 377
- ImposeMax
 - FloatNode, 769
 - IntegerNode, 982
 - Spinnaker::GenApi, 377, 378
- ImposeMin
 - FloatNode, 769
 - IntegerNode, 982
 - Spinnaker::GenApi, 378
- ImposeVisibility
 - Node, 1046
 - Spinnaker::GenApi, 378
- include/AVIRecorder.h, 1167
- include/BasePtr.h, 1168
- include/Camera.h, 1169
- include/CameraBase.h, 1169
- include/CameraDefs.h, 1170
- include/CameraList.h, 1202
- include/CameraPtr.h, 1203
- include/ChunkData.h, 1203
- include/ChunkDataInference.h, 1204
- include/DeviceArrivalEventHandler.h, 1205
- include/DeviceEventHandler.h, 1206
- include/DeviceEventUtility.h, 1207
- include/DeviceRemovalEventHandler.h, 1208
- include/EventHandler.h, 1209
- include/Exception.h, 1209
- include/Image.h, 1210
- include/ImageEventHandler.h, 1211
- include/ImageList.h, 1211
- include/ImageListEventHandler.h, 1212
- include/ImageProcessor.h, 1213
- include/ImagePtr.h, 1214
- include/ImageStatistics.h, 1215

include/ImageUtility.h, 1216
 include/ImageUtilityCCM.h, 1217
 include/ImageUtilityHeatmap.h, 1218
 include/ImageUtilityPolarization.h, 1219
 include/Interface.h, 1219
 include/Interface/ICameraBase.h, 1220
 include/Interface/ICameraList.h, 1221
 include/Interface/IChunkData.h, 1221
 include/Interface/IDeviceArrivalEventHandler.h, 1222
 include/Interface/IDeviceEventHandler.h, 1223
 include/Interface/IDeviceRemovalEventHandler.h, 1223
 include/Interface/IImage.h, 1224
 include/Interface/IImageEventHandler.h, 1225
 include/Interface/IImageList.h, 1225
 include/Interface/IImageListEventHandler.h, 1226
 include/Interface/IImageProcessor.h, 1227
 include/Interface/IImageStatistics.h, 1228
 include/Interface/IInterface.h, 1228
 include/Interface/IInterfaceArrivalEventHandler.h, 1229
 include/Interface/IInterfaceEventHandler.h, 1230
 include/Interface/IInterfaceList.h, 1231
 include/Interface/IInterfaceRemovalEventHandler.h, 1231
 include/Interface/ILoggingEventHandler.h, 1232
 include/Interface/IStream.h, 1232
 include/Interface/ISystem.h, 1233
 include/Interface/ISystemEventHandler.h, 1234
 include/InterfaceArrivalEventHandler.h, 1235
 include/InterfaceEventHandler.h, 1236
 include/InterfaceList.h, 1237
 include/InterfacePtr.h, 1237
 include/InterfaceRemovalEventHandler.h, 1238
 include/LoggingEventData.h, 1239
 include/LoggingEventDataPtr.h, 1239
 include/LoggingEventHandler.h, 1240
 include/SpinGenApi/Autovector.h, 1241
 include/SpinGenApi/Base.h, 1242
 include/SpinGenApi/BooleanNode.h, 1243
 include/SpinGenApi/CategoryNode.h, 1244
 include/SpinGenApi/ChunkAdapter.h, 1245
 include/SpinGenApi/ChunkAdapterDcam.h, 1246
 include/SpinGenApi/ChunkAdapterGeneric.h, 1247
 include/SpinGenApi/ChunkAdapterGEV.h, 1248
 include/SpinGenApi/ChunkAdapterU3V.h, 1249
 include/SpinGenApi/ChunkPort.h, 1250
 include/SpinGenApi/CommandNode.h, 1250
 include/SpinGenApi/Compatibility.h, 1251
 include/SpinGenApi/Container.h, 1252
 include/SpinGenApi/Counter.h, 1252
 include/SpinGenApi/EnumClasses.h, 1253
 include/SpinGenApi/EnumEntryNode.h, 1254
 include/SpinGenApi/EnumNode.h, 1255
 include/SpinGenApi/EnumNodeT.h, 1256
 include/SpinGenApi/EventAdapter.h, 1256
 include/SpinGenApi/EventAdapter1394.h, 1257
 include/SpinGenApi/EventAdapterGeneric.h, 1258
 include/SpinGenApi/EventAdapterGEV.h, 1258
 include/SpinGenApi/EventAdapterU3V.h, 1259
 include/SpinGenApi/EventPort.h, 1260
 include/SpinGenApi/Filestream.h, 1261
 include/SpinGenApi/FloatNode.h, 1262
 include/SpinGenApi/FloatRegNode.h, 1263
 include/SpinGenApi/GCBase.h, 1264
 include/SpinGenApi/GCString.h, 1264
 include/SpinGenApi/GCStringVector.h, 1266
 include/SpinGenApi/GCSynch.h, 1267
 include/SpinGenApi/GCTypes.h, 1268
 include/SpinGenApi/GCUtilities.h, 1271
 include/SpinGenApi/IBoolean.h, 1275
 include/SpinGenApi/ICategory.h, 1276
 include/SpinGenApi/IChunkPort.h, 1277
 include/SpinGenApi/ICommand.h, 1279
 include/SpinGenApi/IDestroy.h, 1280
 include/SpinGenApi/IDeviceInfo.h, 1281
 include/SpinGenApi/IEnumEntry.h, 1282
 include/SpinGenApi/IEnumeration.h, 1283
 include/SpinGenApi/IEnumerationT.h, 1284
 include/SpinGenApi/IFloat.h, 1285
 include/SpinGenApi/IInteger.h, 1287
 include/SpinGenApi/INode.h, 1288
 include/SpinGenApi/INodeMap.h, 1291
 include/SpinGenApi/INodeMapDyn.h, 1293
 include/SpinGenApi/IntegerNode.h, 1294
 include/SpinGenApi/IntRegNode.h, 1295
 include/SpinGenApi/IPort.h, 1296
 include/SpinGenApi/IPortConstruct.h, 1297
 include/SpinGenApi/IPortRecorder.h, 1298
 include/SpinGenApi/IRegister.h, 1299
 include/SpinGenApi/ISelector.h, 1300
 include/SpinGenApi/ISelectorDigit.h, 1301
 include/SpinGenApi/IString.h, 1302
 include/SpinGenApi/IValue.h, 1303
 include/SpinGenApi/Node.h, 1304
 include/SpinGenApi/NodeCallback.h, 1305
 include/SpinGenApi/NodeCallbackImpl.h, 1307
 include/SpinGenApi/NodeMap.h, 1307
 include/SpinGenApi/NodeMapFactory.h, 1308
 include/SpinGenApi/NodeMapRef.h, 1309
 include/SpinGenApi/Persistence.h, 1310
 include/SpinGenApi/Pointer.h, 1311
 include/SpinGenApi/PortImpl.h, 1313
 include/SpinGenApi/PortNode.h, 1314
 include/SpinGenApi/PortRecorder.h, 1315
 include/SpinGenApi/PortReplay.h, 1316
 include/SpinGenApi/PortWriteList.h, 1317
 include/SpinGenApi/Reference.h, 1318
 include/SpinGenApi/RegisterNode.h, 1319
 include/SpinGenApi/RegisterPortImpl.h, 1319
 include/SpinGenApi/SelectorSet.h, 1320
 include/SpinGenApi/SpinnakerGenApi.h, 1320
 include/SpinGenApi/SpinTestCamera.h, 1321
 include/SpinGenApi/StringNode.h, 1321
 include/SpinGenApi/StringRegNode.h, 1322
 include/SpinGenApi/StructPort.h, 1323
 include/SpinGenApi/Synch.h, 1323
 include/SpinGenApi/Types.h, 1324

- include/SpinGenApi/ValueNode.h, [1327](#)
- include/Spinnaker.h, [1328](#)
- include/SpinnakerDefs.h, [1329](#)
- include/SpinnakerDirectShow.h, [1334](#)
- include/SpinnakerPlatform.h, [1352](#)
- include/SpinUpdate.h, [1353](#)
- include/SpinVideo.h, [1356](#)
- include/SpinVideoDefs.h, [1357](#)
- include/System.h, [1358](#)
- include/SystemEventHandler.h, [1359](#)
- include/SystemPtr.h, [1360](#)
- include/TransportLayerDefs.h, [1360](#)
- include/TransportLayerDevice.h, [1362](#)
- include/TransportLayerInterface.h, [1363](#)
- include/TransportLayerStream.h, [1364](#)
- include/TransportLayerSystem.h, [1364](#)
- IncompatibleDeviceCount
 - TransportLayerInterface, [1139](#)
- IncompatibleDeviceID
 - TransportLayerInterface, [1139](#)
- IncompatibleDeviceModelName
 - TransportLayerInterface, [1140](#)
- IncompatibleDeviceSelector
 - TransportLayerInterface, [1140](#)
- IncompatibleDeviceVendorName
 - TransportLayerInterface, [1140](#)
- IncompatibleGevDeviceIPAddress
 - TransportLayerInterface, [1140](#)
- IncompatibleGevDeviceMACAddress
 - TransportLayerInterface, [1140](#)
- IncompatibleGevDeviceSubnetMask
 - TransportLayerInterface, [1140](#)
- Increasing
 - Spinnaker::GenApi, [361](#)
- Indent
 - NodeMapInfo.cpp, [1464](#)
- indexedColor_8bit
 - BMPOption, [411](#)
- INDIVIDUAL
 - NodeMapInfo.cpp, [1464](#)
- Inference.cpp
 - AcquireImages, [1452](#)
 - arrayLabelClassification, [1455](#)
 - arrayLabelDetection, [1455](#)
 - CameraCloseFile, [1452](#)
 - CameraDeleteFile, [1452](#)
 - CameraOpenFile, [1452](#)
 - CameraWriteToFile, [1452](#)
 - chosenFileUploadPersistence, [1456](#)
 - chosenInferenceNetworkType, [1456](#)
 - CLASSIFICATION, [1452](#)
 - ConfigureChunkData, [1453](#)
 - ConfigureInference, [1453](#)
 - ConfigureTestPattern, [1453](#)
 - ConfigureTrigger, [1453](#)
 - DDR, [1451](#)
 - DeleteFileOnCamera, [1453](#)
 - DETECTION, [1452](#)
 - DisableChunkData, [1453](#)
 - DisableTrigger, [1454](#)
 - DisplayChunkData, [1454](#)
 - FileUploadPersistence, [1451](#)
 - FLASH, [1451](#)
 - InferenceNetworkType, [1451](#)
 - injectedImageFilePath, [1456](#)
 - injectedImageHeight, [1456](#)
 - injectedImageWidth, [1456](#)
 - labelClassification, [1454](#)
 - labelDetection, [1454](#)
 - LoadFileIntoMemory, [1454](#)
 - main, [1454](#)
 - networkFilePath, [1456](#)
 - PrintDeviceInfo, [1455](#)
 - RunSingleCamera, [1455](#)
 - SetChunkEnable, [1455](#)
 - UploadFileToCamera, [1455](#)
- INFERENCE_BOX_TYPE_CIRCLE
 - Spinnaker, [291](#)
- INFERENCE_BOX_TYPE_RECTANGLE
 - Spinnaker, [291](#)
- INFERENCE_BOX_TYPE_ROTATED_RECTANGLE
 - Spinnaker, [291](#)
- InferenceAvailable
 - DeviceEvents.cpp, [1427](#)
- InferenceBoundingBox, [968](#)
 - boxType, [968](#)
 - circle, [968](#)
 - classId, [968](#)
 - confidence, [969](#)
 - rect, [969](#)
 - rotatedRect, [969](#)
- InferenceBoundingBoxResult, [969](#)
 - ~InferenceBoundingBoxResult, [970](#)
 - ChunkDataImpl, [972](#)
 - GetBoxAt, [971](#)
 - GetBoxCount, [971](#)
 - GetBoxSize, [971](#)
 - GetVersion, [971](#)
 - InferenceBoundingBoxResult, [970](#)
 - operator=, [971](#)
- InferenceBoxCircle, [972](#)
 - centerXCoord, [972](#)
 - centerYCoord, [972](#)
 - radius, [972](#)
- InferenceBoxRect, [973](#)
 - bottomRightXCoord, [973](#)
 - bottomRightYCoord, [973](#)
 - topLeftXCoord, [973](#)
 - topLeftYCoord, [973](#)
- InferenceBoxRotatedRect, [974](#)
 - bottomRightXCoord, [974](#)
 - bottomRightYCoord, [974](#)
 - rotationAngle, [974](#)
 - topLeftXCoord, [974](#)
 - topLeftYCoord, [974](#)
- InferenceBoxType

- Spinnaker, 291
- InferenceNetworkType
 - Inference.cpp, 1451
- Init
 - Camera, 442
 - CameraBase, 575
 - ICameraBase, 803
- InitChunkAdapter
 - IDataStream, 825
- InitializeSystem
 - FileAccess_QuickSpin.cpp, 1436
- injectedImagePath
 - Inference.cpp, 1456
- injectedImageHeight
 - Inference.cpp, 1456
- injectedImageWidth
 - Inference.cpp, 1456
- INode
 - Spinnaker::GenApi, 391
- INode Interface, 143
 - Combine, 143, 144
 - IsAvailable, 144
 - IsCacheable, 145
 - IsImplemented, 145
 - IsReadable, 145, 146
 - IsVisible, 146
 - IsWritable, 146
- INodeMap
 - Spinnaker::GenApi, 391
- INodeMap Interface, 147
- INodeMapDyn
 - Spinnaker::GenApi, 391
- INodeMapDyn Interface, 148
- int64_autovector_t, 975
 - _pCount, 977
 - _pv, 977
 - ~int64_autovector_t, 976
 - int64_autovector_t, 975, 976
 - operator delete, 976
 - operator new, 976
 - operator=, 976
 - operator[], 976, 977
 - size, 977
- IntegerGetInc
 - SpinnakerDirectShow.h, 1343
- IntegerGetIncMode
 - SpinnakerDirectShow.h, 1343
- IntegerGetMax
 - SpinnakerDirectShow.h, 1344
- IntegerGetMin
 - SpinnakerDirectShow.h, 1344
- IntegerGetValue
 - SpinnakerDirectShow.h, 1345
- IntegerNode, 978
 - ~IntegerNode, 980
 - GetFloatAlias, 980
 - GetInc, 981
 - GetIncMode, 981
 - GetListOfValidValues, 981
 - GetMax, 981
 - GetMin, 981
 - GetRepresentation, 981
 - GetUnit, 982
 - GetValue, 982
 - ImposeMax, 982
 - ImposeMin, 982
 - IntegerNode, 980
 - operator*, 983
 - operator(), 982
 - operator=, 983
 - SetReference, 983
 - SetValue, 983
- IntegerNode Class, 149
- IntegerSetValue
 - SpinnakerDirectShow.h, 1345
- INTEGRAL_CAST
 - Spinnaker::GenICam, 398
- INTEGRAL_CAST2
 - Spinnaker::GenICam, 398
- Interface, 984
 - ~Interface, 985
 - GetCameras, 985
 - GetTLNodeMap, 986
 - InterfaceInternal, 989
 - IsInUse, 986
 - IsValid, 986
 - RegisterEventHandler, 987
 - SendActionCommand, 987
 - TransportLayerInterface, 1133
 - UnregisterEventHandler, 988
 - UpdateCameras, 988
- interface
 - Types.h, 1327
- Interface Class, 67
- InterfaceArrivalEventHandler, 989
 - ~InterfaceArrivalEventHandler, 991
 - InterfaceArrivalEventHandler, 990
 - OnInterfaceArrival, 991
 - operator=, 991
- InterfaceArrivalEventHandler Class, 68
- InterfaceDisplayName
 - TransportLayerInterface, 1141
 - TransportLayerSystem, 1155
- InterfaceEventHandler, 992
 - ~InterfaceEventHandler, 993
 - InterfaceEventHandler, 993
 - OnDeviceArrival, 993
 - OnDeviceRemoval, 994
 - operator=, 994
- InterfaceEventHandler Class, 69
- InterfaceEventHandlerImpl, 995
 - ~InterfaceEventHandlerImpl, 997
 - GetInterfaceId, 997
 - InterfaceEventHandlerImpl, 996, 997
 - OnDeviceArrival, 997, 998
 - OnDeviceRemoval, 998

- PrintGenericHandlerMessage, 999
 - InterfaceID
 - TransportLayerInterface, 1141
 - TransportLayerSystem, 1155
 - InterfaceImpl
 - CameraBase, 581
 - ICameraBase, 806
 - ICameraList, 811
 - InterfaceInternal
 - IInterface, 872
 - Interface, 989
 - TransportLayerInterface, 1133
 - InterfaceList, 999
 - ~InterfaceList, 1000
 - Add, 1001
 - Append, 1001
 - Clear, 1001
 - GetByIndex, 1002
 - GetByInterfaceID, 1002
 - GetSize, 1002
 - InterfaceList, 1000, 1001
 - operator=, 1003
 - operator[], 1003
 - ProducerImpl, 1004
 - Remove, 1003
 - SystemImpl, 1004
 - InterfaceList Class, 70
 - InterfaceListImpl
 - BasePtr< T, B >, 410
 - IInterfaceList, 881
 - InterfacePtr, 1004
 - InterfacePtr, 1005
 - InterfacePtr Class, 71
 - InterfaceRemovalEventHandler, 1006
 - ~InterfaceRemovalEventHandler, 1007
 - InterfaceRemovalEventHandler, 1007
 - OnInterfaceRemoval, 1007
 - operator=, 1008
 - InterfaceRemovalEventHandler Class, 72
 - InterfaceSelector
 - TransportLayerSystem, 1155
 - InterfaceType
 - TransportLayerInterface, 1141
 - InterfaceType_CameraLink
 - Spinnaker, 291
 - InterfaceType_CameraLinkHS
 - Spinnaker, 291
 - InterfaceType_CoaXPress
 - Spinnaker, 291
 - InterfaceType_Custom
 - Spinnaker, 291
 - InterfaceType_GigEVision
 - Spinnaker, 291
 - InterfaceType_USB3Vision
 - Spinnaker, 291
 - InterfaceTypeEnum
 - Spinnaker, 291
 - InterfaceUpdateList
 - TransportLayerSystem, 1155
 - interlaced
 - PNGOption, 1065
 - intfIBase
 - Spinnaker::GenApi, 359
 - intfIBoolean
 - Spinnaker::GenApi, 359
 - intfICategory
 - Spinnaker::GenApi, 359
 - intfICommand
 - Spinnaker::GenApi, 359
 - intfIEnumEntry
 - Spinnaker::GenApi, 359
 - intfIEnumeration
 - Spinnaker::GenApi, 359
 - intfIFloat
 - Spinnaker::GenApi, 359
 - intfIInteger
 - Spinnaker::GenApi, 359
 - intfIPort
 - Spinnaker::GenApi, 359
 - intfIRegister
 - Spinnaker::GenApi, 359
 - intfIString
 - Spinnaker::GenApi, 359
 - intfIValue
 - Spinnaker::GenApi, 359
- IntRegNode, 1008
 - ~IntRegNode, 1010
 - IntRegNode, 1009, 1010
 - SetReference, 1010
- IntRegNode Class, 150
- Invalidate
 - Spinnaker::GenApi, 392
- InvalidateNode
 - CChunkPort, 609
 - CEventPort, 632
 - CPortImpl, 696
 - Node, 1046
 - Spinnaker::GenApi, 378
- InvalidateNodes
 - NodeMap, 1056
 - Spinnaker::GenApi, 378
- InverseChunkLength
 - DCAM_CHUNK_TRAILER, 712
- Invisible
 - Spinnaker::GenApi, 361
- ios_type
 - IDevFileStreamBase< CharType, Traits >, 829
 - ODevFileStreamBase< CharType, Traits >, 1060
- IPersistScript
 - Spinnaker::GenApi, 392
- IPort
 - Spinnaker::GenApi, 392
- IPort Interface, 151
- IPortConstruct
 - Spinnaker::GenApi, 392
- IPortConstruct Interface, 152

- IPortRecorder
 - Spinnaker::GenApi, [392](#)
- IPortRecorder Interface, [153](#)
- IPortReplay
 - Spinnaker::GenApi, [392](#)
- IPortWriteList
 - Spinnaker::GenApi, [393](#)
- IPv4Address
 - Spinnaker::GenApi, [360](#)
- IReference
 - Spinnaker::GenApi, [393](#)
- IRegister
 - Spinnaker::GenApi, [393](#)
- IRegister Interfaces, [154](#)
- is_open
 - IDevFileStreamBase< CharType, Traits >, [829](#)
 - IDevFileStreamBuf< CharType, Traits >, [832](#)
 - ODevFileStreamBase< CharType, Traits >, [1060](#)
 - ODevFileStreamBuf< CharType, Traits >, [1062](#)
- IsAccessModeCacheable
 - Node, [1046](#)
 - Spinnaker::GenApi, [379](#)
- IsAutoPollingEventHandlerRegistered
 - IDataStream, [826](#)
- IsAvailable
 - INode Interface, [144](#)
 - Pointer Class, [170](#)
- IsCachable
 - Node, [1046](#)
 - Spinnaker::GenApi, [379](#)
- IsCacheable
 - INode Interface, [145](#)
- IsCameraDescriptionFileDataReleased
 - CNodeMapFactory, [672](#)
- IsCompressed
 - Image, [847](#)
 - Image, [904](#)
- IsCRCCheckEnabled
 - IDataStream, [826](#)
- IsDeprecated
 - Node, [1047](#)
 - Spinnaker::GenApi, [379](#)
- IsDone
 - CommandNode, [685](#)
 - Spinnaker::GenApi, [379](#)
- ISelector
 - Spinnaker::GenApi, [393](#)
- ISelector Interface, [155](#)
- ISelectorDigit
 - Spinnaker::GenApi, [393](#)
- ISelectorDigit Interface, [156](#)
- IsEmpty
 - CNodeMapFactory, [672](#)
 - CSelectorSet, [706](#)
- IsEventHandlerRegistered
 - IDataStream, [826](#)
- IsFeature
 - Node, [1047](#)
- Spinnaker::GenApi, [379](#)
- IsImageBufferInUse
 - IDataStream, [826](#)
- IsImplemented
 - INode Interface, [145](#)
 - Pointer Class, [170](#)
- IsIncomplete
 - Image, [848](#)
 - Image, [905](#)
- IsInitialized
 - CameraBase, [576](#)
 - ICameraBase, [803](#)
- IsInUse
 - Image, [848](#)
 - IInterface, [871](#)
 - Image, [905](#)
 - Interface, [986](#)
 - ISystem, [1013](#)
 - System, [1104](#)
- IsLoaded
 - CNodeMapFactory, [672](#)
- IspEnable
 - Camera, [533](#)
- isPixelFormatColor
 - Polarization.cpp, [1469](#)
- IsPreprocessed
 - CNodeMapFactory, [673](#)
- IsReadable
 - INode Interface, [145](#), [146](#)
 - Pointer Class, [170](#)
- IsRelease
 - GigEVisionPerformance.cpp, [1444](#)
- IsSelector
 - Node, [1047](#)
- IsSelfClearing
 - EnumEntryNode, [738](#)
 - Spinnaker::GenApi, [380](#)
- IsStreamable
 - Node, [1047](#)
 - Spinnaker::GenApi, [380](#)
- IsStreaming
 - CameraBase, [576](#)
 - ICameraBase, [804](#)
 - IDataStream, [826](#)
 - SpinnakerDirectShow.h, [1345](#)
- istream_type
 - IDevFileStreamBase< CharType, Traits >, [829](#)
- IStrng
 - Spinnaker::GenApi, [394](#)
- IStrng Class, [157](#)
- IsValid
 - BasePtr< T, B >, [407](#)
 - CameraBase, [576](#)
 - CGlobalLock, [640](#)
 - CPointer< T, B >, [691](#)
 - ICameraBase, [804](#)
 - IInterface, [871](#)
 - Interface, [986](#)

- IsValueCacheValid
 - Spinnaker::GenApi, [380](#)
 - ValueNode, [1163](#)
- IsVisible
 - INode Interface, [146](#)
- IsWritable
 - INode Interface, [146](#)
 - Pointer Class, [170](#)
- ISystem, [1011](#)
 - ~ISystem, [1012](#)
 - GetCameras, [1012](#)
 - GetInterfaces, [1013](#)
 - GetLibraryVersion, [1013](#)
 - GetLoggingEventPriorityLevel, [1013](#)
 - GetTLNodeMap, [1013](#)
 - IsInUse, [1013](#)
 - ISystem, [1012](#)
 - operator=, [1013](#)
 - RegisterEventHandler, [1014](#)
 - RegisterLoggingEventHandler, [1014](#)
 - ReleaseInstance, [1014](#)
 - SendActionCommand, [1014](#)
 - SetLoggingEventPriorityLevel, [1014](#)
 - SystemPtrInternal, [1016](#)
 - TLSystem, [1016](#)
 - TransportLayerSystem, [1152](#)
 - UnregisterAllLoggingEventHandlers, [1015](#)
 - UnregisterEventHandler, [1015](#)
 - UnregisterLoggingEventHandler, [1015](#)
 - UpdateCameras, [1015](#)
 - UpdateInterfaceList, [1015](#)
- ISystem Class, [99](#)
- ISystemEventHandler, [1016](#)
 - ~ISystemEventHandler, [1017](#)
 - ISystemEventHandler, [1017](#)
 - OnInterfaceArrival, [1018](#)
 - OnInterfaceRemoval, [1018](#)
 - operator=, [1018](#)
- IsZero
 - Counter, [689](#)
- Items
 - GVCP_EVENT_REQUEST, [791](#)
 - GVCP_EVENT_REQUEST_EXTENDED_ID, [792](#)
- IValue
 - Spinnaker::GenApi, [394](#)
- IValue Class, [158](#)
- JPEGOption, [1018](#)
 - JPEGOption, [1019](#)
 - progressive, [1019](#)
 - quality, [1019](#)
 - reserved, [1019](#)
- JPG2Option, [1020](#)
 - JPG2Option, [1020](#)
 - quality, [1020](#)
 - reserved, [1021](#)
- k_LoggingLevel
 - Logging.cpp, [1457](#)
- k_numImages
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1412](#)
- k_numLoops
 - BufferHandling.cpp, [1416](#)
- kDestinationDirectory
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1412](#)
- KillBufferEvent
 - IDataStream, [826](#)
- labelClassification
 - Inference.cpp, [1454](#)
- labelDetection
 - Inference.cpp, [1454](#)
- Length
 - GVCP_REQUEST_HEADER, [795](#)
 - Spinnaker::GenApi, [394](#)
 - U3V_COMMAND_HEADER, [1158](#)
- length
 - gcstring, [781](#)
- LibraryVersion, [1021](#)
 - build, [1021](#)
 - major, [1022](#)
 - minor, [1022](#)
 - type, [1022](#)
- Linear
 - Spinnaker::GenApi, [360](#)
- LineFilterWidth
 - Camera, [533](#)
- LineFormat
 - Camera, [534](#)
- LineFormat_LVDS
 - Spinnaker, [292](#)
- LineFormat_NoConnect
 - Spinnaker, [292](#)
- LineFormat_OpenDrain
 - Spinnaker, [292](#)
- LineFormat_OptoCoupled
 - Spinnaker, [292](#)
- LineFormat_RS422
 - Spinnaker, [292](#)
- LineFormat_TriState
 - Spinnaker, [292](#)
- LineFormat_TTL
 - Spinnaker, [292](#)
- LineFormatEnums
 - Spinnaker, [291](#)
- LineInputFilterSelector
 - Camera, [534](#)
- LineInputFilterSelector_Debounce
 - Spinnaker, [292](#)
- LineInputFilterSelector_Deglitch
 - Spinnaker, [292](#)
- LineInputFilterSelectorEnums
 - Spinnaker, [292](#)
- LineInverter
 - Camera, [534](#)
- LineMode
 - Camera, [534](#)
- LineMode_Input

- Spinnaker, [292](#)
- LineMode_Output
 - Spinnaker, [292](#)
- LineModeEnums
 - Spinnaker, [292](#)
- LinePitch
 - Camera, [534](#)
- LineSelector
 - Camera, [534](#)
- LineSelector_Line0
 - Spinnaker, [293](#)
- LineSelector_Line1
 - Spinnaker, [293](#)
- LineSelector_Line2
 - Spinnaker, [293](#)
- LineSelector_Line3
 - Spinnaker, [293](#)
- LineSelectorEnums
 - Spinnaker, [292](#)
- LineSource
 - Camera, [535](#)
- LineSource_AllPixel
 - Spinnaker, [293](#)
- LineSource_AnyPixel
 - Spinnaker, [293](#)
- LineSource_Counter0Active
 - Spinnaker, [293](#)
- LineSource_Counter1Active
 - Spinnaker, [293](#)
- LineSource_ExposureActive
 - Spinnaker, [293](#)
- LineSource_FrameTriggerWait
 - Spinnaker, [293](#)
- LineSource_Line0
 - Spinnaker, [293](#)
- LineSource_Line1
 - Spinnaker, [293](#)
- LineSource_Line2
 - Spinnaker, [293](#)
- LineSource_Line3
 - Spinnaker, [293](#)
- LineSource_LogicBlock0
 - Spinnaker, [293](#)
- LineSource_LogicBlock1
 - Spinnaker, [293](#)
- LineSource_Off
 - Spinnaker, [293](#)
- LineSource_PPSSignal
 - Spinnaker, [293](#)
- LineSource_SerialPort0
 - Spinnaker, [293](#)
- LineSource_UserOutput0
 - Spinnaker, [293](#)
- LineSource_UserOutput1
 - Spinnaker, [293](#)
- LineSource_UserOutput2
 - Spinnaker, [293](#)
- LineSource_UserOutput3
 - Spinnaker, [293](#)
- Spinnaker, [293](#)
- LineSourceEnums
 - Spinnaker, [293](#)
- LineStatus
 - Camera, [535](#)
- LineStatusAll
 - Camera, [535](#)
- LinkErrorCount
 - Camera, [535](#)
- LinkUptime
 - Camera, [535](#)
- listIncrement
 - Spinnaker::GenApi, [358](#)
- LittleEndian
 - Spinnaker::GenApi, [358](#)
- Load
 - Image, [905](#)
 - ImageList, [924](#)
- LoadAndInject
 - CNodeMapFactory, [673](#)
- LoadFileIntoMemory
 - Inference.cpp, [1454](#)
- LoadFromBag
 - CFeatureBag, [634](#)
- LoadXMLFromFile
 - NodeMap, [1056](#)
 - Spinnaker::GenApi, [380](#)
- LoadXMLFromFileInject
 - NodeMap, [1056](#)
 - Spinnaker::GenApi, [380](#)
- LoadXMLFromString
 - NodeMap, [1056](#)
 - Spinnaker::GenApi, [380](#)
- LoadXMLFromStringInject
 - NodeMap, [1056](#)
 - Spinnaker::GenApi, [381](#)
- LoadXMLFromZIPData
 - NodeMap, [1057](#)
 - Spinnaker::GenApi, [381](#)
- LoadXMLFromZIPFile
 - NodeMap, [1057](#)
 - Spinnaker::GenApi, [381](#)
- Lock
 - CGlobalLock, [640](#)
 - CLOCK, [658](#), [660](#)
 - LockableObject< Object >, [1024](#)
 - LockableObject< Object >::Lock, [1023](#)
 - LockableObject< Object >, [1023](#)
 - GetLock, [1024](#)
 - Lock, [1024](#)
 - m_Lock, [1025](#)
 - LockableObject< Object >::Lock, [1022](#)
 - ~Lock, [1023](#)
 - Lock, [1023](#)
- LockEventHandlerMutex
 - SystemEventHandlerImpl, [1114](#)
- Logarithmic
 - Spinnaker::GenApi, [360](#)

- Logging EventHandler Class, [73](#)
- Logging.cpp
 - k_LoggingLevel, [1457](#)
 - main, [1457](#)
- LoggingEventData, [1025](#)
 - ~LoggingEventData, [1026](#)
 - GetCategoryName, [1026](#)
 - GetLogMessage, [1026](#)
 - GetNDC, [1026](#)
 - GetPriority, [1027](#)
 - GetPriorityName, [1027](#)
 - GetThreadName, [1027](#)
 - GetTimestamp, [1027](#)
 - LoggingEventData, [1026](#)
 - SystemImpl, [1028](#)
- LoggingEventDataPtr, [1028](#)
 - LoggingEventDataPtr, [1029](#)
- LoggingEventDataPtr Class, [74](#)
- LoggingEventHandler, [1030](#)
 - ~LoggingEventHandler, [1032](#)
 - LoggingEventHandler, [1031](#)
 - OnLogEvent, [1032](#)
 - operator=, [1032](#)
- LoggingEventHandler Class, [75](#)
- LoggingEventHandlerImpl, [1033](#)
- LogicBlock.cpp
 - AcquireImages, [1458](#)
 - ConfigureLogicBlock, [1458](#)
 - ConfigureTrigger, [1458](#)
 - GrabTwoImages, [1459](#)
 - main, [1459](#)
 - PrintDeviceInfo, [1459](#)
 - ResetExposure, [1459](#)
 - ResetTrigger, [1459](#)
 - RunSingleCamera, [1459](#)
- LogicBlockLUTInputActivation
 - Camera, [535](#)
- LogicBlockLUTInputActivation_AnyEdge
 - Spinnaker, [294](#)
- LogicBlockLUTInputActivation_FallingEdge
 - Spinnaker, [294](#)
- LogicBlockLUTInputActivation_LevelHigh
 - Spinnaker, [294](#)
- LogicBlockLUTInputActivation_LevelLow
 - Spinnaker, [294](#)
- LogicBlockLUTInputActivation_RisingEdge
 - Spinnaker, [294](#)
- LogicBlockLUTInputActivationEnums
 - Spinnaker, [293](#)
- LogicBlockLUTInputSelector
 - Camera, [536](#)
- LogicBlockLUTInputSelector_Input0
 - Spinnaker, [294](#)
- LogicBlockLUTInputSelector_Input1
 - Spinnaker, [294](#)
- LogicBlockLUTInputSelector_Input2
 - Spinnaker, [294](#)
- LogicBlockLUTInputSelector_Input3
 - Spinnaker, [294](#)
- LogicBlockLUTInputSelectorEnums
 - Spinnaker, [294](#)
- LogicBlockLUTInputSelectorEnums
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource
 - Camera, [536](#)
- LogicBlockLUTInputSource_AcquisitionActive
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Counter0End
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Counter0Start
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Counter1End
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Counter1Start
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_ExposureEnd
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_ExposureStart
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_FrameTriggerWait
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Line0
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_Line1
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_Line2
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_Line3
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_LogicBlock0
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_LogicBlock1
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_UserOutput0
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_UserOutput1
 - Spinnaker, [294](#)
- LogicBlockLUTInputSource_UserOutput2
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_UserOutput3
 - Spinnaker, [295](#)
- LogicBlockLUTInputSource_Zero
 - Spinnaker, [294](#)
- LogicBlockLUTInputSourceEnums
 - Spinnaker, [294](#)
- LogicBlockLUTOutputValue
 - Camera, [536](#)
- LogicBlockLUTOutputValueAll
 - Camera, [536](#)
- LogicBlockLUTRowIndex
 - Camera, [536](#)
- LogicBlockLUTSelector
 - Camera, [536](#)
- LogicBlockLUTSelector_Enable
 - Spinnaker, [295](#)
- LogicBlockLUTSelector_Value
 - Spinnaker, [295](#)
- LogicBlockLUTSelectorEnums

- Spinnaker, 295
- LogicBlockSelector
 - Camera, 537
- LogicBlockSelector_LogicBlock0
 - Spinnaker, 295
- LogicBlockSelector_LogicBlock1
 - Spinnaker, 295
- LogicBlockSelectorEnums
 - Spinnaker, 295
- LookupTable.cpp
 - AcquireImages, 1460
 - ConfigureLookupTables, 1460
 - main, 1460
 - PrintDeviceInfo, 1460
 - PrintRetrieveNodeFailure, 1461
 - ResetLookupTables, 1461
 - RunSingleCamera, 1461
- LUTEnable
 - Camera, 537
- LUTIndex
 - Camera, 537
- LUTSelector
 - Camera, 537
- LUTSelector_LUT1
 - Spinnaker, 296
- LUTSelectorEnums
 - Spinnaker, 295
- LUTValue
 - Camera, 537
- LUTValueAll
 - Camera, 538
- m_BaseAddress
 - CTestPortStruct< CDataStruct >, 711
- m_bOwnLock
 - CLock, 660
- m_Callbacks
 - Node, 1049
- m_CallbackType
 - CNodeCallback, 665
- m_DebugCount
 - CGlobalLock, 641
- m_enabled
 - CGlobalLockUnlocker, 643
- m_Lock
 - CGlobalLockUnlocker, 643
 - LockableObject< Object >, 1025
- m_lock
 - CLock, 661
- m_lockEx
 - CLockEx, 663
- m_NumReads
 - CTestPortStruct< CDataStruct >, 711
- m_NumWrites
 - CTestPortStruct< CDataStruct >, 711
- m_pCameraBaseData
 - ICameraBase, 806
- m_pCameraListData
 - ICameraList, 811
- m_pChunkAdapter
 - CChunkAdapter, 595
- m_pChunkDataData
 - IChunkData, 820
- m_pChunkPort
 - CChunkPort, 610
- m_pEnumeration
 - EnumNode, 744
- m_pEventAdapter
 - CEventAdapter, 619
- m_pEventData
 - EventHandler, 752
- m_pEventPort
 - CEventPort, 632
- m_plmImageListData
 - ImageList, 858
- m_plmImageProcessorData
 - ImageProcessor, 864
- m_plInterfaceData
 - IInterface, 872
- m_plInterfaceListData
 - IInterfaceList, 881
- m_pNode
 - CEventPort, 633
 - CNodeCallback, 665
- m_pNodeData
 - Node, 1049
- m_pNodeMap
 - Node, 1049
- m_pPort
 - CChunkPort, 610
- m_pPortAdapter
 - CChunkPort, 610
 - CEventPort, 633
- m_pT
 - BasePtr< T, B >, 411
 - CPointer< T, B >, 694
- m_ptrPort
 - CPortImpl, 697
- m_pWriteList
 - CPortWriteList, 700
- MACAddress
 - Spinnaker::GenApi, 360
- Magic
 - GVCP_REQUEST_HEADER, 796
- main
 - Acquisition.cpp, 1366
 - AcquisitionMultipleCameraRecovery.cpp, 1408
 - AcquisitionMultipleCamerasWriteToFile.cpp, 1411
 - AcquisitionMultipleThread.cpp, 1413
 - ActionCommand.cpp, 1415
 - BufferHandling.cpp, 1417
 - ChunkData.cpp, 1420
 - Compression.cpp, 1423
 - CounterAndTimer.cpp, 1425
 - DeviceEvents.cpp, 1427
 - Enumeration.cpp, 1428
 - Enumeration_QuickSpin.cpp, 1429

- EnumerationEvents.cpp, [1430](#)
- ExceptionHandling.cpp, [1431](#)
- Exposure.cpp, [1432](#)
- Exposure_QuickSpin.cpp, [1434](#)
- FileAccess_QuickSpin.cpp, [1436](#)
- GigEVisionPerformance.cpp, [1441](#)
- ImageEvents.cpp, [1446](#)
- ImageFormatControl.cpp, [1448](#)
- ImageFormatControl_QuickSpin.cpp, [1450](#)
- Inference.cpp, [1454](#)
- Logging.cpp, [1457](#)
- LogicBlock.cpp, [1459](#)
- LookupTable.cpp, [1460](#)
- NodeMapCallback.cpp, [1462](#)
- NodeMapInfo.cpp, [1464](#)
- Polarization.cpp, [1469](#)
- SaveToAvi.cpp, [1471](#)
- Sequencer.cpp, [1473](#)
- SerialRxTx.cpp, [1476](#)
- Trigger.cpp, [1479](#)
- Trigger_QuickSpin.cpp, [1481](#)
- Major
 - Version_t, [1164](#)
- major
 - LibraryVersion, [1022](#)
- make_NodeCallback
 - Spinnaker::GenApi, [381](#)
- MAX_LENGTH
 - SpinnakerDirectShow.h, [1352](#)
- max_size
 - gcstring, [781](#)
- maxChars
 - NodeMapInfo.cpp, [1466](#)
- MaxDeviceResetTime
 - Camera, [538](#)
- Member_NodeCallback
 - Member_NodeCallback< Client, Member >, [1035](#)
- Member_NodeCallback< Client, Member >, [1034](#)
 - Destroy, [1035](#)
 - Member_NodeCallback, [1035](#)
 - operator(), [1036](#)
 - PMEMBERFUNC, [1035](#)
- MemSet
 - CTestPortStruct< CDataStruct >, [710](#)
- MergeXMLFiles
 - Spinnaker::GenApi, [382](#)
- MILLISECOND
 - SerialRxTx.cpp, [1475](#)
- Minor
 - Version_t, [1164](#)
- minor
 - LibraryVersion, [1022](#)
- MJPEG
 - SaveToAvi.cpp, [1470](#)
- MJPEGOption, [1036](#)
 - frameRate, [1037](#)
 - height, [1037](#)
 - MJPEGOption, [1037](#)
 - quality, [1037](#)
 - reserved, [1037](#)
 - width, [1037](#)
- NA
 - Spinnaker::GenApi, [356](#)
- networkFilePath
 - Inference.cpp, [1456](#)
- NI
 - Spinnaker::GenApi, [356](#)
- No
 - Spinnaker::GenApi, [362](#)
- NoCache
 - Spinnaker::GenApi, [357](#)
- Node, [1038](#)
 - ~Node, [1041](#)
 - DeregisterCallback, [1041](#)
 - GetAccessMode, [1041](#)
 - GetAlias, [1042](#)
 - GetCachingMode, [1042](#)
 - GetCastAlias, [1042](#)
 - GetChildren, [1042](#)
 - GetDescription, [1043](#)
 - GetDeviceName, [1043](#)
 - GetDisplayName, [1043](#)
 - GetDocuURL, [1043](#)
 - GetEventID, [1043](#)
 - GetName, [1043](#)
 - GetNameSpace, [1043](#)
 - GetNodeHandle, [1044](#)
 - GetNodeMap, [1044](#)
 - GetParents, [1044](#)
 - GetPollingTime, [1044](#)
 - GetPrincipalInterfaceType, [1044](#)
 - GetProperty, [1045](#)
 - GetPropertyNames, [1045](#)
 - GetSelectedFeatures, [1045](#)
 - GetSelectingFeatures, [1045](#)
 - GetToolTip, [1045](#)
 - GetVisibility, [1046](#)
 - ImposeAccessMode, [1046](#)
 - ImposeVisibility, [1046](#)
 - InvalidateNode, [1046](#)
 - IsAccessModeCacheable, [1046](#)
 - IsCachable, [1046](#)
 - IsDeprecated, [1047](#)
 - IsFeature, [1047](#)
 - IsSelector, [1047](#)
 - IsStreamable, [1047](#)
 - m_Callbacks, [1049](#)
 - m_pNodeData, [1049](#)
 - m_pNodeMap, [1049](#)
 - Node, [1041](#)
 - operator!=, [1047](#)
 - operator==, [1047](#)
 - RegisterCallback, [1048](#)
 - SetNodeHandle, [1048](#)
 - SetNodeMap, [1048](#)
 - SetReference, [1048](#)

- Node Class, [159](#)
- NodeCallback Class, [160](#)
- NodeGetDisplayName
 - SpinnakerDirectShow.h, [1346](#)
- NodeGetType
 - SpinnakerDirectShow.h, [1346](#)
- NodesAvailable
 - SpinnakerDirectShow.h, [1347](#)
- NodesImplemented
 - SpinnakerDirectShow.h, [1347](#)
- NodesReadable
 - SpinnakerDirectShow.h, [1347](#)
- NodesWritable
 - SpinnakerDirectShow.h, [1348](#)
- NodeList_t
 - Spinnaker::GenApi, [355](#)
- NODEMAP
 - ChunkData.cpp, [1419](#)
- NodeMap, [1049](#)
 - _Ptr, [1057](#)
 - ~NodeMap, [1052](#)
 - ClearXMLCache, [1052](#)
 - CLock, [660](#)
 - Connect, [1052](#)
 - Destroy, [1052](#)
 - GetDeviceName, [1053](#)
 - GetDeviceVersion, [1053](#)
 - GetGenApiVersion, [1053](#)
 - GetLock, [1053](#)
 - GetModelName, [1053](#)
 - GetNode, [1053](#)
 - GetNodeMapHandle, [1054](#)
 - GetNodes, [1054](#)
 - GetNumNodes, [1054](#)
 - GetProductGuid, [1054](#)
 - GetSchemaVersion, [1054](#)
 - GetStandardNameSpace, [1054](#)
 - GetSupportedSchemaVersions, [1055](#)
 - GetToolTip, [1055](#)
 - GetVendorName, [1055](#)
 - GetVersionGuid, [1055](#)
 - InvalidateNodes, [1056](#)
 - LoadXMLFromFile, [1056](#)
 - LoadXMLFromFileInject, [1056](#)
 - LoadXMLFromString, [1056](#)
 - LoadXMLFromStringInject, [1056](#)
 - LoadXMLFromZIPData, [1057](#)
 - LoadXMLFromZIPFile, [1057](#)
 - NodeMap, [1051](#)
 - Poll, [1057](#)
- NodeMap Class, [161](#)
- NodeMapCallback.cpp
 - ChangeHeightAndGain, [1462](#)
 - ConfigureCallbacks, [1462](#)
 - main, [1462](#)
 - OnGainNodeUpdate, [1462](#)
 - OnHeightNodeUpdate, [1462](#)
 - PrintDeviceInfo, [1462](#)
 - ResetCallbacks, [1462](#)
 - RunSingleCamera, [1463](#)
- NodeMapFactory Class, [162](#)
- NodeMapGetNodeAtIndex
 - SpinnakerDirectShow.h, [1348](#)
- NodeMapGetNumNodes
 - SpinnakerDirectShow.h, [1349](#)
- NodeMapInfo.cpp
 - chosenRead, [1466](#)
 - Indent, [1464](#)
 - INDIVIDUAL, [1464](#)
 - main, [1464](#)
 - maxChars, [1466](#)
 - PrintBooleanNode, [1464](#)
 - PrintCategoryNodeAndAllFeatures, [1464](#)
 - PrintCommandNode, [1465](#)
 - PrintEnumerationNodeAndCurrentEntry, [1465](#)
 - PrintEnumerationSelector, [1465](#)
 - PrintFloatNode, [1465](#)
 - PrintIntegerNode, [1465](#)
 - PrintNode, [1465](#)
 - PrintStringNode, [1466](#)
 - PrintValueNode, [1466](#)
 - readType, [1464](#)
 - RunSingleCamera, [1466](#)
 - VALUE, [1464](#)
- NodeMapRef Class, [163](#)
- NodeToString
 - SpinnakerDirectShow.h, [1349](#)
- noIncrement
 - Spinnaker::GenApi, [358](#)
- None
 - Spinnaker::GenApi, [361](#)
- npos
 - gcstring, [785](#)
- NUM_ACQUISITIONMODE
 - Spinnaker, [234](#)
- NUM_ACQUISITIONSTATUSSELECTOR
 - Spinnaker, [234](#)
- NUM_ACTIONUNCONDITIONALMODE
 - Spinnaker, [235](#)
- NUM_ADCBITDEPTH
 - Spinnaker, [235](#)
- NUM_AUTOALGORITHMSELECTOR
 - Spinnaker, [235](#)
- NUM_AUTOEXPOSURECONTROLPRIORITY
 - Spinnaker, [236](#)
- NUM_AUTOEXPOSURELIGHTINGMODE
 - Spinnaker, [236](#)
- NUM_AUTOEXPOSUREMETERINGMODE
 - Spinnaker, [237](#)
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO
 - Spinnaker, [237](#)
- NUM_BALANCERATIOSELECTOR
 - Spinnaker, [237](#)
- NUM_BALANCEWHITEAUTO
 - Spinnaker, [238](#)
- NUM_BALANCEWHITEAUTOPROFILE

- Spinnaker, [238](#)
- NUM_BINNINGHORIZONTALMODE
 - Spinnaker, [238](#)
- NUM_BINNINGSELECTOR
 - Spinnaker, [239](#)
- NUM_BINNINGVERTICALMODE
 - Spinnaker, [239](#)
- NUM_BLACKLEVELAUTO
 - Spinnaker, [240](#)
- NUM_BLACKLEVELAUTOBALANCE
 - Spinnaker, [239](#)
- NUM_BLACKLEVELSELECTOR
 - Spinnaker, [240](#)
- NUM_CHUNKBLACKLEVELSELECTOR
 - Spinnaker, [242](#)
- NUM_CHUNKCOUNTERSELECTOR
 - Spinnaker, [242](#)
- NUM_CHUNKENCODERSELECTOR
 - Spinnaker, [243](#)
- NUM_CHUNKENCODERSTATUS
 - Spinnaker, [243](#)
- NUM_CHUNKEXPOSURETIMESELECTOR
 - Spinnaker, [243](#)
- NUM_CHUNKGAINSELECTOR
 - Spinnaker, [244](#)
- NUM_CHUNKIMAGECOMPONENT
 - Spinnaker, [244](#)
- NUM_CHUNKPIXELFORMAT
 - Spinnaker, [245](#)
- NUM_CHUNKREGIONID
 - Spinnaker, [245](#)
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR
 - Spinnaker, [245](#)
- NUM_CHUNKSCAN3DCOORDINATESELECTOR
 - Spinnaker, [246](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEM
 - Spinnaker, [246](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCESELECTOR
 - Spinnaker, [246](#)
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR
 - Spinnaker, [247](#)
- NUM_CHUNKSCAN3DDISTANCEUNIT
 - Spinnaker, [247](#)
- NUM_CHUNKSCAN3DOUTPUTMODE
 - Spinnaker, [248](#)
- NUM_CHUNKSELECTOR
 - Spinnaker, [249](#)
- NUM_CHUNKSOURCEID
 - Spinnaker, [249](#)
- NUM_CHUNKTIMERSELECTOR
 - Spinnaker, [249](#)
- NUM_CHUNKTRANSFERSTREAMID
 - Spinnaker, [250](#)
- NUM_CLCONFIGURATION
 - Spinnaker, [250](#)
- NUM_CLTIMESLOTSCOUNT
 - Spinnaker, [250](#)
- NUM_COLORTRANSFORMATIONSELECTOR
 - Spinnaker, [251](#)
- NUM_COLORTRANSFORMATIONVALUESELECTOR
 - Spinnaker, [252](#)
- NUM_COMPRESSIONSATURATIONPRIORITY
 - Spinnaker, [252](#)
- NUM_COUNTEREVENTACTIVATION
 - Spinnaker, [253](#)
- NUM_COUNTEREVENTSOURCE
 - Spinnaker, [253](#)
- NUM_COUNTERRESETACTIVATION
 - Spinnaker, [254](#)
- NUM_COUNTERRESETSOURCE
 - Spinnaker, [254](#)
- NUM_COUNTERSELECTOR
 - Spinnaker, [255](#)
- NUM_COUNTERSTATUS
 - Spinnaker, [255](#)
- NUM_COUNTERTRIGGERACTIVATION
 - Spinnaker, [255](#)
- NUM_COUNTERTRIGGERSOURCE
 - Spinnaker, [256](#)
- NUM_CXPCONNECTIONTESTMODE
 - Spinnaker, [256](#)
- NUM_CXPLINKCONFIGURATION
 - Spinnaker, [257](#)
- NUM_CXPLINKCONFIGURATIONPREFERRED
 - Spinnaker, [258](#)
- NUM_CXPLINKCONFIGURATIONSTATUS
 - Spinnaker, [259](#)
- NUM_CXPPOCXPSTATUS
 - Spinnaker, [259](#)
- NUM_DECIMATIONHORIZONTALMODE
 - Spinnaker, [260](#)
- NUM_DECIMATIONSELECTOR
 - Spinnaker, [260](#)
- NUM_DECIMATIONVERTICALMODE
 - Spinnaker, [260](#)
- NUM_DEFECTCORRECTIONMODE
 - Spinnaker, [261](#)
- NUM_DEINTERLACING
 - Spinnaker, [261](#)
- NUM_DEVICECHARACTERSET
 - Spinnaker, [262](#)
- NUM_DEVICECLOCKSELECTOR
 - Spinnaker, [262](#)
- NUM_DEVICECONNECTIONSTATUS
 - Spinnaker, [262](#)
- NUM_DEVICEINDICATORMODE
 - Spinnaker, [263](#)
- NUM_DEVICELINKHEARTBEATMODE
 - Spinnaker, [264](#)
- NUM_DEVICELINKTHROUGHPUTLIMITMODE
 - Spinnaker, [264](#)
- NUM_DEVICEPOWERSUPPLYSELECTOR
 - Spinnaker, [264](#)
- NUM_DEVICEREGISTERSENDIANNESS
 - Spinnaker, [264](#)
- NUM_DEVICESCANTYPE

- Spinnaker, [265](#)
NUM_DEVICESERIALPORTBAUDRATE
Spinnaker, [265](#)
NUM_DEVICESERIALPORTSELECTOR
Spinnaker, [265](#)
NUM_DEVICESTREAMCHANNELENDIANNESS
Spinnaker, [266](#)
NUM_DEVICESTREAMCHANNELTYPE
Spinnaker, [266](#)
NUM_DEVICETAPGEOMETRY
Spinnaker, [267](#)
NUM_DEVICETEMPERATURESELECTOR
Spinnaker, [268](#)
NUM_DEVICETLTYPE
Spinnaker, [268](#)
NUM_DEVICETYPE
Spinnaker, [270](#)
NUM_ENCODERMODE
Spinnaker, [270](#)
NUM_ENCODEROUTPUTMODE
Spinnaker, [270](#)
NUM_ENCODERRESETACTIVATION
Spinnaker, [271](#)
NUM_ENCODERRESETSOURCE
Spinnaker, [272](#)
NUM_ENCODERSELECTOR
Spinnaker, [272](#)
NUM_ENCODERSOURCEA
Spinnaker, [273](#)
NUM_ENCODERSOURCEB
Spinnaker, [273](#)
NUM_ENCODERSTATUS
Spinnaker, [273](#)
NUM_EVENTNOTIFICATION
Spinnaker, [275](#)
NUM_EVENTSELECTOR
Spinnaker, [275](#)
NUM_EXPOSUREACTIVEMODE
Spinnaker, [276](#)
NUM_EXPOSUREAUTO
Spinnaker, [276](#)
NUM_EXPOSUREMODE
Spinnaker, [276](#)
NUM_EXPOSURETIMEMODE
Spinnaker, [277](#)
NUM_EXPOSURETIMESELECTOR
Spinnaker, [277](#)
NUM_FILEOPENMODE
Spinnaker, [279](#)
NUM_FILEOPERATIONSELECTOR
Spinnaker, [279](#)
NUM_FILEOPERATIONSTATUS
Spinnaker, [279](#)
NUM_FILESELECTOR
Spinnaker, [280](#)
NUM_GAINAUTO
Spinnaker, [281](#)
NUM_GAINAUTOBALANCE
Spinnaker, [280](#)
NUM_GAINSELECTOR
Spinnaker, [281](#)
NUM_GEVCPP
Spinnaker, [282](#)
NUM_GEVCURRENTPHYSICALLINKCONFIGURATION
Spinnaker, [282](#)
NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR
Spinnaker, [283](#)
NUM_GEVGVSPEXTENDEDIDMODE
Spinnaker, [283](#)
NUM_GEVEEEE1588CLOCKACCURACY
Spinnaker, [283](#)
NUM_GEVEEEE1588MODE
Spinnaker, [283](#)
NUM_GEVEEEE1588STATUS
Spinnaker, [284](#)
NUM_GEVIPCONFIGURATIONSTATUS
Spinnaker, [284](#)
NUM_GEVPHYSCALLINKCONFIGURATION
Spinnaker, [285](#)
NUM_GEVSUPPORTEDOPTIONSELECTOR
Spinnaker, [285](#)
NUM_IMAGECOMPONENTSELECTOR
Spinnaker, [288](#)
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION
Spinnaker, [288](#)
NUM_IMAGECOMPRESSIONMODE
Spinnaker, [288](#)
NUM_IMAGECOMPRESSIONRATEOPTION
Spinnaker, [289](#)
NUM_LINEFORMAT
Spinnaker, [292](#)
NUM_LINEINPUTFILTERSELECTOR
Spinnaker, [292](#)
NUM_LINEMODE
Spinnaker, [292](#)
NUM_LINESELECTOR
Spinnaker, [293](#)
NUM_LINESOURCE
Spinnaker, [293](#)
NUM_LOGICBLOCKLUTINPUTACTIVATION
Spinnaker, [294](#)
NUM_LOGICBLOCKLUTINPUTSELECTOR
Spinnaker, [294](#)
NUM_LOGICBLOCKLUTINPUTSOURCE
Spinnaker, [295](#)
NUM_LOGICBLOCKLUTSELECTOR
Spinnaker, [295](#)
NUM_LOGICBLOCKSELECTOR
Spinnaker, [295](#)
NUM_LUTSELECTOR
Spinnaker, [296](#)
NUM_PIXELCOLORFILTER
Spinnaker, [296](#)
NUM_PIXELFORMAT
Spinnaker, [302](#)
NUM_PIXELFORMATINFOSELECTOR

- Spinnaker, [307](#)
- NUM_PIXELSIZE
 - Spinnaker, [308](#)
- NUM_REGIONDESTINATION
 - Spinnaker, [309](#)
- NUM_REGIONMODE
 - Spinnaker, [310](#)
- NUM_REGIONSELECTOR
 - Spinnaker, [310](#)
- NUM_RGBTRANSFORMLIGHTSOURCE
 - Spinnaker, [311](#)
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR
 - Spinnaker, [311](#)
- NUM_SCAN3DCOORDINATESELECTOR
 - Spinnaker, [311](#)
- NUM_SCAN3DCOORDINATESYSTEM
 - Spinnaker, [312](#)
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE
 - Spinnaker, [312](#)
- NUM_SCAN3DCOORDINATETRANSFORMSELECTOR
 - Spinnaker, [312](#)
- NUM_SCAN3DDISTANCEUNIT
 - Spinnaker, [313](#)
- NUM_SCAN3DOUTPUTMODE
 - Spinnaker, [314](#)
- NUM_SENSORDIGITIZATIONTAPS
 - Spinnaker, [314](#)
- NUM_SENSORSHUTTERMODE
 - Spinnaker, [314](#)
- NUM_SENSORTAPS
 - Spinnaker, [315](#)
- NUM_SEQUENCERCONFIGURATIONMODE
 - Spinnaker, [315](#)
- NUM_SEQUENCERCONFIGURATIONVALID
 - Spinnaker, [315](#)
- NUM_SEQUENCERMODE
 - Spinnaker, [316](#)
- NUM_SEQUENCERSETVALID
 - Spinnaker, [316](#)
- NUM_SEQUENCERTRIGGERACTIVATION
 - Spinnaker, [316](#)
- NUM_SEQUENCERTRIGGERSOURCE
 - Spinnaker, [317](#)
- NUM_SERIALPORTBAUDRATE
 - Spinnaker, [317](#)
- NUM_SERIALPORTPARITY
 - Spinnaker, [317](#)
- NUM_SERIALPORTSELECTOR
 - Spinnaker, [318](#)
- NUM_SERIALPORTSOURCE
 - Spinnaker, [318](#)
- NUM_SERIALPORTSTOPBITS
 - Spinnaker, [318](#)
- NUM_SOFTWARESIGNALSELECTOR
 - Spinnaker, [319](#)
- NUM_SOURCESELECTOR
 - Spinnaker, [320](#)
- NUM_TESTPATTERN
 - Spinnaker, [322](#)
- NUM_TESTPATTERNGENERATORSELECTOR
 - Spinnaker, [323](#)
- NUM_TIMERSELECTOR
 - Spinnaker, [325](#)
- NUM_TIMERSTATUS
 - Spinnaker, [325](#)
- NUM_TIMERTRIGGERACTIVATION
 - Spinnaker, [325](#)
- NUM_TIMERTRIGGERSOURCE
 - Spinnaker, [327](#)
- NUM_TRANSFERCOMPONENTSELECTOR
 - Spinnaker, [329](#)
- NUM_TRANSFERCONTROLMODE
 - Spinnaker, [329](#)
- NUM_TRANSFEROPERATIONMODE
 - Spinnaker, [329](#)
- NUM_TRANSFERQUEUEMODE
 - Spinnaker, [330](#)
- NUM_TRANSFERSELECTOR
 - Spinnaker, [330](#)
- NUM_TRANSFERSTATUSSELECTOR
 - Spinnaker, [330](#)
- NUM_TRANSFERTRIGGERACTIVATION
 - Spinnaker, [331](#)
- NUM_TRANSFERTRIGGERMODE
 - Spinnaker, [331](#)
- NUM_TRANSFERTRIGGERSELECTOR
 - Spinnaker, [331](#)
- NUM_TRANSFERTRIGGERSOURCE
 - Spinnaker, [332](#)
- NUM_TRIGGERACTIVATION
 - Spinnaker, [333](#)
- NUM_TRIGGERMODE
 - Spinnaker, [333](#)
- NUM_TRIGGEROVERLAP
 - Spinnaker, [333](#)
- NUM_TRIGGERSELECTOR
 - Spinnaker, [334](#)
- NUM_TRIGGERSOURCE
 - Spinnaker, [334](#)
- NUM_USEROUTPUTSELECTOR
 - Spinnaker, [335](#)
- NUM_USERSETDEFAULT
 - Spinnaker, [335](#)
- NUM_USERSETSELECTOR
 - Spinnaker, [335](#)
- NUM_WHITECLIPSELECTOR
 - Spinnaker, [336](#)
- NumAttachedChunks
 - AttachStatistics_t, [402](#)
- numBuffers
 - BufferHandling.cpp, [1417](#)
- NumChunkPorts
 - AttachStatistics_t, [402](#)
- NumChunks
 - AttachStatistics_t, [402](#)
- NUMDEVICEACCESSSTATUS

- Spinnaker, [261](#)
- NUMDEVICECURRENTSPEED
 - Spinnaker, [263](#)
- NUMDEVICEENDIANESSMECHANISM
 - Spinnaker, [263](#)
- NUMDEVICETYPE
 - Spinnaker, [268](#)
- NUMFILTERDRIVERSTATUS
 - Spinnaker, [280](#)
- NUMGENICAMXMLLOCATION
 - Spinnaker, [281](#)
- NUMGEVCCP
 - Spinnaker, [282](#)
- NUMGUIXMLLOCATION
 - Spinnaker, [287](#)
- numImagesGrabbed
 - GrabInfo, [785](#)
- NumImagesToGrab
 - GigEVisionPerformance.cpp, [1444](#)
- numIncompleteImages
 - GrabInfo, [786](#)
- NUMINTERFACETYPE
 - Spinnaker, [291](#)
- NumLinks
 - CNodeMapFactory::NodeStatistics_t, [1058](#)
- NumNodes
 - CNodeMapFactory::NodeStatistics_t, [1058](#)
- NUMPOESTATUS
 - Spinnaker, [309](#)
- NumProperties
 - CNodeMapFactory::NodeStatistics_t, [1058](#)
- numRemovals
 - GrabInfo, [786](#)
- NUMSTREAMBUFFERCOUNTMODE
 - Spinnaker, [321](#)
- NUMSTREAMBUFFERHANDLINGMODE
 - Spinnaker, [321](#)
- NUMSTREAMMODE
 - Spinnaker, [322](#)
- NUMSTREAMTYPE
 - Spinnaker, [322](#)
- NumStrings
 - CNodeMapFactory::NodeStatistics_t, [1058](#)
- NUMTLTYPE
 - Spinnaker, [328](#)
- NumToCString
 - Conversion, [189](#)
- ODevFileStream
 - Spinnaker::GenApi, [356](#)
- ODevFileStreamBase< CharType, Traits >, [1059](#)
 - close, [1060](#)
 - filebuf_type, [1060](#)
 - ios_type, [1060](#)
 - is_open, [1060](#)
 - open, [1060](#)
 - ostream_type, [1060](#)
 - rdbuf, [1061](#)
- ODevFileStreamBuf
 - ODevFileStreamBuf< CharType, Traits >, [1062](#)
 - ODevFileStreamBuf< CharType, Traits >, [1061](#)
 - ~ODevFileStreamBuf, [1062](#)
 - close, [1062](#)
 - is_open, [1062](#)
 - ODevFileStreamBuf, [1062](#)
 - open, [1062](#)
 - overflow, [1063](#)
 - sync, [1063](#)
 - xspuIn, [1063](#)
- OffsetX
 - Camera, [538](#)
- OffsetY
 - Camera, [538](#)
- OnDeviceArrival
 - DeviceArrivalEventHandler, [714](#)
 - IDeviceArrivalEventHandler, [834](#)
 - IInterfaceEventHandler, [877](#)
 - InterfaceEventHandler, [993](#)
 - InterfaceEventHandlerImpl, [997](#), [998](#)
- OnDeviceEvent
 - DeviceEventHandler, [718](#)
 - DeviceEventHandlerImpl, [721](#)
 - IDeviceEventHandler, [837](#)
- OnDeviceRemoval
 - DeviceRemovalEventHandler, [725](#)
 - IDeviceRemovalEventHandler, [839](#)
 - IInterfaceEventHandler, [877](#)
 - InterfaceEventHandler, [994](#)
 - InterfaceEventHandlerImpl, [998](#)
- OnGainNodeUpdate
 - NodeMapCallback.cpp, [1462](#)
- OnHeightNodeUpdate
 - NodeMapCallback.cpp, [1462](#)
- OnImageEvent
 - IImageEventHandler, [853](#)
 - ImageEventHandler, [914](#)
 - ImageEventHandlerImpl, [917](#), [918](#)
- OnImageListEvent
 - IImageListEventHandler, [860](#)
 - ImageListEventHandler, [928](#)
- OnInterfaceArrival
 - IInterfaceArrivalEventHandler, [875](#)
 - InterfaceArrivalEventHandler, [991](#)
 - ISystemEventHandler, [1018](#)
 - SystemEventHandler, [1111](#)
 - SystemEventHandlerImpl, [1114](#)
- OnInterfaceRemoval
 - IInterfaceRemovalEventHandler, [883](#)
 - InterfaceRemovalEventHandler, [1007](#)
 - ISystemEventHandler, [1018](#)
 - SystemEventHandler, [1111](#)
 - SystemEventHandlerImpl, [1114](#)
- OnLogEvent
 - ILoggingEventHandler, [885](#)
 - LoggingEventHandler, [1032](#)
- Open
 - SpinVideo, [1090](#), [1092](#)

- open
 - IDevFileStreamBase< CharType, Traits >, 830
 - IDevFileStreamBuf< CharType, Traits >, 832
 - ODevFileStreamBase< CharType, Traits >, 1060
 - ODevFileStreamBuf< CharType, Traits >, 1062
- openFile
 - FileProtocolAdapter, 763
- OpenFileToRead
 - FileAccess_QuickSpin.cpp, 1436
- OpenFileToWrite
 - FileAccess_QuickSpin.cpp, 1437
- operator bool
 - BasePtr< T, B >, 408
 - CPointer< T, B >, 691
- operator const char *
 - gcstring, 781
- operator delete
 - double_autovector_t, 727
 - gcstring, 781
 - int64_autovector_t, 976
- operator new
 - double_autovector_t, 727
 - gcstring, 781
 - int64_autovector_t, 976
- operator T*
 - BasePtr< T, B >, 408
 - CPointer< T, B >, 692
- operator unsigned int
 - Counter, 689
- operator!=
 - BasePtr< T, B >, 408
 - CPointer< T, B >, 692
 - Exception, 758
 - gcstring, 782
 - Node, 1047
 - Spinnaker::GenApi, 382
- operator<
 - gcstring, 783
- operator<<
 - GCString.h, 1265
 - Spinnaker::GenApi, 382
- operator>
 - gcstring, 783
- operator>>
 - GCString.h, 1266
 - Spinnaker::GenApi, 384
- operator*
 - CPointer< T, B >, 693
 - EnumNode, 743
 - FloatNode, 770
 - IntegerNode, 983
 - Spinnaker::GenApi, 382
 - StringNode, 1096
- operator()
 - CEnumerationTRef< EnumT >, 615
 - CNodeCallback, 665
 - CommandNode, 686
 - CPointer< T, B >, 693
 - FloatNode, 770
 - Function_NodeCallback< Function >, 775
 - IntegerNode, 982
 - Member_NodeCallback< Client, Member >, 1036
 - Spinnaker::GenApi, 382
 - StringNode, 1096
- operator+
 - gcstring, 784
- operator++
 - Counter, 689
- operator+=
 - gcstring, 782
- operator->
 - BasePtr< T, B >, 409
 - CPointer< T, B >, 693
- operator--
 - Counter, 689
- operator=
 - BasePtr< T, B >, 409
 - BooleanNode, 414
 - CameraBase, 577
 - CameraList, 586
 - CEnumerationTRef< EnumT >, 616
 - CFloatPtr, 637
 - CNodeMapFactory, 673
 - CNodeMapRef, 676
 - CNodeMapRefT< TCameraParams >, 682, 683
 - CPointer< T, B >, 693
 - DeviceArrivalEventHandler, 715
 - DeviceEventHandler, 718
 - DeviceRemovalEventHandler, 725
 - double_autovector_t, 727
 - EnumNode, 743
 - EventHandler, 751
 - Exception, 758
 - FloatNode, 770
 - gcstring, 783
 - ICameraBase, 804
 - ICameraList, 810
 - IDeviceArrivalEventHandler, 834
 - IDeviceEventHandler, 837
 - IDeviceRemovalEventHandler, 839
 - IImageEventHandler, 853
 - IImageList, 856
 - IImageListEventHandler, 860
 - IImageProcessor, 863
 - IInterface, 871
 - IInterfaceArrivalEventHandler, 875
 - IInterfaceEventHandler, 877
 - IInterfaceList, 880
 - IInterfaceRemovalEventHandler, 883
 - ILoggingEventHandler, 885
 - ImageEventHandler, 915
 - ImageList, 924
 - ImageListEventHandler, 928
 - ImageProcessor, 939
 - ImagePtr, 942
 - ImageStatistics, 949

- InferenceBoundingBoxResult, 971
- int64_autovector_t, 976
- IntegerNode, 983
- InterfaceArrivalEventHandler, 991
- InterfaceEventHandler, 994
- InterfaceList, 1003
- InterfaceRemovalEventHandler, 1008
- ISystem, 1013
- ISystemEventHandler, 1018
- LoggingEventHandler, 1032
- Spinnaker::GenApi, 383
- StringNode, 1096
- SystemEventHandler, 1112
- operator==
 - BasePtr< T, B >, 409, 410
 - CFeatureBag, 635
 - CPointer< T, B >, 693, 694
 - Exception, 758
 - gcstring, 783
 - Node, 1047
 - Spinnaker, 337
 - Spinnaker::GenApi, 384
- operator[]
 - CameraList, 586
 - double_autovector_t, 727
 - ICameraList, 810
 - ImageList, 857
 - InterfaceList, 880
 - ImageList, 924
 - int64_autovector_t, 976, 977
 - InterfaceList, 1003
- ostream_type
 - ODevFileStreamBase< CharType, Traits >, 1060
- overflow
 - ODevFileStreamBuf< CharType, Traits >, 1063
- PacketDelayToSet
 - GigEVisionPerformance.cpp, 1444
- PacketResendRequestCount
 - Camera, 538
- PacketSizeToSet
 - GigEVisionPerformance.cpp, 1444
- ParseArguments
 - GigEVisionPerformance.cpp, 1441
- ParseDeviceEventExposureEnd
 - DeviceEventUtility, 722
- ParseDeviceEventInference
 - DeviceEventUtility, 723
- PayloadSize
 - Camera, 539
- pbackfail
 - IDevFileStreamBuf< CharType, Traits >, 832
- PCFreq
 - PerformanceCounter, 191
- PerformanceCounter, 190
 - CounterStart, 191
 - GetPerformanceCounter, 191
 - PCFreq, 191
 - StartPerformanceCounter, 191
- Persistence Class, 164
- PersistFeature
 - CFeatureBag, 635
 - Spinnaker::GenApi, 384
- PGMOption, 1063
 - binaryFile, 1064
 - PGMOption, 1064
 - reserved, 1064
- PixelColorFilter
 - Camera, 539
- PixelColorFilter_BayerBG
 - Spinnaker, 296
- PixelColorFilter_BayerGB
 - Spinnaker, 296
- PixelColorFilter_BayerGR
 - Spinnaker, 296
- PixelColorFilter_BayerRG
 - Spinnaker, 296
- PixelColorFilter_None
 - Spinnaker, 296
- PixelColorFilterEnums
 - Spinnaker, 296
- PixelDynamicRangeMax
 - Camera, 539
- PixelDynamicRangeMin
 - Camera, 539
- PixelFormat
 - Camera, 539
- pixelFormat
 - ImageInfo, 919
- PixelFormat_B10
 - Spinnaker, 298
- PixelFormat_B12
 - Spinnaker, 299
- PixelFormat_B12_Jpeg
 - Spinnaker, 302
- PixelFormat_B16
 - Spinnaker, 299
- PixelFormat_B8
 - Spinnaker, 298
- PixelFormat_BayerBG10
 - Spinnaker, 297
- PixelFormat_BayerBG10p
 - Spinnaker, 297
- PixelFormat_BayerBG10Packed
 - Spinnaker, 297
- PixelFormat_BayerBG12
 - Spinnaker, 297
- PixelFormat_BayerBG12p
 - Spinnaker, 297
- PixelFormat_BayerBG12Packed
 - Spinnaker, 297
- PixelFormat_BayerBG16
 - Spinnaker, 297
- PixelFormat_BayerBG8
 - Spinnaker, 296
- PixelFormat_BayerGB10
 - Spinnaker, 297

- PixelFormat_BayerGB10p
 - Spinnaker, [297](#)
- PixelFormat_BayerGB10Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerGB12
 - Spinnaker, [297](#)
- PixelFormat_BayerGB12p
 - Spinnaker, [297](#)
- PixelFormat_BayerGB12Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerGB16
 - Spinnaker, [297](#)
- PixelFormat_BayerGB8
 - Spinnaker, [296](#)
- PixelFormat_BayerGR10
 - Spinnaker, [297](#)
- PixelFormat_BayerGR10p
 - Spinnaker, [297](#)
- PixelFormat_BayerGR10Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerGR12
 - Spinnaker, [297](#)
- PixelFormat_BayerGR12p
 - Spinnaker, [297](#)
- PixelFormat_BayerGR12Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerGR16
 - Spinnaker, [296](#)
- PixelFormat_BayerGR8
 - Spinnaker, [296](#)
- PixelFormat_BayerRG10
 - Spinnaker, [297](#)
- PixelFormat_BayerRG10p
 - Spinnaker, [297](#)
- PixelFormat_BayerRG10Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerRG12
 - Spinnaker, [297](#)
- PixelFormat_BayerRG12p
 - Spinnaker, [297](#)
- PixelFormat_BayerRG12Packed
 - Spinnaker, [297](#)
- PixelFormat_BayerRG16
 - Spinnaker, [296](#)
- PixelFormat_BayerRG8
 - Spinnaker, [296](#)
- PixelFormat_BayerRGPolarized10p
 - Spinnaker, [301](#)
- PixelFormat_BayerRGPolarized12p
 - Spinnaker, [301](#)
- PixelFormat_BayerRGPolarized16
 - Spinnaker, [301](#)
- PixelFormat_BayerRGPolarized8
 - Spinnaker, [301](#)
- PixelFormat_BGR10
 - Spinnaker, [298](#)
- PixelFormat_BGR10p
 - Spinnaker, [298](#)
- PixelFormat_BGR12
 - Spinnaker, [298](#)
- PixelFormat_BGR12p
 - Spinnaker, [298](#)
- PixelFormat_BGR14
 - Spinnaker, [298](#)
- PixelFormat_BGR16
 - Spinnaker, [298](#)
- PixelFormat_BGR565p
 - Spinnaker, [298](#)
- PixelFormat_BGR8
 - Spinnaker, [297](#)
- PixelFormat_BGRa10
 - Spinnaker, [298](#)
- PixelFormat_BGRa10p
 - Spinnaker, [298](#)
- PixelFormat_BGRa12
 - Spinnaker, [298](#)
- PixelFormat_BGRa12p
 - Spinnaker, [298](#)
- PixelFormat_BGRa14
 - Spinnaker, [298](#)
- PixelFormat_BGRa16
 - Spinnaker, [298](#)
- PixelFormat_BGRa8
 - Spinnaker, [297](#)
- PixelFormat_BiColorBGRG10
 - Spinnaker, [299](#)
- PixelFormat_BiColorBGRG10p
 - Spinnaker, [299](#)
- PixelFormat_BiColorBGRG12
 - Spinnaker, [299](#)
- PixelFormat_BiColorBGRG12p
 - Spinnaker, [299](#)
- PixelFormat_BiColorBGRG8
 - Spinnaker, [299](#)
- PixelFormat_BiColorRGBG10
 - Spinnaker, [300](#)
- PixelFormat_BiColorRGBG10p
 - Spinnaker, [300](#)
- PixelFormat_BiColorRGBG12
 - Spinnaker, [300](#)
- PixelFormat_BiColorRGBG12p
 - Spinnaker, [300](#)
- PixelFormat_BiColorRGBG8
 - Spinnaker, [300](#)
- PixelFormat_Confidence1
 - Spinnaker, [299](#)
- PixelFormat_Confidence16
 - Spinnaker, [299](#)
- PixelFormat_Confidence1p
 - Spinnaker, [299](#)
- PixelFormat_Confidence32f
 - Spinnaker, [299](#)
- PixelFormat_Confidence8
 - Spinnaker, [299](#)
- PixelFormat_Coord3D_A10p
 - Spinnaker, [299](#)

- PixelFormat_Coord3D_A12p
Spinnaker, [299](#)
- PixelFormat_Coord3D_A16
Spinnaker, [299](#)
- PixelFormat_Coord3D_A32f
Spinnaker, [299](#)
- PixelFormat_Coord3D_A8
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC10p
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC10p_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC12p
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC12p_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC16
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC16_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC32f
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC32f_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC8
Spinnaker, [299](#)
- PixelFormat_Coord3D_ABC8_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC10p
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC10p_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC12p
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC12p_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC16
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC16_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC32f
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC32f_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC8
Spinnaker, [299](#)
- PixelFormat_Coord3D_AC8_Planar
Spinnaker, [299](#)
- PixelFormat_Coord3D_B10p
Spinnaker, [299](#)
- PixelFormat_Coord3D_B12p
Spinnaker, [299](#)
- PixelFormat_Coord3D_B16
Spinnaker, [299](#)
- PixelFormat_Coord3D_B32f
Spinnaker, [299](#)
- PixelFormat_Coord3D_B8
Spinnaker, [299](#)
- PixelFormat_Coord3D_C10p
Spinnaker, [299](#)
- PixelFormat_Coord3D_C12p
Spinnaker, [299](#)
- PixelFormat_Coord3D_C16
Spinnaker, [299](#)
- PixelFormat_Coord3D_C32f
Spinnaker, [299](#)
- PixelFormat_Coord3D_C8
Spinnaker, [299](#)
- PixelFormat_G10
Spinnaker, [298](#)
- PixelFormat_G12
Spinnaker, [298](#)
- PixelFormat_G16
Spinnaker, [298](#)
- PixelFormat_G8
Spinnaker, [298](#)
- PixelFormat_GB12
Spinnaker, [302](#)
- PixelFormat_GB12_Jpeg
Spinnaker, [302](#)
- PixelFormat_GR12
Spinnaker, [302](#)
- PixelFormat_GR12_Jpeg
Spinnaker, [302](#)
- PixelFormat_JPEGColor8
Spinnaker, [302](#)
- PixelFormat_JPEGMono8
Spinnaker, [302](#)
- PixelFormat_LLCBayerRG8
Spinnaker, [301](#)
- PixelFormat_LLCMono8
Spinnaker, [301](#)
- PixelFormat_Mono10
Spinnaker, [297](#)
- PixelFormat_Mono10p
Spinnaker, [297](#)
- PixelFormat_Mono10Packed
Spinnaker, [297](#)
- PixelFormat_Mono12
Spinnaker, [297](#)
- PixelFormat_Mono12p
Spinnaker, [297](#)
- PixelFormat_Mono12Packed
Spinnaker, [297](#)
- PixelFormat_Mono14
Spinnaker, [297](#)
- PixelFormat_Mono16
Spinnaker, [296](#)
- PixelFormat_Mono16s
Spinnaker, [297](#)
- PixelFormat_Mono1p
Spinnaker, [297](#)
- PixelFormat_Mono2p
Spinnaker, [297](#)
- PixelFormat_Mono32f
Spinnaker, [297](#)

- PixelFormat_Mono4p
 - Spinnaker, [297](#)
- PixelFormat_Mono8
 - Spinnaker, [296](#)
- PixelFormat_Mono8s
 - Spinnaker, [297](#)
- PixelFormat_Polarized10p
 - Spinnaker, [301](#)
- PixelFormat_Polarized12p
 - Spinnaker, [301](#)
- PixelFormat_Polarized16
 - Spinnaker, [301](#)
- PixelFormat_Polarized8
 - Spinnaker, [301](#)
- PixelFormat_R10
 - Spinnaker, [298](#)
- PixelFormat_R12
 - Spinnaker, [298](#)
- PixelFormat_R12_Jpeg
 - Spinnaker, [302](#)
- PixelFormat_R16
 - Spinnaker, [298](#)
- PixelFormat_R8
 - Spinnaker, [298](#)
- PixelFormat_Raw16
 - Spinnaker, [302](#)
- PixelFormat_Raw8
 - Spinnaker, [302](#)
- PixelFormat_RGB10
 - Spinnaker, [298](#)
- PixelFormat_RGB10_Planar
 - Spinnaker, [298](#)
- PixelFormat_RGB10p
 - Spinnaker, [298](#)
- PixelFormat_RGB10p32
 - Spinnaker, [298](#)
- PixelFormat_RGB12
 - Spinnaker, [298](#)
- PixelFormat_RGB12_Planar
 - Spinnaker, [298](#)
- PixelFormat_RGB12p
 - Spinnaker, [298](#)
- PixelFormat_RGB14
 - Spinnaker, [298](#)
- PixelFormat_RGB16
 - Spinnaker, [298](#)
- PixelFormat_RGB16_Planar
 - Spinnaker, [298](#)
- PixelFormat_RGB16s
 - Spinnaker, [298](#)
- PixelFormat_RGB32f
 - Spinnaker, [298](#)
- PixelFormat_RGB565p
 - Spinnaker, [298](#)
- PixelFormat_RGB8
 - Spinnaker, [298](#)
- PixelFormat_RGB8_Planar
 - Spinnaker, [298](#)
- PixelFormat_RGB8Packed
 - Spinnaker, [296](#)
- PixelFormat_RGBa10
 - Spinnaker, [298](#)
- PixelFormat_RGBa10p
 - Spinnaker, [298](#)
- PixelFormat_RGBa12
 - Spinnaker, [298](#)
- PixelFormat_RGBa12p
 - Spinnaker, [298](#)
- PixelFormat_RGBa14
 - Spinnaker, [298](#)
- PixelFormat_RGBa16
 - Spinnaker, [298](#)
- PixelFormat_RGBa32f
 - Spinnaker, [298](#)
- PixelFormat_RGBa8
 - Spinnaker, [298](#)
- PixelFormat_SCF1WBWG10
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG10p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG12
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG12p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG14
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG16
 - Spinnaker, [300](#)
- PixelFormat_SCF1WBWG8
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB10
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB10p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB12
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB12p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB14
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB16
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWB8
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR10
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR10p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR12
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR12p
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR14
 - Spinnaker, [300](#)
- PixelFormat_SCF1WGWR16
 - Spinnaker, [300](#)

- PixelFormat_YUV411_8_UYYVYY
 - Spinnaker, [301](#)
- PixelFormat_YUV411Packed
 - Spinnaker, [297](#)
- PixelFormat_YUV422_8
 - Spinnaker, [301](#)
- PixelFormat_YUV422_8_UYVY
 - Spinnaker, [301](#)
- PixelFormat_YUV422Packed
 - Spinnaker, [297](#)
- PixelFormat_YUV444Packed
 - Spinnaker, [297](#)
- PixelFormat_YUV8_UYV
 - Spinnaker, [301](#)
- PixelFormatEnums
 - Spinnaker, [296](#)
- PixelFormatInfoID
 - Camera, [540](#)
- PixelFormatInfoSelector
 - Camera, [540](#)
- PixelFormatInfoSelector_B10
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_B12
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_B16
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_B8
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BayerBG10
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerBG10p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerBG12
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerBG12p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerBG16
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerBG8
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGB10
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGB10p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGB12
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGB12p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGB16
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGB8
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_BayerGR10
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGR10p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGR12
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGR12p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGR16
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerGR8
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BayerRGPolarized10p
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized12p
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized16
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized8
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_BGR10
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGR10p
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR12
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR12p
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR14
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR16
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR565p
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_BGR8
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa10
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa10p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa12
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa12p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa14
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa16
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BGRa8
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_BiColorBGRG10
 - Spinnaker, [305](#)

- PixelFormatInfoSelector_BiColorBGRG10p
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorBGRG12
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorBGRG12p
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorBGRG8
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorRGBG10
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorRGBG10p
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorRGBG12
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorRGBG12p
Spinnaker, [305](#)
- PixelFormatInfoSelector_BiColorRGBG8
Spinnaker, [305](#)
- PixelFormatInfoSelector_Confidence1
Spinnaker, [305](#)
- PixelFormatInfoSelector_Confidence16
Spinnaker, [305](#)
- PixelFormatInfoSelector_Confidence1p
Spinnaker, [305](#)
- PixelFormatInfoSelector_Confidence32f
Spinnaker, [305](#)
- PixelFormatInfoSelector_Confidence8
Spinnaker, [305](#)
- PixelFormatInfoSelector_Coord3D_A10p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_A12p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_A16
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_A32f
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_A8
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC10p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC10p_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC12p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC12p_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC16
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC16_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC32f
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC32f_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC8
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC10p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC12p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC16
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC32f
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC8
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_B10p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_B12p
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_B16
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_B32f
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_B8
Spinnaker, [304](#)
- PixelFormatInfoSelector_Coord3D_C10p
Spinnaker, [305](#)
- PixelFormatInfoSelector_Coord3D_C12p
Spinnaker, [305](#)
- PixelFormatInfoSelector_Coord3D_C16
Spinnaker, [305](#)
- PixelFormatInfoSelector_Coord3D_C32f
Spinnaker, [305](#)
- PixelFormatInfoSelector_Coord3D_C8
Spinnaker, [305](#)
- PixelFormatInfoSelector_G10
Spinnaker, [304](#)
- PixelFormatInfoSelector_G12
Spinnaker, [304](#)
- PixelFormatInfoSelector_G16
Spinnaker, [304](#)
- PixelFormatInfoSelector_G8
Spinnaker, [304](#)
- PixelFormatInfoSelector_JPEGColor8
Spinnaker, [307](#)
- PixelFormatInfoSelector_JPEGMono8
Spinnaker, [307](#)
- PixelFormatInfoSelector_LLCBayerRG8
Spinnaker, [307](#)
- PixelFormatInfoSelector_LLCMono8
Spinnaker, [307](#)
- PixelFormatInfoSelector_Mono10
Spinnaker, [302](#)

- PixelFormatInfoSelector_Mono10p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono12
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono12p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono14
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono16
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono16s
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono1p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono2p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono32f
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono4p
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono8
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Mono8s
 - Spinnaker, [302](#)
- PixelFormatInfoSelector_Polarized10p
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_Polarized12p
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_Polarized16
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_Polarized8
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_R10
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_R12
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_R16
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_R8
 - Spinnaker, [304](#)
- PixelFormatInfoSelector_RGB10
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB10_Planar
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB10p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB10p32
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB12
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB12_Planar
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB12p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB14
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB16
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB16_Planar
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB16s
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB32f
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB565p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB8
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGB8_Planar
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa10
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa10p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa12
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa12p
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa14
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa16
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa32f
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_RGBa8
 - Spinnaker, [303](#)
- PixelFormatInfoSelector_SCF1WBWG10
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG10p
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG12
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG12p
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG14
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG16
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WBWG8
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB10
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB10p
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB12
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB12p
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB14
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB16
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWB8
 - Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WGWR10
 - Spinnaker, [305](#)

- PixelFormatInfoSelector_SCF1WGWR10p
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WGWR12
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WGWR12p
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WGWR14
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WGWR16
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WGWR8
Spinnaker, [305](#)
- PixelFormatInfoSelector_SCF1WRWG10
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG10p
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG12
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG12p
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG14
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG16
Spinnaker, [306](#)
- PixelFormatInfoSelector_SCF1WRWG8
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr10_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr10p_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr12_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr12p_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr411_8
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_10
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_10_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_10p
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_10p_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_12
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_12_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_12p
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_12p_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_8
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr422_8_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_10_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_10p_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_12_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_12p_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_10
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_10p
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12p
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_8
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr601_8_CbYCr
Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr709_10_CbYCr
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_10p_CbYCr
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_12_CbYCr
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_12p_CbYCr
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10p
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12p
Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY
Spinnaker, [307](#)

- PixelFormatInfoSelector_YCbCr709_422_8
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr709_8_CbYCr
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YCbCr8
 - Spinnaker, [306](#)
- PixelFormatInfoSelector_YCbCr8_CbYCr
 - Spinnaker, [306](#)
- PixelFormatInfoSelector_YUV411_8_UYYVYY
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YUV422_8
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YUV422_8_UYVY
 - Spinnaker, [307](#)
- PixelFormatInfoSelector_YUV8_UYV
 - Spinnaker, [307](#)
- PixelFormatInfoSelectorEnums
 - Spinnaker, [302](#)
- PixelFormatIntType
 - Spinnaker, [308](#)
- PixelFormatToSet
 - GigEVisionPerformance.cpp, [1444](#)
- PixelSize
 - Camera, [540](#)
- PixelSize_Bpp1
 - Spinnaker, [308](#)
- PixelSize_Bpp10
 - Spinnaker, [308](#)
- PixelSize_Bpp12
 - Spinnaker, [308](#)
- PixelSize_Bpp14
 - Spinnaker, [308](#)
- PixelSize_Bpp16
 - Spinnaker, [308](#)
- PixelSize_Bpp2
 - Spinnaker, [308](#)
- PixelSize_Bpp20
 - Spinnaker, [308](#)
- PixelSize_Bpp24
 - Spinnaker, [308](#)
- PixelSize_Bpp30
 - Spinnaker, [308](#)
- PixelSize_Bpp32
 - Spinnaker, [308](#)
- PixelSize_Bpp36
 - Spinnaker, [308](#)
- PixelSize_Bpp4
 - Spinnaker, [308](#)
- PixelSize_Bpp48
 - Spinnaker, [308](#)
- PixelSize_Bpp64
 - Spinnaker, [308](#)
- PixelSize_Bpp8
 - Spinnaker, [308](#)
- PixelSize_Bpp96
 - Spinnaker, [308](#)
- PixelSizeEnums
 - Spinnaker, [308](#)
- PMEBERFUNC
 - Member_NodeCallback< Client, Member >, [1035](#)
- PNGOption, [1064](#)
 - compressionLevel, [1065](#)
 - interlaced, [1065](#)
 - PNGOption, [1065](#)
 - reserved, [1065](#)
- POEStatus
 - TransportLayerInterface, [1141](#)
- POEStatus_NotSupported
 - Spinnaker, [309](#)
- POEStatus_PowerOff
 - Spinnaker, [309](#)
- POEStatus_PowerOn
 - Spinnaker, [309](#)
- POEStatusEnum
 - Spinnaker, [309](#)
- Pointer Class, [165](#)
 - CBasePtr, [166](#)
 - CBooleanPtr, [166](#)
 - CCategoryPtr, [167](#)
 - CChunkPortPtr, [167](#)
 - CCommandPtr, [167](#)
 - CDeviceInfoPtr, [167](#)
 - CEnumEntryPtr, [167](#)
 - CEnumerationPtr, [167](#)
 - CIntegerPtr, [168](#)
 - CNodeMapDynPtr, [168](#)
 - CNodeMapPtr, [168](#)
 - CNodePtr, [168](#)
 - CPortConstructPtr, [168](#)
 - CPortPtr, [168](#)
 - CPortRecorderPtr, [169](#)
 - CPortReplayPtr, [169](#)
 - CPortWriteListPtr, [169](#)
 - CRegisterPtr, [169](#)
 - CSelectorPtr, [169](#)
 - CStringPtr, [169](#)
 - CValuePtr, [170](#)
 - GetInterfaceName, [170](#)
 - IsAvailable, [170](#)
 - IsImplemented, [170](#)
 - IsReadable, [170](#)
 - IsWritable, [170](#)
- Polarization.cpp
 - AcquireImages, [1467](#)
 - ConfigureStream, [1467](#)
 - CreateAndSaveAolpDolpImages, [1468](#)
 - CreateAndSaveGlareReducedImage, [1468](#)
 - CreateAndSaveStokesImages, [1468](#)
 - CreateHeatmapImages, [1468](#)
 - CreateNormalizedImage, [1468](#)
 - ExtractAndSavePolarQuadImages, [1468](#)
 - GetQuadFileNameAppendage, [1469](#)
 - isPixelFormatColor, [1469](#)
 - main, [1469](#)

- PrintDeviceInfo, 1469
- RunSingleCamera, 1469
- SaveImage, 1469
- PolarizationQuadrant
 - Spinnaker, 309
- Poll
 - NodeMap, 1057
 - Spinnaker::GenApi, 384
- PortImpl Class, 171
- PortNode, 1066
 - ~PortNode, 1068
 - CacheChunkData, 1068
 - GetChunkID, 1068
 - GetPortHandle, 1069
 - GetSwapEndianness, 1069
 - PortNode, 1068
 - Read, 1069
 - Replay, 1069
 - SetPortImpl, 1069
 - SetReference, 1070
 - StartRecording, 1071
 - StopRecording, 1071
 - Write, 1071
- PortNode Class, 172
- PortRecorder, 1072
 - ~PortRecorder, 1073
 - GetAccessMode, 1073
 - PortRecorder, 1073
 - Read, 1074
 - Replay, 1074
 - SetReference, 1074
 - StartRecording, 1074
 - StopRecording, 1075
 - Write, 1075
- PortRecorder Class, 173
- PortReplay, 1076
 - ~PortReplay, 1077
 - GetAccessMode, 1077
 - GetPortReplayHandle, 1078
 - PortReplay, 1077
 - Read, 1078
 - Replay, 1078
 - SetReference, 1078
 - Write, 1079
- PortReplay Class, 174
- PortWriteList Class, 175
- PowerSupplyCurrent
 - Camera, 540
- PowerSupplyVoltage
 - Camera, 540
- PPMOption, 1079
 - binaryFile, 1080
 - PPMOption, 1080
 - reserved, 1080
- Prefix
 - U3V_COMMAND_HEADER, 1158
- Preprocess
 - CNodeMapFactory, 673
 - PreprocessXMLFromFile
 - Spinnaker::GenApi, 384
 - PreprocessXMLFromZIPFile
 - Spinnaker::GenApi, 385
 - PrintAllNodes
 - GigEVisionPerformance.cpp, 1442
 - PrintBooleanNode
 - NodeMapInfo.cpp, 1464
 - PrintCategoryNodeAndAllFeatures
 - NodeMapInfo.cpp, 1464
 - PrintCommandNode
 - NodeMapInfo.cpp, 1465
 - PrintCPUUsage
 - GigEVisionPerformance.cpp, 1442
 - PrintDataStreamInfo
 - GigEVisionPerformance.cpp, 1442
 - PrintDebugMessage
 - FileAccess_QuickSpin.cpp, 1437
 - PrintDeviceInfo
 - Acquisition.cpp, 1366
 - AcquisitionMultipleCamerasWriteToFile.cpp, 1411
 - AcquisitionMultipleThread.cpp, 1413
 - ActionCommand.cpp, 1415
 - BufferHandling.cpp, 1418
 - ChunkData.cpp, 1420
 - Compression.cpp, 1423
 - CounterAndTimer.cpp, 1425
 - DeviceEvents.cpp, 1427
 - Exposure.cpp, 1433
 - Exposure_QuickSpin.cpp, 1434
 - FileAccess_QuickSpin.cpp, 1437
 - GigEVisionPerformance.cpp, 1442
 - ImageEvents.cpp, 1447
 - ImageFormatControl.cpp, 1448
 - ImageFormatControl_QuickSpin.cpp, 1450
 - Inference.cpp, 1455
 - LogicBlock.cpp, 1459
 - LookupTable.cpp, 1460
 - NodeMapCallback.cpp, 1462
 - Polarization.cpp, 1469
 - SaveToAvi.cpp, 1471
 - Sequencer.cpp, 1473
 - SerialRxTx.cpp, 1476
 - Trigger.cpp, 1479
 - Trigger_QuickSpin.cpp, 1482
 - PrintEnumerationNodeAndCurrentEntry
 - NodeMapInfo.cpp, 1465
 - PrintEnumerationSelector
 - NodeMapInfo.cpp, 1465
 - PrintExampleStatistics
 - AcquisitionMultipleCameraRecovery.cpp, 1408
 - PrintFloatNode
 - NodeMapInfo.cpp, 1465
 - PrintGenericHandlerMessage
 - InterfaceEventHandlerImpl, 999
 - PrintIntegerNode
 - NodeMapInfo.cpp, 1465
 - PrintNode

- NodeMapInfo.cpp, [1465](#)
- PrintResultMessage
 - FileAccess_QuickSpin.cpp, [1437](#)
- PrintRetrieveNodeFailure
 - LookupTable.cpp, [1461](#)
 - Sequencer.cpp, [1473](#)
- PrintStringNode
 - NodeMapInfo.cpp, [1466](#)
- PrintUsage
 - FileAccess_QuickSpin.cpp, [1437](#)
 - GigEVisionPerformance.cpp, [1442](#)
- PrintValueNode
 - NodeMapInfo.cpp, [1466](#)
- ProcessCompressedImagesFromFile
 - Compression.cpp, [1423](#)
- ProducerImpl
 - Interface, [872](#)
 - InterfaceList, [1004](#)
- progressive
 - JPEGOption, [1019](#)
- PureNumber
 - Spinnaker::GenApi, [360](#)
- quality
 - JPEGOption, [1019](#)
 - JPG2Option, [1020](#)
 - MJPGOption, [1037](#)
- QueryInterface
 - Enumeration.cpp, [1428](#)
 - Enumeration_QuickSpin.cpp, [1429](#)
- radius
 - InferenceBoxCircle, [972](#)
- rdbuf
 - IDevFileStreamBase< CharType, Traits >, [830](#)
 - ODevFileStreamBase< CharType, Traits >, [1061](#)
- Read
 - CChunkPort, [609](#)
 - CEventPort, [632](#)
 - CPortImpl, [696](#)
 - CRegisterPortImpl, [703](#)
 - CTestPortStruct< CDataStruct >, [710](#)
 - PortNode, [1069](#)
 - PortRecorder, [1074](#)
 - PortReplay, [1078](#)
- read
 - FileProtocolAdapter, [763](#)
- ReadPort
 - CameraBase, [577](#)
 - ICameraBase, [804](#)
- ReadRegister
 - CRegisterPortImpl, [703](#)
- readType
 - NodeMapInfo.cpp, [1464](#)
- rect
 - InferenceBoundingBox, [969](#)
- Reference Interfaces, [176](#)
- RefreshCameraList
 - AcquisitionMultipleCameraRecovery.cpp, [1409](#)
- RegionDestination
 - Camera, [541](#)
- RegionDestination_Stream0
 - Spinnaker, [309](#)
- RegionDestination_Stream1
 - Spinnaker, [309](#)
- RegionDestination_Stream2
 - Spinnaker, [309](#)
- RegionDestinationEnums
 - Spinnaker, [309](#)
- RegionMode
 - Camera, [541](#)
- RegionMode_Off
 - Spinnaker, [310](#)
- RegionMode_On
 - Spinnaker, [310](#)
- RegionModeEnums
 - Spinnaker, [309](#)
- RegionSelector
 - Camera, [541](#)
- RegionSelector_All
 - Spinnaker, [310](#)
- RegionSelector_Region0
 - Spinnaker, [310](#)
- RegionSelector_Region1
 - Spinnaker, [310](#)
- RegionSelector_Region2
 - Spinnaker, [310](#)
- RegionSelectorEnums
 - Spinnaker, [310](#)
- Register
 - Spinnaker::GenApi, [385](#)
- RegisterAllInterfaceEvents
 - SystemEventHandlerImpl, [1114](#)
- RegisterCallback
 - Node, [1048](#)
 - Spinnaker::GenApi, [385](#)
- RegisterEventHandler
 - CameraBase, [577](#), [578](#)
 - ICameraBase, [804](#), [805](#)
 - Interface, [871](#)
 - Interface, [987](#)
 - ISystem, [1014](#)
 - System, [1105](#)
- RegisterImageEventHandler
 - IDataStream, [826](#)
- RegisterInterfaceEventToSystem
 - SystemEventHandlerImpl, [1115](#)
- RegisterLoggingEventHandler
 - ISystem, [1014](#)
 - System, [1105](#)
- RegisterNode, [1080](#)
 - ~RegisterNode, [1082](#)
 - Get, [1082](#)
 - GetAddress, [1084](#)
 - GetLength, [1084](#)
 - RegisterNode, [1082](#)
 - Set, [1084](#)

- SetReference, [1084](#)
- RegisterNode Class, [177](#)
- RegisterPortImpl Class, [178](#)
- Release
 - Image, [848](#)
 - ImageList, [857](#)
 - Image, [906](#)
 - ImageList, [925](#)
- ReleaseCameraDescriptionFileData
 - CNodeMapFactory, [673](#)
- ReleaseImageBuffer
 - IDataStream, [827](#)
- ReleaseInstance
 - ISystem, [1014](#)
 - System, [1105](#)
- Remove
 - CameraList, [586](#)
 - ICameraList, [810](#)
 - IInterfaceList, [880](#)
 - InterfaceList, [1003](#)
- RemoveByDeviceID
 - CameraList, [587](#)
 - ICameraList, [810](#)
- RemoveByIndex
 - CameraList, [587](#)
 - ICameraList, [810](#)
 - ImageList, [857](#)
 - ImageList, [925](#)
- RemoveByPixelFormat
 - ImageList, [857](#)
 - ImageList, [925](#)
- RemoveBySerial
 - CameraList, [587](#)
 - ICameraList, [810](#)
- ReplaceEnvironmentVariables
 - Spinnaker::GenICam, [399](#)
- Replay
 - CPortImpl, [697](#)
 - CPortWriteList, [700](#)
 - PortNode, [1069](#)
 - PortRecorder, [1074](#)
 - PortReplay, [1078](#)
 - Spinnaker::GenApi, [386](#)
- ReqId
 - GVCP_REQUEST_HEADER, [796](#)
 - U3V_COMMAND_HEADER, [1159](#)
- Reserved
 - U3V_EVENT_DATA, [1159](#)
- reserved
 - AVIOption, [405](#)
 - BMPOption, [412](#)
 - H264Option, [797](#)
 - JPEGOption, [1019](#)
 - JPG2Option, [1021](#)
 - MJPGOption, [1037](#)
 - PGMOption, [1064](#)
 - PNGOption, [1065](#)
 - PPMOption, [1080](#)
 - SIOption, [1087](#)
 - TIFFOption, [1119](#)
- ReservedOrEventSize
 - GVCP_EVENT_ITEM, [787](#)
 - GVCP_EVENT_ITEM_BASIC, [789](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
- ResetCallbacks
 - NodeMapCallback.cpp, [1462](#)
- ResetCameraUserSetToDefault
 - AcquisitionMultipleCameraRecovery.cpp, [1409](#)
- ResetDeviceEvents
 - DeviceEvents.cpp, [1427](#)
- ResetExposure
 - Exposure.cpp, [1433](#)
 - Exposure_QuickSpin.cpp, [1434](#)
 - LogicBlock.cpp, [1459](#)
- ResetGVCPHeartbeat
 - Acquisition.cpp, [1366](#)
 - AcquisitionMultipleThread.cpp, [1413](#)
 - Compression.cpp, [1423](#)
- ResetImage
 - Image, [848](#), [849](#)
 - Image, [906](#), [907](#)
- ResetImageEvents
 - ImageEvents.cpp, [1447](#)
- ResetLookupTables
 - LookupTable.cpp, [1461](#)
- ResetSequencer
 - Sequencer.cpp, [1473](#)
- ResetStatistics
 - CTestPortStruct< CDataStruct >, [710](#)
- ResetStreamHandle
 - IDataStream, [827](#)
- ResetTrigger
 - BufferHandling.cpp, [1418](#)
 - CounterAndTimer.cpp, [1425](#)
 - LogicBlock.cpp, [1459](#)
 - Trigger.cpp, [1479](#)
 - Trigger_QuickSpin.cpp, [1482](#)
- resize
 - gcstring, [783](#)
- Restore
 - CSelectorSet, [707](#)
 - Spinnaker::GenApi, [386](#)
- result
 - DeviceEventInferenceData, [722](#)
- RetrieveImagesFromFiles
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1411](#)
- ReverseX
 - Camera, [541](#)
- ReverseY
 - Camera, [541](#)
- RevokeImages
 - IDataStream, [827](#)
- RgbTransformLightSource
 - Camera, [542](#)
- RgbTransformLightSource_Cloudy6500K
 - Spinnaker, [311](#)

- RgbTransformLightSource_CoolFluorescent4000K
 - Spinnaker, [310](#)
- RgbTransformLightSource_Custom
 - Spinnaker, [311](#)
- RgbTransformLightSource_Daylight5000K
 - Spinnaker, [311](#)
- RgbTransformLightSource_General
 - Spinnaker, [310](#)
- RgbTransformLightSource_Shade8000K
 - Spinnaker, [311](#)
- RgbTransformLightSource_Tungsten2800K
 - Spinnaker, [310](#)
- RgbTransformLightSource_WarmFluorescent3000K
 - Spinnaker, [310](#)
- RgbTransformLightSourceEnums
 - Spinnaker, [310](#)
- RO
 - Spinnaker::GenApi, [356](#)
- rotatedRect
 - InferenceBoundingBox, [969](#)
- rotationAngle
 - InferenceBoxRotatedRect, [974](#)
- RunCameras
 - AcquisitionMultipleCamerasWriteToFile.cpp, [1411](#)
- RunMultipleCameras
 - AcquisitionMultipleThread.cpp, [1413](#)
 - ActionCommand.cpp, [1415](#)
- RunSingleCamera
 - Acquisition.cpp, [1366](#)
 - BufferHandling.cpp, [1418](#)
 - ChunkData.cpp, [1420](#)
 - Compression.cpp, [1423](#)
 - CounterAndTimer.cpp, [1425](#)
 - DeviceEvents.cpp, [1427](#)
 - Exposure.cpp, [1433](#)
 - Exposure_QuickSpin.cpp, [1434](#)
 - GigEVisionPerformance.cpp, [1442](#)
 - ImageEvents.cpp, [1447](#)
 - ImageFormatControl.cpp, [1449](#)
 - ImageFormatControl_QuickSpin.cpp, [1450](#)
 - Inference.cpp, [1455](#)
 - LogicBlock.cpp, [1459](#)
 - LookupTable.cpp, [1461](#)
 - NodeMapCallback.cpp, [1463](#)
 - NodeMapInfo.cpp, [1466](#)
 - Polarization.cpp, [1469](#)
 - SaveToAvi.cpp, [1471](#)
 - Sequencer.cpp, [1473](#)
 - SerialRxTx.cpp, [1476](#)
 - Trigger.cpp, [1480](#)
 - Trigger_QuickSpin.cpp, [1482](#)
- RW
 - Spinnaker::GenApi, [356](#)
- Saturation
 - Camera, [542](#)
- SaturationEnable
 - Camera, [542](#)
- Save
 - Image, [849–851](#)
 - ImageList, [857](#)
 - Image, [907–910](#)
 - ImageList, [925](#)
- SaveImage
 - Polarization.cpp, [1469](#)
- SaveToAvi.cpp
 - AcquireImages, [1471](#)
 - chosenVideoType, [1471](#)
 - H264, [1470](#)
 - main, [1471](#)
 - MJPEG, [1470](#)
 - PrintDeviceInfo, [1471](#)
 - RunSingleCamera, [1471](#)
 - SaveVectorToVideo, [1471](#)
 - UNCOMPRESSED, [1470](#)
 - videoType, [1470](#)
- SaveVectorToVideo
 - SaveToAvi.cpp, [1471](#)
- Scan3dAxisMax
 - Camera, [542](#)
- Scan3dAxisMin
 - Camera, [542](#)
- Scan3dCoordinateOffset
 - Camera, [543](#)
- Scan3dCoordinateReferenceSelector
 - Camera, [543](#)
- Scan3dCoordinateReferenceSelector_RotationX
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelector_RotationY
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelector_RotationZ
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationX
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationY
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationZ
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceSelectorEnums
 - Spinnaker, [311](#)
- Scan3dCoordinateReferenceValue
 - Camera, [543](#)
- Scan3dCoordinateScale
 - Camera, [543](#)
- Scan3dCoordinateSelector
 - Camera, [543](#)
- Scan3dCoordinateSelector_CoordinateA
 - Spinnaker, [311](#)
- Scan3dCoordinateSelector_CoordinateB
 - Spinnaker, [311](#)
- Scan3dCoordinateSelector_CoordinateC
 - Spinnaker, [311](#)
- Scan3dCoordinateSelectorEnums
 - Spinnaker, [311](#)
- Scan3dCoordinateSystem
 - Camera, [543](#)
- Scan3dCoordinateSystem_Cartesian

- Spinnaker, [312](#)
- Scan3dCoordinateSystem_Cylindrical
 - Spinnaker, [312](#)
- Scan3dCoordinateSystem_Spherical
 - Spinnaker, [312](#)
- Scan3dCoordinateSystemEnums
 - Spinnaker, [311](#)
- Scan3dCoordinateSystemReference
 - Camera, [544](#)
- Scan3dCoordinateSystemReference_Anchor
 - Spinnaker, [312](#)
- Scan3dCoordinateSystemReference_Transformed
 - Spinnaker, [312](#)
- Scan3dCoordinateSystemReferenceEnums
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector
 - Camera, [544](#)
- Scan3dCoordinateTransformSelector_RotationX
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector_RotationY
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector_RotationZ
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector_TranslationX
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector_TranslationY
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelector_TranslationZ
 - Spinnaker, [312](#)
- Scan3dCoordinateTransformSelectorEnums
 - Spinnaker, [312](#)
- Scan3dDistanceUnit
 - Camera, [544](#)
- Scan3dDistanceUnit_Inch
 - Spinnaker, [313](#)
- Scan3dDistanceUnit_Millimeter
 - Spinnaker, [313](#)
- Scan3dDistanceUnitEnums
 - Spinnaker, [313](#)
- Scan3dInvalidDataFlag
 - Camera, [544](#)
- Scan3dInvalidDataValue
 - Camera, [544](#)
- Scan3dOutputMode
 - Camera, [544](#)
- Scan3dOutputMode_CalibratedABC_Grid
 - Spinnaker, [313](#)
- Scan3dOutputMode_CalibratedABC_PointCloud
 - Spinnaker, [313](#)
- Scan3dOutputMode_CalibratedAC
 - Spinnaker, [313](#)
- Scan3dOutputMode_CalibratedAC_Linescan
 - Spinnaker, [313](#)
- Scan3dOutputMode_CalibratedC
 - Spinnaker, [313](#)
- Scan3dOutputMode_CalibratedC_Linescan
 - Spinnaker, [313](#)
- Scan3dOutputMode_DisparityC
 - Spinnaker, [314](#)
- Scan3dOutputMode_DisparityC_Linescan
 - Spinnaker, [314](#)
- Scan3dOutputMode_RectifiedC
 - Spinnaker, [313](#)
- Scan3dOutputMode_RectifiedC_Linescan
 - Spinnaker, [314](#)
- Scan3dOutputMode_UncalibratedC
 - Spinnaker, [313](#)
- Scan3dOutputModeEnums
 - Spinnaker, [313](#)
- Scan3dTransformValue
 - Camera, [545](#)
- SecondsCounter, [191](#)
 - endTime, [192](#)
 - GetSecondsCounter, [192](#)
 - StartSecondsCounter, [192](#)
 - startTime, [192](#)
 - timeDiff, [192](#)
- SelectorSet Class, [179](#)
- SendActionCommand
 - IInterface, [871](#)
 - Interface, [987](#)
 - ISystem, [1014](#)
 - System, [1106](#)
- Sensor
 - CCMSettings, [611](#)
- SensorDescription
 - Camera, [545](#)
- SensorDigitizationTaps
 - Camera, [545](#)
- SensorDigitizationTaps_Eight
 - Spinnaker, [314](#)
- SensorDigitizationTaps_Four
 - Spinnaker, [314](#)
- SensorDigitizationTaps_One
 - Spinnaker, [314](#)
- SensorDigitizationTaps_Ten
 - Spinnaker, [314](#)
- SensorDigitizationTaps_Three
 - Spinnaker, [314](#)
- SensorDigitizationTaps_Two
 - Spinnaker, [314](#)
- SensorDigitizationTapsEnums
 - Spinnaker, [314](#)
- SensorHeight
 - Camera, [545](#)
- SensorShutterMode
 - Camera, [545](#)
- SensorShutterMode_Global
 - Spinnaker, [314](#)
- SensorShutterMode_GlobalReset
 - Spinnaker, [314](#)
- SensorShutterMode_Rolling
 - Spinnaker, [314](#)
- SensorShutterModeEnums
 - Spinnaker, [314](#)
- SensorTaps

- Camera, [545](#)
- SensorTaps_Eight
 - Spinnaker, [315](#)
- SensorTaps_Four
 - Spinnaker, [315](#)
- SensorTaps_One
 - Spinnaker, [315](#)
- SensorTaps_Ten
 - Spinnaker, [315](#)
- SensorTaps_Three
 - Spinnaker, [315](#)
- SensorTaps_Two
 - Spinnaker, [315](#)
- SensorTapsEnums
 - Spinnaker, [315](#)
- SensorWidth
 - Camera, [546](#)
- Sequencer.cpp
 - AcquireImages, [1472](#)
 - ConfigureSequencerPartOne, [1472](#)
 - ConfigureSequencerPartTwo, [1472](#)
 - main, [1473](#)
 - PrintDeviceInfo, [1473](#)
 - PrintRetrieveNodeFailure, [1473](#)
 - ResetSequencer, [1473](#)
 - RunSingleCamera, [1473](#)
 - SetSingleState, [1473](#)
- SequencerConfigurationMode
 - Camera, [546](#)
- SequencerConfigurationMode_Off
 - Spinnaker, [315](#)
- SequencerConfigurationMode_On
 - Spinnaker, [315](#)
- SequencerConfigurationModeEnums
 - Spinnaker, [315](#)
- SequencerConfigurationValid
 - Camera, [546](#)
- SequencerConfigurationValid_No
 - Spinnaker, [315](#)
- SequencerConfigurationValid_Yes
 - Spinnaker, [315](#)
- SequencerConfigurationValidEnums
 - Spinnaker, [315](#)
- SequencerFeatureEnable
 - Camera, [546](#)
- SequencerMode
 - Camera, [546](#)
- SequencerMode_Off
 - Spinnaker, [316](#)
- SequencerMode_On
 - Spinnaker, [316](#)
- SequencerModeEnums
 - Spinnaker, [315](#)
- SequencerPathSelector
 - Camera, [547](#)
- SequencerSetActive
 - Camera, [547](#)
- SequencerSetLoad
 - Camera, [547](#)
- SequencerSetNext
 - Camera, [547](#)
- SequencerSetSave
 - Camera, [547](#)
- SequencerSetSelector
 - Camera, [548](#)
- SequencerSetStart
 - Camera, [548](#)
- SequencerSetValid
 - Camera, [548](#)
- SequencerSetValid_No
 - Spinnaker, [316](#)
- SequencerSetValid_Yes
 - Spinnaker, [316](#)
- SequencerSetValidEnums
 - Spinnaker, [316](#)
- SequencerTriggerActivation
 - Camera, [548](#)
- SequencerTriggerActivation_AnyEdge
 - Spinnaker, [316](#)
- SequencerTriggerActivation_FallingEdge
 - Spinnaker, [316](#)
- SequencerTriggerActivation_LevelHigh
 - Spinnaker, [316](#)
- SequencerTriggerActivation_LevelLow
 - Spinnaker, [316](#)
- SequencerTriggerActivation_RisingEdge
 - Spinnaker, [316](#)
- SequencerTriggerActivationEnums
 - Spinnaker, [316](#)
- SequencerTriggerSource
 - Camera, [549](#)
- SequencerTriggerSource_FrameStart
 - Spinnaker, [317](#)
- SequencerTriggerSource_Off
 - Spinnaker, [317](#)
- SequencerTriggerSourceEnums
 - Spinnaker, [316](#)
- SERIAL_PORT_BAUD_RATE
 - SerialRxTx.cpp, [1475](#)
- SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND
 - SerialRxTx.cpp, [1475](#)
- SERIAL_PORT_DELAY
 - SerialRxTx.cpp, [1475](#)
- SERIAL_PORT_PARITY_BITS
 - SerialRxTx.cpp, [1475](#)
- SERIAL_PORT_STOP_BITS
 - SerialRxTx.cpp, [1475](#)
- SerialPortBaudRate
 - Camera, [549](#)
- SerialPortBaudRate_Baud115200
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud1200
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud14400
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud19200

- Spinnaker, [317](#)
- SerialPortBaudRate_Baud230400
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud2400
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud300
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud38400
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud460800
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud4800
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud57600
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud600
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud921600
 - Spinnaker, [317](#)
- SerialPortBaudRate_Baud9600
 - Spinnaker, [317](#)
- SerialPortBaudRateEnums
 - Spinnaker, [317](#)
- SerialPortDataBits
 - Camera, [549](#)
- SerialPortParity
 - Camera, [549](#)
- SerialPortParity_Even
 - Spinnaker, [317](#)
- SerialPortParity_Mark
 - Spinnaker, [317](#)
- SerialPortParity_None
 - Spinnaker, [317](#)
- SerialPortParity_Odd
 - Spinnaker, [317](#)
- SerialPortParity_Space
 - Spinnaker, [317](#)
- SerialPortParityEnums
 - Spinnaker, [317](#)
- SerialPortSelector
 - Camera, [549](#)
- SerialPortSelector_SerialPort0
 - Spinnaker, [318](#)
- SerialPortSelectorEnums
 - Spinnaker, [318](#)
- SerialPortSource
 - Camera, [550](#)
- SerialPortSource_Line0
 - Spinnaker, [318](#)
- SerialPortSource_Line1
 - Spinnaker, [318](#)
- SerialPortSource_Line2
 - Spinnaker, [318](#)
- SerialPortSource_Line3
 - Spinnaker, [318](#)
- SerialPortSource_Off
 - Spinnaker, [318](#)
- SerialPortSourceEnums
 - Spinnaker, [318](#)
- SerialPortStopBits
 - Camera, [550](#)
- SerialPortStopBits_Bits1
 - Spinnaker, [318](#)
- SerialPortStopBits_Bits1AndAHalf
 - Spinnaker, [318](#)
- SerialPortStopBits_Bits2
 - Spinnaker, [318](#)
- SerialPortStopBitsEnums
 - Spinnaker, [318](#)
- SerialReceiveFramingErrorCount
 - Camera, [550](#)
- SerialReceiveParityErrorCount
 - Camera, [550](#)
- SerialReceiveQueueClear
 - Camera, [550](#)
- SerialReceiveQueueCurrentCharacterCount
 - Camera, [550](#)
- SerialReceiveQueueMaxCharacterCount
 - Camera, [551](#)
- SerialRx
 - SerialRxTx.cpp, [1476](#)
- SerialRxTx.cpp
 - CleanUp, [1476](#)
 - COM_PORT_COUNT_MAX, [1474](#)
 - ConfigureDevice, [1476](#)
 - DATA_BITS, [1475](#)
 - main, [1476](#)
 - MILLISECOND, [1475](#)
 - PrintDeviceInfo, [1476](#)
 - RunSingleCamera, [1476](#)
 - SERIAL_PORT_BAUD_RATE, [1475](#)
 - SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND, [1475](#)
 - SERIAL_PORT_DELAY, [1475](#)
 - SERIAL_PORT_PARITY_BITS, [1475](#)
 - SERIAL_PORT_STOP_BITS, [1475](#)
 - SerialRx, [1476](#)
 - SerialTx, [1477](#)
 - TWO_SECOND_DELAY, [1475](#)
- SerialTransmitQueueCurrentCharacterCount
 - Camera, [551](#)
- SerialTransmitQueueMaxCharacterCount
 - Camera, [551](#)
- SerialTx
 - SerialRxTx.cpp, [1477](#)
- Set
 - RegisterNode, [1084](#)
- SET_GUID
 - Spinnaker::GenApi, [386](#)
- SetBufferOwnership
 - CameraBase, [579](#)
 - ICameraBase, [805](#)
- SetChannelStatus
 - ImageStatistics, [868](#)
 - ImageStatistics, [949](#)
- SetChunkEnable

- Inference.cpp, 1455
- SetChunks
 - ChunkData, 656
 - ICChunkData, 820
- SetColorProcessing
 - ImageProcessor, 863
 - ImageProcessor, 939
- SetCookie
 - CPortWriteList, 700
 - Spinnaker::GenApi, 386
- SetEnumReference
 - CEnumerationTRef< EnumT >, 616
- SetEventPayload
 - EventHandler, 751
- SetEventType
 - EventHandler, 751
- SetFirst
 - CSelectorSet, 707
- SetFrameRate
 - GigEVisionPerformance.cpp, 1442
- SetGenICamCacheFolder
 - Spinnaker::GenICam, 399
- SetGenICamCLProtocolFolder
 - Spinnaker::GenICam, 399
- SetGenICamLogConfig
 - Spinnaker::GenICam, 399
- SetHeatmapColorGradient
 - ImageUtilityHeatmap, 958
- SetHeatmapRange
 - ImageUtilityHeatmap, 959
- SetInfo
 - CFeatureBag, 635
- SetIntValue
 - EnumNode, 743
 - Spinnaker::GenApi, 386
- SetLoggingEventPriorityLevel
 - ISystem, 1014
 - System, 1106
- SetMaximumFileSize
 - SpinVideo, 1092
- SetMessageCallback
 - SpinUpdate.h, 1354
- SetNext
 - CSelectorSet, 707
 - Spinnaker::GenApi, 387
- SetNodeHandle
 - Node, 1048
- SetNodeMap
 - Node, 1048
- SetNumDecompressionThreads
 - ImageProcessor, 863
 - ImageProcessor, 939
- SetNumEnums
 - CEnumerationTRef< EnumT >, 616
 - Spinnaker::GenApi, 387
- SetPortImpl
 - CChunkPort, 609
 - CEventPort, 632
 - CPortImpl, 697
 - CRegisterPortImpl, 704
 - PortNode, 1069
- SetProgressCallback
 - SpinUpdate.h, 1354
- SetReference
 - BooleanNode, 414
 - CategoryNode, 592
 - CEnumerationTRef< EnumT >, 616
 - CommandNode, 686
 - EnumEntryNode, 738
 - EnumNode, 743
 - FloatNode, 770
 - FloatRegNode, 773
 - IntegerNode, 983
 - IntRegNode, 1010
 - Node, 1048
 - PortNode, 1070
 - PortRecorder, 1074
 - PortReplay, 1078
 - RegisterNode, 1084
 - StringNode, 1096
 - StringRegNode, 1099
 - ValueNode, 1163
- SetSelectedCameraIndex
 - SpinnakerDirectShow.h, 1349
- SetSingleState
 - Sequencer.cpp, 1473
- SetupCounterAndTimer
 - CounterAndTimer.cpp, 1425
- SetUserBuffers
 - CameraBase, 579, 580
 - ICameraBase, 805
- SetValue
 - BooleanNode, 415
 - CEnumerationTRef< EnumT >, 617
 - FloatNode, 770
 - IntegerNode, 983
 - StringNode, 1096
- Sharpening
 - Camera, 551
- SharpeningAuto
 - Camera, 551
- SharpeningEnable
 - Camera, 552
- SharpeningThreshold
 - Camera, 552
- Signed
 - Spinnaker::GenApi, 360
- SingleChunkData_t, 1085
 - ChunkID, 1085
 - ChunkLength, 1085
 - ChunkOffset, 1085
- SingleChunkDataStr_t, 1086
 - ChunkID, 1086
 - ChunkLength, 1086
 - ChunkOffset, 1086
- SIOption, 1087

- reserved, [1087](#)
- SIOption, [1087](#)
- size
 - double_autovector_t, [727](#)
 - gcstring, [783](#)
 - int64_autovector_t, [977](#)
- SleepyWrapper
 - AcquisitionMultipleCameraRecovery.cpp, [1409](#)
 - ActionCommand.cpp, [1416](#)
 - BufferHandling.cpp, [1418](#)
 - ImageEvents.cpp, [1447](#)
- SOFTWARE
 - Trigger.cpp, [1479](#)
 - Trigger_QuickSpin.cpp, [1481](#)
- SoftwareSignalPulse
 - Camera, [552](#)
- SoftwareSignalSelector
 - Camera, [552](#)
- SoftwareSignalSelector_SoftwareSignal0
 - Spinnaker, [319](#)
- SoftwareSignalSelector_SoftwareSignal1
 - Spinnaker, [319](#)
- SoftwareSignalSelector_SoftwareSignal2
 - Spinnaker, [319](#)
- SoftwareSignalSelectorEnums
 - Spinnaker, [318](#)
- SourceCount
 - Camera, [553](#)
- SourceDataRange
 - Spinnaker, [319](#)
- SourceSelector
 - Camera, [553](#)
- SourceSelector_All
 - Spinnaker, [320](#)
- SourceSelector_Source0
 - Spinnaker, [320](#)
- SourceSelector_Source1
 - Spinnaker, [320](#)
- SourceSelector_Source2
 - Spinnaker, [320](#)
- SourceSelectorEnums
 - Spinnaker, [319](#)
- SPECIFIC
 - DeviceEvents.cpp, [1426](#)
- Spinnaker, [192](#)
 - AcquisitionMode_Continuous, [234](#)
 - AcquisitionMode_MultiFrame, [234](#)
 - AcquisitionMode_SingleFrame, [234](#)
 - AcquisitionModeEnums, [233](#)
 - AcquisitionStatusSelector_AcquisitionActive, [234](#)
 - AcquisitionStatusSelector_AcquisitionTransfer, [234](#)
 - AcquisitionStatusSelector_AcquisitionTriggerWait, [234](#)
 - AcquisitionStatusSelector_ExposureActive, [234](#)
 - AcquisitionStatusSelector_FrameActive, [234](#)
 - AcquisitionStatusSelector_FrameTriggerWait, [234](#)
 - AcquisitionStatusSelectorEnums, [234](#)
 - ActionCommandStatus, [234](#)
 - ActionUnconditionalMode_Off, [235](#)
 - ActionUnconditionalMode_On, [235](#)
 - ActionUnconditionalModeEnums, [235](#)
 - AdcBitDepth_Bit10, [235](#)
 - AdcBitDepth_Bit12, [235](#)
 - AdcBitDepth_Bit14, [235](#)
 - AdcBitDepth_Bit8, [235](#)
 - AdcBitDepthEnums, [235](#)
 - AutoAlgorithmSelector_Ae, [235](#)
 - AutoAlgorithmSelector_Awb, [235](#)
 - AutoAlgorithmSelectorEnums, [235](#)
 - AutoExposureControlPriority_ExposureTime, [236](#)
 - AutoExposureControlPriority_Gain, [236](#)
 - AutoExposureControlPriorityEnums, [235](#)
 - AutoExposureLightingMode_AutoDetect, [236](#)
 - AutoExposureLightingMode_Backlight, [236](#)
 - AutoExposureLightingMode_Frontlight, [236](#)
 - AutoExposureLightingMode_Normal, [236](#)
 - AutoExposureLightingModeEnums, [236](#)
 - AutoExposureMeteringMode_Average, [237](#)
 - AutoExposureMeteringMode_CenterWeighted, [237](#)
 - AutoExposureMeteringMode_HistogramPeak, [237](#)
 - AutoExposureMeteringMode_Partial, [237](#)
 - AutoExposureMeteringMode_Spot, [237](#)
 - AutoExposureMeteringModeEnums, [236](#)
 - AutoExposureTargetGreyValueAuto_Continuous, [237](#)
 - AutoExposureTargetGreyValueAuto_Off, [237](#)
 - AutoExposureTargetGreyValueAutoEnums, [237](#)
 - BalanceRatioSelector_Blue, [237](#)
 - BalanceRatioSelector_Red, [237](#)
 - BalanceRatioSelectorEnums, [237](#)
 - BalanceWhiteAuto_Continuous, [238](#)
 - BalanceWhiteAuto_Off, [238](#)
 - BalanceWhiteAuto_Once, [238](#)
 - BalanceWhiteAutoEnums, [237](#)
 - BalanceWhiteAutoProfile_Indoor, [238](#)
 - BalanceWhiteAutoProfile_Outdoor, [238](#)
 - BalanceWhiteAutoProfileEnums, [238](#)
 - BinningHorizontalMode_Average, [238](#)
 - BinningHorizontalMode_Sum, [238](#)
 - BinningHorizontalModeEnums, [238](#)
 - BinningSelector_All, [239](#)
 - BinningSelector_ISP, [239](#)
 - BinningSelector_Sensor, [239](#)
 - BinningSelectorEnums, [238](#)
 - BinningVerticalMode_Average, [239](#)
 - BinningVerticalMode_Sum, [239](#)
 - BinningVerticalModeEnums, [239](#)
 - BlackLevelAuto_Continuous, [240](#)
 - BlackLevelAuto_Off, [240](#)
 - BlackLevelAuto_Once, [240](#)
 - BlackLevelAutoBalance_Continuous, [239](#)
 - BlackLevelAutoBalance_Off, [239](#)
 - BlackLevelAutoBalance_Once, [239](#)
 - BlackLevelAutoBalanceEnums, [239](#)
 - BlackLevelAutoEnums, [239](#)

- BlackLevelSelector_All, [240](#)
- BlackLevelSelector_Analog, [240](#)
- BlackLevelSelector_Digital, [240](#)
- BlackLevelSelectorEnums, [240](#)
- BufferOwnership, [240](#)
- CCMApplication, [240](#)
- CCMColorSpace, [241](#)
- CCMColorTemperature, [241](#)
- CCMSensor, [241](#)
- CCMType, [241](#)
- ChunkBlackLevelSelector_All, [242](#)
- ChunkBlackLevelSelectorEnums, [242](#)
- ChunkCounterSelector_Counter0, [242](#)
- ChunkCounterSelector_Counter1, [242](#)
- ChunkCounterSelector_Counter2, [242](#)
- ChunkCounterSelectorEnums, [242](#)
- ChunkEncoderSelector_Encoder0, [242](#)
- ChunkEncoderSelector_Encoder1, [243](#)
- ChunkEncoderSelector_Encoder2, [243](#)
- ChunkEncoderSelectorEnums, [242](#)
- ChunkEncoderStatus_EncoderDown, [243](#)
- ChunkEncoderStatus_EncoderIdle, [243](#)
- ChunkEncoderStatus_EncoderStatic, [243](#)
- ChunkEncoderStatus_EncoderUp, [243](#)
- ChunkEncoderStatusEnums, [243](#)
- ChunkExposureTimeSelector_Blue, [243](#)
- ChunkExposureTimeSelector_Common, [243](#)
- ChunkExposureTimeSelector_Cyan, [243](#)
- ChunkExposureTimeSelector_Green, [243](#)
- ChunkExposureTimeSelector_Infrared, [243](#)
- ChunkExposureTimeSelector_Magenta, [243](#)
- ChunkExposureTimeSelector_Red, [243](#)
- ChunkExposureTimeSelector_Stage1, [243](#)
- ChunkExposureTimeSelector_Stage2, [243](#)
- ChunkExposureTimeSelector_Ultraviolet, [243](#)
- ChunkExposureTimeSelector_Yellow, [243](#)
- ChunkExposureTimeSelectorEnums, [243](#)
- ChunkGainSelector_All, [244](#)
- ChunkGainSelector_Blue, [244](#)
- ChunkGainSelector_Green, [244](#)
- ChunkGainSelector_Red, [244](#)
- ChunkGainSelectorEnums, [243](#)
- ChunkImageComponent_Color, [244](#)
- ChunkImageComponent_Confidence, [244](#)
- ChunkImageComponent_Disparity, [244](#)
- ChunkImageComponent_Infrared, [244](#)
- ChunkImageComponent_Intensity, [244](#)
- ChunkImageComponent_Range, [244](#)
- ChunkImageComponent_Scatter, [244](#)
- ChunkImageComponent_Ultraviolet, [244](#)
- ChunkImageComponentEnums, [244](#)
- ChunkPixelFormat_BayerBG8, [245](#)
- ChunkPixelFormat_BayerGB8, [245](#)
- ChunkPixelFormat_BayerGR8, [245](#)
- ChunkPixelFormat_BayerRG8, [245](#)
- ChunkPixelFormat_Mono12Packed, [244](#)
- ChunkPixelFormat_Mono16, [244](#)
- ChunkPixelFormat_Mono8, [244](#)
- ChunkPixelFormat_RGB8Packed, [244](#)
- ChunkPixelFormat_YCbCr601_422_8_CbYCrY, [245](#)
- ChunkPixelFormat_YUV422Packed, [245](#)
- ChunkPixelFormatEnums, [244](#)
- ChunkRegionID_Region0, [245](#)
- ChunkRegionID_Region1, [245](#)
- ChunkRegionID_Region2, [245](#)
- ChunkRegionIDEnums, [245](#)
- ChunkScan3dCoordinateReferenceSelector_RotationX, [245](#)
- ChunkScan3dCoordinateReferenceSelector_RotationY, [245](#)
- ChunkScan3dCoordinateReferenceSelector_RotationZ, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationX, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationY, [245](#)
- ChunkScan3dCoordinateReferenceSelector_TranslationZ, [245](#)
- ChunkScan3dCoordinateReferenceSelectorEnums, [245](#)
- ChunkScan3dCoordinateSelector_CoordinateA, [246](#)
- ChunkScan3dCoordinateSelector_CoordinateB, [246](#)
- ChunkScan3dCoordinateSelector_CoordinateC, [246](#)
- ChunkScan3dCoordinateSelectorEnums, [245](#)
- ChunkScan3dCoordinateSystem_Cartesian, [246](#)
- ChunkScan3dCoordinateSystem_Cylindrical, [246](#)
- ChunkScan3dCoordinateSystem_Spherical, [246](#)
- ChunkScan3dCoordinateSystemEnums, [246](#)
- ChunkScan3dCoordinateSystemReference_Anchor, [246](#)
- ChunkScan3dCoordinateSystemReference_Transformed, [246](#)
- ChunkScan3dCoordinateSystemReferenceEnums, [246](#)
- ChunkScan3dCoordinateTransformSelector_RotationX, [247](#)
- ChunkScan3dCoordinateTransformSelector_RotationY, [247](#)
- ChunkScan3dCoordinateTransformSelector_RotationZ, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationX, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationY, [247](#)
- ChunkScan3dCoordinateTransformSelector_TranslationZ, [247](#)
- ChunkScan3dCoordinateTransformSelectorEnums, [246](#)
- ChunkScan3dDistanceUnit_Inch, [247](#)
- ChunkScan3dDistanceUnit_Millimeter, [247](#)
- ChunkScan3dDistanceUnitEnums, [247](#)
- ChunkScan3dOutputMode_CalibratedABC_Grid,

- [247](#)
- [ChunkScan3dOutputMode_CalibratedABC_PointCloud, 247](#)
- [ChunkScan3dOutputMode_CalibratedAC, 248](#)
- [ChunkScan3dOutputMode_CalibratedAC_Linescan, 248](#)
- [ChunkScan3dOutputMode_CalibratedC, 248](#)
- [ChunkScan3dOutputMode_CalibratedC_Linescan, 248](#)
- [ChunkScan3dOutputMode_DisparityC, 248](#)
- [ChunkScan3dOutputMode_DisparityC_Linescan, 248](#)
- [ChunkScan3dOutputMode_RectifiedC, 248](#)
- [ChunkScan3dOutputMode_RectifiedC_Linescan, 248](#)
- [ChunkScan3dOutputMode_UncalibratedC, 247](#)
- [ChunkScan3dOutputModeEnums, 247](#)
- [ChunkSelector_BlackLevel, 249](#)
- [ChunkSelector_CRC, 248](#)
- [ChunkSelector_ExposureEndLineStatusAll, 249](#)
- [ChunkSelector_ExposureTime, 248](#)
- [ChunkSelector_FrameID, 248](#)
- [ChunkSelector_Gain, 249](#)
- [ChunkSelector_Height, 248](#)
- [ChunkSelector_Image, 248](#)
- [ChunkSelector_OffsetX, 248](#)
- [ChunkSelector_OffsetY, 248](#)
- [ChunkSelector_PixelFormat, 249](#)
- [ChunkSelector_SequencerSetActive, 249](#)
- [ChunkSelector_SerialData, 249](#)
- [ChunkSelector_Timestamp, 249](#)
- [ChunkSelector_Width, 248](#)
- [ChunkSelectorEnums, 248](#)
- [ChunkSourceID_Source0, 249](#)
- [ChunkSourceID_Source1, 249](#)
- [ChunkSourceID_Source2, 249](#)
- [ChunkSourceIDEnums, 249](#)
- [ChunkTimerSelector_Timer0, 249](#)
- [ChunkTimerSelector_Timer1, 249](#)
- [ChunkTimerSelector_Timer2, 249](#)
- [ChunkTimerSelectorEnums, 249](#)
- [ChunkTransferStreamID_Stream0, 250](#)
- [ChunkTransferStreamID_Stream1, 250](#)
- [ChunkTransferStreamID_Stream2, 250](#)
- [ChunkTransferStreamID_Stream3, 250](#)
- [ChunkTransferStreamIDEnums, 249](#)
- [CIConfiguration_Base, 250](#)
- [CIConfiguration_DualBase, 250](#)
- [CIConfiguration_EightyBit, 250](#)
- [CIConfiguration_Full, 250](#)
- [CIConfiguration_Medium, 250](#)
- [CIConfigurationEnums, 250](#)
- [CITimeSlotsCount_One, 250](#)
- [CITimeSlotsCount_Three, 250](#)
- [CITimeSlotsCount_Two, 250](#)
- [CITimeSlotsCountEnums, 250](#)
- [ColorProcessingAlgorithm, 250](#)
- [ColorTransformationSelector_RGBtoRGB, 251](#)
- [ColorTransformationSelector_RGBtoYUV, 251](#)
- [ColorTransformationSelectorEnums, 251](#)
- [ColorTransformationValueSelector_Gain00, 252](#)
- [ColorTransformationValueSelector_Gain01, 252](#)
- [ColorTransformationValueSelector_Gain02, 252](#)
- [ColorTransformationValueSelector_Gain10, 252](#)
- [ColorTransformationValueSelector_Gain11, 252](#)
- [ColorTransformationValueSelector_Gain12, 252](#)
- [ColorTransformationValueSelector_Gain20, 252](#)
- [ColorTransformationValueSelector_Gain21, 252](#)
- [ColorTransformationValueSelector_Gain22, 252](#)
- [ColorTransformationValueSelector_Offset0, 252](#)
- [ColorTransformationValueSelector_Offset1, 252](#)
- [ColorTransformationValueSelector_Offset2, 252](#)
- [ColorTransformationValueSelectorEnums, 251](#)
- [CompressionSaturationPriority_DropFrame, 252](#)
- [CompressionSaturationPriority_ReduceFrameRate, 252](#)
- [CompressionSaturationPriorityEnums, 252](#)
- [CounterEventActivation_AnyEdge, 253](#)
- [CounterEventActivation_FallingEdge, 253](#)
- [CounterEventActivation_LevelHigh, 253](#)
- [CounterEventActivation_LevelLow, 253](#)
- [CounterEventActivation_RisingEdge, 253](#)
- [CounterEventActivationEnums, 252](#)
- [CounterEventSource_Counter0End, 253](#)
- [CounterEventSource_Counter0Start, 253](#)
- [CounterEventSource_Counter1End, 253](#)
- [CounterEventSource_Counter1Start, 253](#)
- [CounterEventSource_ExposureEnd, 253](#)
- [CounterEventSource_ExposureStart, 253](#)
- [CounterEventSource_FrameTriggerWait, 253](#)
- [CounterEventSource_Line0, 253](#)
- [CounterEventSource_Line1, 253](#)
- [CounterEventSource_Line2, 253](#)
- [CounterEventSource_Line3, 253](#)
- [CounterEventSource_LogicBlock0, 253](#)
- [CounterEventSource_LogicBlock1, 253](#)
- [CounterEventSource_MHzTick, 253](#)
- [CounterEventSource_Off, 253](#)
- [CounterEventSource_UserOutput0, 253](#)
- [CounterEventSource_UserOutput1, 253](#)
- [CounterEventSource_UserOutput2, 253](#)
- [CounterEventSource_UserOutput3, 253](#)
- [CounterEventSourceEnums, 253](#)
- [CounterResetActivation_AnyEdge, 254](#)
- [CounterResetActivation_FallingEdge, 254](#)
- [CounterResetActivation_LevelHigh, 254](#)
- [CounterResetActivation_LevelLow, 254](#)
- [CounterResetActivation_RisingEdge, 254](#)
- [CounterResetActivationEnums, 253](#)
- [CounterResetSource_Counter0End, 254](#)
- [CounterResetSource_Counter0Start, 254](#)
- [CounterResetSource_Counter1End, 254](#)
- [CounterResetSource_Counter1Start, 254](#)
- [CounterResetSource_ExposureEnd, 254](#)
- [CounterResetSource_ExposureStart, 254](#)
- [CounterResetSource_FrameTriggerWait, 254](#)

- CounterResetSource_Line0, [254](#)
- CounterResetSource_Line1, [254](#)
- CounterResetSource_Line2, [254](#)
- CounterResetSource_Line3, [254](#)
- CounterResetSource_LogicBlock0, [254](#)
- CounterResetSource_LogicBlock1, [254](#)
- CounterResetSource_Off, [254](#)
- CounterResetSource_UserOutput0, [254](#)
- CounterResetSource_UserOutput1, [254](#)
- CounterResetSource_UserOutput2, [254](#)
- CounterResetSource_UserOutput3, [254](#)
- CounterResetSourceEnums, [254](#)
- CounterSelector_Counter0, [255](#)
- CounterSelector_Counter1, [255](#)
- CounterSelectorEnums, [254](#)
- CounterStatus_CounterActive, [255](#)
- CounterStatus_CounterCompleted, [255](#)
- CounterStatus_CounterIdle, [255](#)
- CounterStatus_CounterOverflow, [255](#)
- CounterStatus_CounterTriggerWait, [255](#)
- CounterStatusEnums, [255](#)
- CounterTriggerActivation_AnyEdge, [255](#)
- CounterTriggerActivation_FallingEdge, [255](#)
- CounterTriggerActivation_LevelHigh, [255](#)
- CounterTriggerActivation_LevelLow, [255](#)
- CounterTriggerActivation_RisingEdge, [255](#)
- CounterTriggerActivationEnums, [255](#)
- CounterTriggerSource_Counter0End, [256](#)
- CounterTriggerSource_Counter0Start, [256](#)
- CounterTriggerSource_Counter1End, [256](#)
- CounterTriggerSource_Counter1Start, [256](#)
- CounterTriggerSource_ExposureEnd, [256](#)
- CounterTriggerSource_ExposureStart, [256](#)
- CounterTriggerSource_FrameTriggerWait, [256](#)
- CounterTriggerSource_Line0, [256](#)
- CounterTriggerSource_Line1, [256](#)
- CounterTriggerSource_Line2, [256](#)
- CounterTriggerSource_Line3, [256](#)
- CounterTriggerSource_LogicBlock0, [256](#)
- CounterTriggerSource_LogicBlock1, [256](#)
- CounterTriggerSource_Off, [256](#)
- CounterTriggerSource_UserOutput0, [256](#)
- CounterTriggerSource_UserOutput1, [256](#)
- CounterTriggerSource_UserOutput2, [256](#)
- CounterTriggerSource_UserOutput3, [256](#)
- CounterTriggerSourceEnums, [255](#)
- CxpConnectionTestMode_Mode1, [256](#)
- CxpConnectionTestMode_Off, [256](#)
- CxpConnectionTestModeEnums, [256](#)
- CxpLinkConfiguration_Auto, [257](#)
- CxpLinkConfiguration_CXP1_X1, [257](#)
- CxpLinkConfiguration_CXP1_X2, [257](#)
- CxpLinkConfiguration_CXP1_X3, [257](#)
- CxpLinkConfiguration_CXP1_X4, [257](#)
- CxpLinkConfiguration_CXP1_X5, [257](#)
- CxpLinkConfiguration_CXP1_X6, [257](#)
- CxpLinkConfiguration_CXP2_X1, [257](#)
- CxpLinkConfiguration_CXP2_X2, [257](#)
- CxpLinkConfiguration_CXP2_X3, [257](#)
- CxpLinkConfiguration_CXP2_X4, [257](#)
- CxpLinkConfiguration_CXP2_X5, [257](#)
- CxpLinkConfiguration_CXP2_X6, [257](#)
- CxpLinkConfiguration_CXP3_X1, [257](#)
- CxpLinkConfiguration_CXP3_X2, [257](#)
- CxpLinkConfiguration_CXP3_X3, [257](#)
- CxpLinkConfiguration_CXP3_X4, [257](#)
- CxpLinkConfiguration_CXP3_X5, [257](#)
- CxpLinkConfiguration_CXP3_X6, [257](#)
- CxpLinkConfiguration_CXP5_X1, [257](#)
- CxpLinkConfiguration_CXP5_X2, [257](#)
- CxpLinkConfiguration_CXP5_X3, [257](#)
- CxpLinkConfiguration_CXP5_X4, [257](#)
- CxpLinkConfiguration_CXP5_X5, [257](#)
- CxpLinkConfiguration_CXP5_X6, [257](#)
- CxpLinkConfiguration_CXP6_X1, [257](#)
- CxpLinkConfiguration_CXP6_X2, [257](#)
- CxpLinkConfiguration_CXP6_X3, [257](#)
- CxpLinkConfiguration_CXP6_X4, [257](#)
- CxpLinkConfiguration_CXP6_X5, [257](#)
- CxpLinkConfiguration_CXP6_X6, [257](#)
- CxpLinkConfigurationEnums, [256](#)
- CxpLinkConfigurationPreferred_CXP1_X1, [257](#)
- CxpLinkConfigurationPreferred_CXP1_X2, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X3, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X4, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X5, [258](#)
- CxpLinkConfigurationPreferred_CXP1_X6, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X1, [257](#)
- CxpLinkConfigurationPreferred_CXP2_X2, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X3, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X4, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X5, [258](#)
- CxpLinkConfigurationPreferred_CXP2_X6, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X1, [257](#)
- CxpLinkConfigurationPreferred_CXP3_X2, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X3, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X4, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X5, [258](#)
- CxpLinkConfigurationPreferred_CXP3_X6, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X1, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X2, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X3, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X4, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X5, [258](#)
- CxpLinkConfigurationPreferred_CXP5_X6, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X1, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X2, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X3, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X4, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X5, [258](#)
- CxpLinkConfigurationPreferred_CXP6_X6, [258](#)
- CxpLinkConfigurationPreferredEnums, [257](#)
- CxpLinkConfigurationStatus_CXP1_X1, [258](#)
- CxpLinkConfigurationStatus_CXP1_X2, [259](#)
- CxpLinkConfigurationStatus_CXP1_X3, [259](#)
- CxpLinkConfigurationStatus_CXP1_X4, [259](#)

- CxpLinkConfigurationStatus_CXP1_X5, 259
- CxpLinkConfigurationStatus_CXP1_X6, 259
- CxpLinkConfigurationStatus_CXP2_X1, 258
- CxpLinkConfigurationStatus_CXP2_X2, 259
- CxpLinkConfigurationStatus_CXP2_X3, 259
- CxpLinkConfigurationStatus_CXP2_X4, 259
- CxpLinkConfigurationStatus_CXP2_X5, 259
- CxpLinkConfigurationStatus_CXP2_X6, 259
- CxpLinkConfigurationStatus_CXP3_X1, 258
- CxpLinkConfigurationStatus_CXP3_X2, 259
- CxpLinkConfigurationStatus_CXP3_X3, 259
- CxpLinkConfigurationStatus_CXP3_X4, 259
- CxpLinkConfigurationStatus_CXP3_X5, 259
- CxpLinkConfigurationStatus_CXP3_X6, 259
- CxpLinkConfigurationStatus_CXP5_X1, 259
- CxpLinkConfigurationStatus_CXP5_X2, 259
- CxpLinkConfigurationStatus_CXP5_X3, 259
- CxpLinkConfigurationStatus_CXP5_X4, 259
- CxpLinkConfigurationStatus_CXP5_X5, 259
- CxpLinkConfigurationStatus_CXP5_X6, 259
- CxpLinkConfigurationStatus_CXP6_X1, 259
- CxpLinkConfigurationStatus_CXP6_X2, 259
- CxpLinkConfigurationStatus_CXP6_X3, 259
- CxpLinkConfigurationStatus_CXP6_X4, 259
- CxpLinkConfigurationStatus_CXP6_X5, 259
- CxpLinkConfigurationStatus_CXP6_X6, 259
- CxpLinkConfigurationStatus_None, 258
- CxpLinkConfigurationStatus_Pending, 258
- CxpLinkConfigurationStatusEnums, 258
- CxpPoCxpStatus_Auto, 259
- CxpPoCxpStatus_Off, 259
- CxpPoCxpStatus_Tripped, 259
- CxpPoCxpStatusEnums, 259
- DecimationHorizontalMode_Discard, 260
- DecimationHorizontalModeEnums, 259
- DecimationSelector_All, 260
- DecimationSelector_Sensor, 260
- DecimationSelectorEnums, 260
- DecimationVerticalMode_Discard, 260
- DecimationVerticalModeEnums, 260
- DefectCorrectionMode_Average, 261
- DefectCorrectionMode_Highlight, 261
- DefectCorrectionMode_Zero, 261
- DefectCorrectionModeEnums, 260
- Deinterlacing_LineDuplication, 261
- Deinterlacing_Off, 261
- Deinterlacing_Weave, 261
- DeinterlacingEnums, 261
- DEPRECATED_CLASS, 336
- DeviceAccessStatus_Busy, 261
- DeviceAccessStatus_NoAccess, 261
- DeviceAccessStatus_OpenReadOnly, 261
- DeviceAccessStatus_OpenReadWrite, 261
- DeviceAccessStatus_ReadOnly, 261
- DeviceAccessStatus_ReadWrite, 261
- DeviceAccessStatus_Unknown, 261
- DeviceAccessStatusEnum, 261
- DeviceCharacterSet_ASCII, 262
- DeviceCharacterSet_UTF8, 262
- DeviceCharacterSetEnums, 261
- DeviceClockSelector_CameraLink, 262
- DeviceClockSelector_Sensor, 262
- DeviceClockSelector_SensorDigitization, 262
- DeviceClockSelectorEnums, 262
- DeviceConnectionStatus_Active, 262
- DeviceConnectionStatus_Inactive, 262
- DeviceConnectionStatusEnums, 262
- DeviceCurrentSpeed_FullSpeed, 263
- DeviceCurrentSpeed_HighSpeed, 263
- DeviceCurrentSpeed_LowSpeed, 263
- DeviceCurrentSpeed_SuperSpeed, 263
- DeviceCurrentSpeed_UnknownSpeed, 263
- DeviceCurrentSpeedEnum, 262
- DeviceEndianessMechanism_Legacy, 263
- DeviceEndianessMechanism_Standard, 263
- DeviceEndianessMechanismEnum, 263
- DeviceIndicatorMode_Active, 263
- DeviceIndicatorMode_ErrorStatus, 263
- DeviceIndicatorMode_Inactive, 263
- DeviceIndicatorModeEnums, 263
- DeviceLinkHeartbeatMode_Off, 264
- DeviceLinkHeartbeatMode_On, 264
- DeviceLinkHeartbeatModeEnums, 263
- DeviceLinkThroughputLimitMode_Off, 264
- DeviceLinkThroughputLimitMode_On, 264
- DeviceLinkThroughputLimitModeEnums, 264
- DevicePowerSupplySelector_External, 264
- DevicePowerSupplySelectorEnums, 264
- DeviceRegistersEndianness_Big, 264
- DeviceRegistersEndianness_Little, 264
- DeviceRegistersEndiannessEnums, 264
- DeviceScanType_Areascan, 265
- DeviceScanTypeEnums, 265
- DeviceSerialPortBaudRate_Baud115200, 265
- DeviceSerialPortBaudRate_Baud19200, 265
- DeviceSerialPortBaudRate_Baud230400, 265
- DeviceSerialPortBaudRate_Baud38400, 265
- DeviceSerialPortBaudRate_Baud460800, 265
- DeviceSerialPortBaudRate_Baud57600, 265
- DeviceSerialPortBaudRate_Baud921600, 265
- DeviceSerialPortBaudRate_Baud9600, 265
- DeviceSerialPortBaudRateEnums, 265
- DeviceSerialPortSelector_CameraLink, 265
- DeviceSerialPortSelectorEnums, 265
- DeviceStreamChannelEndianness_Big, 266
- DeviceStreamChannelEndianness_Little, 266
- DeviceStreamChannelEndiannessEnums, 265
- DeviceStreamChannelType_Receiver, 266
- DeviceStreamChannelType_Transmitter, 266
- DeviceStreamChannelTypeEnums, 266
- DeviceTapGeometry_Geometry_10X, 267
- DeviceTapGeometry_Geometry_10X_1Y, 267
- DeviceTapGeometry_Geometry_1X, 267
- DeviceTapGeometry_Geometry_1X10, 267
- DeviceTapGeometry_Geometry_1X10_1Y, 267
- DeviceTapGeometry_Geometry_1X2, 267

- DeviceTapGeometry_Geometry_1X2_1Y, [266](#)
- DeviceTapGeometry_Geometry_1X2_1Y2, [266](#)
- DeviceTapGeometry_Geometry_1X2_2YE, [267](#)
- DeviceTapGeometry_Geometry_1X3, [267](#)
- DeviceTapGeometry_Geometry_1X3_1Y, [266](#)
- DeviceTapGeometry_Geometry_1X4, [267](#)
- DeviceTapGeometry_Geometry_1X4_1Y, [267](#)
- DeviceTapGeometry_Geometry_1X8, [267](#)
- DeviceTapGeometry_Geometry_1X8_1Y, [267](#)
- DeviceTapGeometry_Geometry_1X_1Y, [266](#)
- DeviceTapGeometry_Geometry_1X_1Y2, [266](#)
- DeviceTapGeometry_Geometry_1X_2YE, [266](#)
- DeviceTapGeometry_Geometry_2X, [267](#)
- DeviceTapGeometry_Geometry_2X2, [267](#)
- DeviceTapGeometry_Geometry_2X2_1Y, [267](#)
- DeviceTapGeometry_Geometry_2X2E, [267](#)
- DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M, [267](#)
- DeviceTapGeometry_Geometry_2X2E_2YE, [267](#)
- DeviceTapGeometry_Geometry_2X2M, [267](#)
- DeviceTapGeometry_Geometry_2X_1Y, [266](#)
- DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y, [266](#)
- DeviceTapGeometry_Geometry_2X_2YE, [267](#)
- DeviceTapGeometry_Geometry_2XE, [267](#)
- DeviceTapGeometry_Geometry_2XE_1Y2, [266](#)
- DeviceTapGeometry_Geometry_2XE_2YE, [267](#)
- DeviceTapGeometry_Geometry_2XM, [267](#)
- DeviceTapGeometry_Geometry_2XM_1Y, [266](#)
- DeviceTapGeometry_Geometry_2XM_1Y2, [266](#)
- DeviceTapGeometry_Geometry_2XM_2YE, [267](#)
- DeviceTapGeometry_Geometry_3X, [267](#)
- DeviceTapGeometry_Geometry_3X_1Y, [267](#)
- DeviceTapGeometry_Geometry_4X, [267](#)
- DeviceTapGeometry_Geometry_4X2, [267](#)
- DeviceTapGeometry_Geometry_4X2_1Y, [267](#)
- DeviceTapGeometry_Geometry_4X2E, [267](#)
- DeviceTapGeometry_Geometry_4X2E_1Y, [267](#)
- DeviceTapGeometry_Geometry_4X_1Y, [267](#)
- DeviceTapGeometry_Geometry_8X, [267](#)
- DeviceTapGeometry_Geometry_8X_1Y, [267](#)
- DeviceTapGeometryEnums, [266](#)
- DeviceTemperatureSelector_Sensor, [268](#)
- DeviceTemperatureSelectorEnums, [267](#)
- DeviceTLType_CameraLink, [268](#)
- DeviceTLType_CameraLinkHS, [268](#)
- DeviceTLType_CoaxPress, [268](#)
- DeviceTLType_Custom, [268](#)
- DeviceTLType_GigEVision, [268](#)
- DeviceTLType_USB3Vision, [268](#)
- DeviceTLTypeEnums, [268](#)
- DeviceType_CameraLink, [268](#)
- DeviceType_CameraLinkHS, [268](#)
- DeviceType_CoaxPress, [268](#)
- DeviceType_Custom, [268](#)
- DeviceType_GigEVision, [268](#)
- DeviceType_Peripheral, [270](#)
- DeviceType_Receiver, [270](#)
- DeviceType_Transceiver, [270](#)
- DeviceType_Transmitter, [270](#)
- DeviceType_USB3Vision, [268](#)
- DeviceTypeEnum, [268](#)
- DeviceTypeEnums, [268](#)
- EncoderMode_FourPhase, [270](#)
- EncoderMode_HighResolution, [270](#)
- EncoderModeEnums, [270](#)
- EncoderOutputMode_DirectionDown, [270](#)
- EncoderOutputMode_DirectionUp, [270](#)
- EncoderOutputMode_Motion, [270](#)
- EncoderOutputMode_Off, [270](#)
- EncoderOutputMode_PositionDown, [270](#)
- EncoderOutputMode_PositionUp, [270](#)
- EncoderOutputModeEnums, [270](#)
- EncoderResetActivation_AnyEdge, [271](#)
- EncoderResetActivation_FallingEdge, [271](#)
- EncoderResetActivation_LevelHigh, [271](#)
- EncoderResetActivation_LevelLow, [271](#)
- EncoderResetActivation_RisingEdge, [271](#)
- EncoderResetActivationEnums, [271](#)
- EncoderResetSource_AcquisitionEnd, [271](#)
- EncoderResetSource_AcquisitionStart, [271](#)
- EncoderResetSource_AcquisitionTrigger, [271](#)
- EncoderResetSource_Action0, [272](#)
- EncoderResetSource_Action1, [272](#)
- EncoderResetSource_Action2, [272](#)
- EncoderResetSource_Counter0End, [271](#)
- EncoderResetSource_Counter0Start, [271](#)
- EncoderResetSource_Counter1End, [271](#)
- EncoderResetSource_Counter1Start, [271](#)
- EncoderResetSource_Counter2End, [271](#)
- EncoderResetSource_Counter2Start, [271](#)
- EncoderResetSource_ExposureEnd, [271](#)
- EncoderResetSource_ExposureStart, [271](#)
- EncoderResetSource_FrameEnd, [271](#)
- EncoderResetSource_FrameStart, [271](#)
- EncoderResetSource_FrameTrigger, [271](#)
- EncoderResetSource_Line0, [271](#)
- EncoderResetSource_Line1, [271](#)
- EncoderResetSource_Line2, [271](#)
- EncoderResetSource_LinkTrigger0, [272](#)
- EncoderResetSource_LinkTrigger1, [272](#)
- EncoderResetSource_LinkTrigger2, [272](#)
- EncoderResetSource_Off, [271](#)
- EncoderResetSource_SoftwareSignal0, [272](#)
- EncoderResetSource_SoftwareSignal1, [272](#)
- EncoderResetSource_SoftwareSignal2, [272](#)
- EncoderResetSource_Timer0End, [272](#)
- EncoderResetSource_Timer0Start, [271](#)
- EncoderResetSource_Timer1End, [272](#)
- EncoderResetSource_Timer1Start, [271](#)
- EncoderResetSource_Timer2End, [272](#)
- EncoderResetSource_Timer2Start, [271](#)
- EncoderResetSource_UserOutput0, [272](#)
- EncoderResetSource_UserOutput1, [272](#)
- EncoderResetSource_UserOutput2, [272](#)
- EncoderResetSourceEnums, [271](#)

- EncoderSelector_Encoder0, [272](#)
- EncoderSelector_Encoder1, [272](#)
- EncoderSelector_Encoder2, [272](#)
- EncoderSelectorEnums, [272](#)
- EncoderSourceA_Line0, [273](#)
- EncoderSourceA_Line1, [273](#)
- EncoderSourceA_Line2, [273](#)
- EncoderSourceA_Off, [273](#)
- EncoderSourceAEnums, [272](#)
- EncoderSourceB_Line0, [273](#)
- EncoderSourceB_Line1, [273](#)
- EncoderSourceB_Line2, [273](#)
- EncoderSourceB_Off, [273](#)
- EncoderSourceBEnums, [273](#)
- EncoderStatus_EncoderDown, [273](#)
- EncoderStatus_EncoderIdle, [273](#)
- EncoderStatus_EncoderStatic, [273](#)
- EncoderStatus_EncoderUp, [273](#)
- EncoderStatusEnums, [273](#)
- Error, [273](#)
- EVENT_TIMEOUT_INFINITE, [338](#)
- EVENT_TIMEOUT_NONE, [338](#)
- EventNotification_Off, [275](#)
- EventNotification_On, [275](#)
- EventNotificationEnums, [274](#)
- EventSelector_Error, [275](#)
- EventSelector_ExposureEnd, [275](#)
- EventSelector_SerialPortReceive, [275](#)
- EventSelectorEnums, [275](#)
- EventType, [275](#)
- ExposureActiveMode_AllPixels, [276](#)
- ExposureActiveMode_AnyPixels, [276](#)
- ExposureActiveMode_Line1, [276](#)
- ExposureActiveModeEnums, [276](#)
- ExposureAuto_Continuous, [276](#)
- ExposureAuto_Off, [276](#)
- ExposureAuto_Once, [276](#)
- ExposureAutoEnums, [276](#)
- ExposureMode_Timed, [276](#)
- ExposureMode_TriggerWidth, [276](#)
- ExposureModeEnums, [276](#)
- ExposureTimeMode_Common, [277](#)
- ExposureTimeMode_Individual, [277](#)
- ExposureTimeModeEnums, [277](#)
- ExposureTimeSelector_Blue, [277](#)
- ExposureTimeSelector_Common, [277](#)
- ExposureTimeSelector_Cyan, [277](#)
- ExposureTimeSelector_Green, [277](#)
- ExposureTimeSelector_Infrared, [277](#)
- ExposureTimeSelector_Magenta, [277](#)
- ExposureTimeSelector_Red, [277](#)
- ExposureTimeSelector_Stage1, [277](#)
- ExposureTimeSelector_Stage2, [277](#)
- ExposureTimeSelector_Ultraviolet, [277](#)
- ExposureTimeSelector_Yellow, [277](#)
- ExposureTimeSelectorEnums, [277](#)
- FileOpenMode_Read, [279](#)
- FileOpenMode_ReadWrite, [279](#)
- FileOpenMode_Write, [279](#)
- FileOpenModeEnums, [277](#)
- FileOperationSelector_Close, [279](#)
- FileOperationSelector_Delete, [279](#)
- FileOperationSelector_Open, [279](#)
- FileOperationSelector_Read, [279](#)
- FileOperationSelector_Write, [279](#)
- FileOperationSelectorEnums, [279](#)
- FileOperationStatus_Failure, [279](#)
- FileOperationStatus_Overflow, [279](#)
- FileOperationStatus_Success, [279](#)
- FileOperationStatusEnums, [279](#)
- FileSelector_SerialPort0, [280](#)
- FileSelector_UserFile1, [280](#)
- FileSelector_UserSet0, [280](#)
- FileSelector_UserSet1, [280](#)
- FileSelector_UserSetDefault, [280](#)
- FileSelectorEnums, [279](#)
- FilterDriverStatus_Disabled, [280](#)
- FilterDriverStatus_Enabled, [280](#)
- FilterDriverStatus_NotSupported, [280](#)
- FilterDriverStatusEnum, [280](#)
- GainAuto_Continuous, [281](#)
- GainAuto_Off, [281](#)
- GainAuto_Once, [281](#)
- GainAutoBalance_Continuous, [280](#)
- GainAutoBalance_Off, [280](#)
- GainAutoBalance_Once, [280](#)
- GainAutoBalanceEnums, [280](#)
- GainAutoEnums, [280](#)
- GainSelector_All, [281](#)
- GainSelectorEnums, [281](#)
- GenICamXMLLocation_Device, [281](#)
- GenICamXMLLocation_Host, [281](#)
- GenICamXMLLocationEnum, [281](#)
- GevCCP_ControlAccess, [282](#)
- GevCCP_EnumEntry_GevCCP_ControlAccess, [282](#)
- GevCCP_EnumEntry_GevCCP_ExclusiveAccess, [282](#)
- GevCCP_EnumEntry_GevCCP_OpenAccess, [282](#)
- GevCCP_ExclusiveAccess, [282](#)
- GevCCP_OpenAccess, [282](#)
- GevCCPEnum, [281](#)
- GevCCPEnums, [282](#)
- GevCurrentPhysicalLinkConfiguration_DynamicLAG, [282](#)
- GevCurrentPhysicalLinkConfiguration_MultiLink, [282](#)
- GevCurrentPhysicalLinkConfiguration_SingleLink, [282](#)
- GevCurrentPhysicalLinkConfiguration_StaticLAG, [282](#)
- GevCurrentPhysicalLinkConfigurationEnums, [282](#)
- GevGVCPExtendedStatusCodesSelector_Version1_1, [283](#)
- GevGVCPExtendedStatusCodesSelector_Version2_0, [283](#)

- GevGVCPExtendedStatusCodesSelectorEnums, 282
- GevGVSPExtendedIDMode_Off, 283
- GevGVSPExtendedIDMode_On, 283
- GevGVSPExtendedIDModeEnums, 283
- GevIEEE1588ClockAccuracy_Unknown, 283
- GevIEEE1588ClockAccuracyEnums, 283
- GevIEEE1588Mode_Auto, 283
- GevIEEE1588Mode_SlaveOnly, 283
- GevIEEE1588ModeEnums, 283
- GevIEEE1588Status_Disabled, 284
- GevIEEE1588Status_Faulty, 284
- GevIEEE1588Status_Initializing, 284
- GevIEEE1588Status_Listening, 284
- GevIEEE1588Status_Master, 284
- GevIEEE1588Status_Passive, 284
- GevIEEE1588Status_PreMaster, 284
- GevIEEE1588Status_Slave, 284
- GevIEEE1588Status_Uncalibrated, 284
- GevIEEE1588StatusEnums, 284
- GevIPConfigurationStatus_DHCP, 284
- GevIPConfigurationStatus_ForceIP, 284
- GevIPConfigurationStatus_LLA, 284
- GevIPConfigurationStatus_None, 284
- GevIPConfigurationStatus_PersistentIP, 284
- GevIPConfigurationStatusEnums, 284
- GevPhysicalLinkConfiguration_DynamicLAG, 285
- GevPhysicalLinkConfiguration_MultiLink, 285
- GevPhysicalLinkConfiguration_SingleLink, 285
- GevPhysicalLinkConfiguration_StaticLAG, 285
- GevPhysicalLinkConfigurationEnums, 284
- GevSupportedOptionSelector_Action, 285
- GevSupportedOptionSelector_CCPApplicationSocket, 285
- GevSupportedOptionSelector_CommandsConcatenation, 285
- GevSupportedOptionSelector_DiscoveryAckDelay, 285
- GevSupportedOptionSelector_DiscoveryAckDelayWritable, 285
- GevSupportedOptionSelector_Event, 285
- GevSupportedOptionSelector_EventData, 285
- GevSupportedOptionSelector_ExtendedStatusCodes, 285
- GevSupportedOptionSelector_HeartbeatDisable, 285
- GevSupportedOptionSelector_IPConfigurationDHCP, 285
- GevSupportedOptionSelector_IPConfigurationLLA, 285
- GevSupportedOptionSelector_IPConfigurationPersistentIP, 285
- GevSupportedOptionSelector_LinkSpeed, 285
- GevSupportedOptionSelector_ManifestTable, 285
- GevSupportedOptionSelector_MessageChannelSourceSocket, 285
- GevSupportedOptionSelector_PacketResend, 285
- GevSupportedOptionSelector_PendingAck, 285
- GevSupportedOptionSelector_SerialNumber, 285
- GevSupportedOptionSelector_StreamChannelSourceSocket, 285
- GevSupportedOptionSelector_TestData, 285
- GevSupportedOptionSelector_UserDefinedName, 285
- GevSupportedOptionSelector_WriteMem, 285
- GevSupportedOptionSelectorEnums, 285
- GUIXMLLocation_Device, 287
- GUIXMLLocation_Host, 287
- GUIXMLLocationEnum, 285
- HeatmapColor, 287
- ImageComponentSelector_Color, 287
- ImageComponentSelector_Confidence, 287
- ImageComponentSelector_Disparity, 287
- ImageComponentSelector_Infrared, 287
- ImageComponentSelector_Intensity, 287
- ImageComponentSelector_Range, 287
- ImageComponentSelector_Scatter, 288
- ImageComponentSelector_Ultraviolet, 287
- ImageComponentSelectorEnums, 287
- ImageCompressionJPEGFormatOption_BaselineOptimized, 288
- ImageCompressionJPEGFormatOption_BaselineStandard, 288
- ImageCompressionJPEGFormatOption_Lossless, 288
- ImageCompressionJPEGFormatOption_Progressive, 288
- ImageCompressionJPEGFormatOptionEnums, 288
- ImageCompressionMode_Lossless, 288
- ImageCompressionMode_Off, 288
- ImageCompressionModeEnums, 288
- ImageCompressionRateOption_FixBitrate, 289
- ImageCompressionRateOption_FixQuality, 289
- ImageCompressionRateOptionEnums, 288
- ImageFileFormat, 289
- ImagePayloadType, 289
- ImageScalingAlgorithm, 290
- ImageStatus, 290
- INFERENCE_BOX_TYPE_CIRCLE, 291
- INFERENCE_BOX_TYPE_RECTANGLE, 291
- INFERENCE_BOX_TYPE_ROTATED_RECTANGLE, 291
- InferenceBoxType, 291
- InterfaceType_CameraLink, 291
- InterfaceType_CameraLinkHS, 291
- InterfaceType_CoaXPress, 291
- InterfaceType_Custom, 291
- InterfaceType_GigEVision, 291
- InterfaceType_USB3Vision, 291
- InterfaceTypeEnum, 291
- LineFormat_LVDS, 292
- LineFormat_NoConnect, 292
- LineFormat_OpenDrain, 292
- LineFormat_OptoCoupled, 292
- LineFormat_RS422, 292

- LineFormat_TriState, [292](#)
- LineFormat_TTL, [292](#)
- LineFormatEnums, [291](#)
- LineInputFilterSelector_Debounce, [292](#)
- LineInputFilterSelector_Degitch, [292](#)
- LineInputFilterSelectorEnums, [292](#)
- LineMode_Input, [292](#)
- LineMode_Output, [292](#)
- LineModeEnums, [292](#)
- LineSelector_Line0, [293](#)
- LineSelector_Line1, [293](#)
- LineSelector_Line2, [293](#)
- LineSelector_Line3, [293](#)
- LineSelectorEnums, [292](#)
- LineSource_AllPixel, [293](#)
- LineSource_AnyPixel, [293](#)
- LineSource_Counter0Active, [293](#)
- LineSource_Counter1Active, [293](#)
- LineSource_ExposureActive, [293](#)
- LineSource_FrameTriggerWait, [293](#)
- LineSource_Line0, [293](#)
- LineSource_Line1, [293](#)
- LineSource_Line2, [293](#)
- LineSource_Line3, [293](#)
- LineSource_LogicBlock0, [293](#)
- LineSource_LogicBlock1, [293](#)
- LineSource_Off, [293](#)
- LineSource_PPSSignal, [293](#)
- LineSource_SerialPort0, [293](#)
- LineSource_UserOutput0, [293](#)
- LineSource_UserOutput1, [293](#)
- LineSource_UserOutput2, [293](#)
- LineSource_UserOutput3, [293](#)
- LineSourceEnums, [293](#)
- LogicBlockLUTInputActivation_AnyEdge, [294](#)
- LogicBlockLUTInputActivation_FallingEdge, [294](#)
- LogicBlockLUTInputActivation_LevelHigh, [294](#)
- LogicBlockLUTInputActivation_LevelLow, [294](#)
- LogicBlockLUTInputActivation_RisingEdge, [294](#)
- LogicBlockLUTInputActivationEnums, [293](#)
- LogicBlockLUTInputSelector_Input0, [294](#)
- LogicBlockLUTInputSelector_Input1, [294](#)
- LogicBlockLUTInputSelector_Input2, [294](#)
- LogicBlockLUTInputSelector_Input3, [294](#)
- LogicBlockLUTInputSelectorEnums, [294](#)
- LogicBlockLUTInputSource_AcquisitionActive, [295](#)
- LogicBlockLUTInputSource_Counter0End, [295](#)
- LogicBlockLUTInputSource_Counter0Start, [295](#)
- LogicBlockLUTInputSource_Counter1End, [295](#)
- LogicBlockLUTInputSource_Counter1Start, [295](#)
- LogicBlockLUTInputSource_ExposureEnd, [295](#)
- LogicBlockLUTInputSource_ExposureStart, [295](#)
- LogicBlockLUTInputSource_FrameTriggerWait, [295](#)
- LogicBlockLUTInputSource_Line0, [294](#)
- LogicBlockLUTInputSource_Line1, [294](#)
- LogicBlockLUTInputSource_Line2, [294](#)
- LogicBlockLUTInputSource_Line3, [294](#)
- LogicBlockLUTInputSource_LogicBlock0, [295](#)
- LogicBlockLUTInputSource_LogicBlock1, [295](#)
- LogicBlockLUTInputSource_UserOutput0, [294](#)
- LogicBlockLUTInputSource_UserOutput1, [294](#)
- LogicBlockLUTInputSource_UserOutput2, [295](#)
- LogicBlockLUTInputSource_UserOutput3, [295](#)
- LogicBlockLUTInputSource_Zero, [294](#)
- LogicBlockLUTInputSourceEnums, [294](#)
- LogicBlockLUTSelector_Enable, [295](#)
- LogicBlockLUTSelector_Value, [295](#)
- LogicBlockLUTSelectorEnums, [295](#)
- LogicBlockSelector_LogicBlock0, [295](#)
- LogicBlockSelector_LogicBlock1, [295](#)
- LogicBlockSelectorEnums, [295](#)
- LUTSelector_LUT1, [296](#)
- LUTSelectorEnums, [295](#)
- NUM_ACQUISITIONMODE, [234](#)
- NUM_ACQUISITIONSTATUSSELECTOR, [234](#)
- NUM_ACTIONUNCONDITIONALMODE, [235](#)
- NUM_ADCBITDEPTH, [235](#)
- NUM_AUTOALGORITHMSELECTOR, [235](#)
- NUM_AUTOEXPOSURECONTROLPRIORITY, [236](#)
- NUM_AUTOEXPOSURELIGHTINGMODE, [236](#)
- NUM_AUTOEXPOSUREMETERINGMODE, [237](#)
- NUM_AUTOEXPOSURETARGETGREYVALUEAUTO, [237](#)
- NUM_BALANCERATIOSELECTOR, [237](#)
- NUM_BALANCEWHITEAUTO, [238](#)
- NUM_BALANCEWHITEAUTOPROFILE, [238](#)
- NUM_BINNINGHORIZONTALMODE, [238](#)
- NUM_BINNINGSELECTOR, [239](#)
- NUM_BINNINGVERTICALMODE, [239](#)
- NUM_BLACKLEVELAUTO, [240](#)
- NUM_BLACKLEVELAUTOBALANCE, [239](#)
- NUM_BLACKLEVELSELECTOR, [240](#)
- NUM_CHUNKBLACKLEVELSELECTOR, [242](#)
- NUM_CHUNKCOUNTERSELECTOR, [242](#)
- NUM_CHUNKENCODERSELECTOR, [243](#)
- NUM_CHUNKENCODERSTATUS, [243](#)
- NUM_CHUNKEXPOSURETIMESELECTOR, [243](#)
- NUM_CHUNKGAINSELECTOR, [244](#)
- NUM_CHUNKIMAGECOMPONENT, [244](#)
- NUM_CHUNKPIXELFORMAT, [245](#)
- NUM_CHUNKREGIONID, [245](#)
- NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR, [245](#)
- NUM_CHUNKSCAN3DCOORDINATESELECTOR, [246](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEM, [246](#)
- NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE, [246](#)
- NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR, [247](#)
- NUM_CHUNKSCAN3DDISTANCEUNIT, [247](#)
- NUM_CHUNKSCAN3DOUTPUTMODE, [248](#)
- NUM_CHUNKSELECTOR, [249](#)

- NUM_CHUNKSOURCEID, [249](#)
- NUM_CHUNKTIMERSELECTOR, [249](#)
- NUM_CHUNKTRANSFERSTREAMID, [250](#)
- NUM_CLCONFIGURATION, [250](#)
- NUM_CLTIMESLOTSCOUNT, [250](#)
- NUM_COLORTRANSFORMATIONSELECTOR, [251](#)
- NUM_COLORTRANSFORMATIONVALUESELECTOR, [252](#)
- NUM_COMPRESSIONSATURATIONPRIORITY, [252](#)
- NUM_COUNTEREVENTACTIVATION, [253](#)
- NUM_COUNTEREVENTSOURCE, [253](#)
- NUM_COUNTERRESETACTIVATION, [254](#)
- NUM_COUNTERRESETSOURCE, [254](#)
- NUM_COUNTERSELECTOR, [255](#)
- NUM_COUNTERSTATUS, [255](#)
- NUM_COUNTERTRIGGERACTIVATION, [255](#)
- NUM_COUNTERTRIGGERSOURCE, [256](#)
- NUM_CXPCONNECTIONTESTMODE, [256](#)
- NUM_CXPLINKCONFIGURATION, [257](#)
- NUM_CXPLINKCONFIGURATIONPREFERRED, [258](#)
- NUM_CXPLINKCONFIGURATIONSTATUS, [259](#)
- NUM_CXPPOCXPSTATUS, [259](#)
- NUM_DECIMATIONHORIZONTALMODE, [260](#)
- NUM_DECIMATIONSELECTOR, [260](#)
- NUM_DECIMATIONVERTICALMODE, [260](#)
- NUM_DEFECTCORRECTIONMODE, [261](#)
- NUM_DEINTERLACING, [261](#)
- NUM_DEVICECHARACTERSET, [262](#)
- NUM_DEVICECLOCKSELECTOR, [262](#)
- NUM_DEVICECONNECTIONSTATUS, [262](#)
- NUM_DEVICEINDICATORMODE, [263](#)
- NUM_DEVICELINKHEARTBEATMODE, [264](#)
- NUM_DEVICELINKTHROUGHPUTLIMITMODE, [264](#)
- NUM_DEVICEPOWERSUPPLYSELECTOR, [264](#)
- NUM_DEVICEREGISTERSENDIANNES, [264](#)
- NUM_DEVICESCANTYPE, [265](#)
- NUM_DEVICESERIALPORTBAUDRATE, [265](#)
- NUM_DEVICESERIALPORTSELECTOR, [265](#)
- NUM_DEVICESTREAMCHANNELENDIANNES, [266](#)
- NUM_DEVICESTREAMCHANNELTYPE, [266](#)
- NUM_DEVICETAPGEOMETRY, [267](#)
- NUM_DEVICETEMPERATURESELECTOR, [268](#)
- NUM_DEVICETLTYPE, [268](#)
- NUM_DEVICETYPE, [270](#)
- NUM_ENCODERMODE, [270](#)
- NUM_ENCODEROUTPUTMODE, [270](#)
- NUM_ENCODERRESETACTIVATION, [271](#)
- NUM_ENCODERRESETSOURCE, [272](#)
- NUM_ENCODERSELECTOR, [272](#)
- NUM_ENCODERSOURCEA, [273](#)
- NUM_ENCODERSOURCEB, [273](#)
- NUM_ENCODERSTATUS, [273](#)
- NUM_EVENTNOTIFICATION, [275](#)
- NUM_EVENTSELECTOR, [275](#)
- NUM_EXPOSUREACTIVEMODE, [276](#)
- NUM_EXPOSUREAUTO, [276](#)
- NUM_EXPOSUREMODE, [276](#)
- NUM_EXPOSURETIMEMODE, [277](#)
- NUM_EXPOSURETIMESELECTOR, [277](#)
- NUM_FILEOPENMODE, [279](#)
- NUM_FILEOPERATIONSELECTOR, [279](#)
- NUM_FILEOPERATIONSTATUS, [279](#)
- NUM_FILESELECTOR, [280](#)
- NUM_GAINAUTO, [281](#)
- NUM_GAINAUTOBALANCE, [280](#)
- NUM_GAINSELECTOR, [281](#)
- NUM_GEVCAP, [282](#)
- NUM_GEVCURRENTPHYSICALLINKCONFIGURATION, [282](#)
- NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR, [283](#)
- NUM_GEVGVSPEXTENDEDIDMODE, [283](#)
- NUM_GEVEEEE1588CLOCKACCURACY, [283](#)
- NUM_GEVEEEE1588MODE, [283](#)
- NUM_GEVEEEE1588STATUS, [284](#)
- NUM_GEVIPCONFIGURATIONSTATUS, [284](#)
- NUM_GEVPHYSCALLINKCONFIGURATION, [285](#)
- NUM_GEVSUPPORTEDOPTIONSELECTOR, [285](#)
- NUM_IMAGECOMPONENTSELECTOR, [288](#)
- NUM_IMAGECOMPRESSIONJPEGFORMATOPTION, [288](#)
- NUM_IMAGECOMPRESSIONMODE, [288](#)
- NUM_IMAGECOMPRESSIONRATEOPTION, [289](#)
- NUM_LINEFORMAT, [292](#)
- NUM_LINEINPUTFILTERSELECTOR, [292](#)
- NUM_LINEMODE, [292](#)
- NUM_LINESELECTOR, [293](#)
- NUM_LINESOURCE, [293](#)
- NUM_LOGICBLOCKLUTINPUTACTIVATION, [294](#)
- NUM_LOGICBLOCKLUTINPUTSELECTOR, [294](#)
- NUM_LOGICBLOCKLUTINPUTSOURCE, [295](#)
- NUM_LOGICBLOCKLUTSELECTOR, [295](#)
- NUM_LOGICBLOCKSELECTOR, [295](#)
- NUM_LUTSELECTOR, [296](#)
- NUM_PIXELCOLORFILTER, [296](#)
- NUM_PIXELFORMAT, [302](#)
- NUM_PIXELFORMATINFOSELECTOR, [307](#)
- NUM_PIXELSIZE, [308](#)
- NUM_REGIONDESTINATION, [309](#)
- NUM_REGIONMODE, [310](#)
- NUM_REGIONSELECTOR, [310](#)
- NUM_RGBTRANSFORMLIGHTSOURCE, [311](#)
- NUM_SCAN3DCOORDINATEREFERENCESELECTOR, [311](#)
- NUM_SCAN3DCOORDINATESELECTOR, [311](#)
- NUM_SCAN3DCOORDINATESYSTEM, [312](#)
- NUM_SCAN3DCOORDINATESYSTEMREFERENCE, [312](#)
- NUM_SCAN3DCOORDINATETTRANSFORMSELECTOR, [312](#)

- NUM_SCAN3DDISTANCEUNIT, [313](#)
- NUM_SCAN3DOUTPUTMODE, [314](#)
- NUM_SENSORDIGITIZATIONTAPS, [314](#)
- NUM_SENSORSHUTTERMODE, [314](#)
- NUM_SENSORTAPS, [315](#)
- NUM_SEQUENCERCONFIGURATIONMODE, [315](#)
- NUM_SEQUENCERCONFIGURATIONVALID, [315](#)
- NUM_SEQUENCERMODE, [316](#)
- NUM_SEQUENCERSETVALID, [316](#)
- NUM_SEQUENCERTRIGGERACTIVATION, [316](#)
- NUM_SEQUENCERTRIGGERSOURCE, [317](#)
- NUM_SERIALPORTBAUDRATE, [317](#)
- NUM_SERIALPORTPARITY, [317](#)
- NUM_SERIALPORTSELECTOR, [318](#)
- NUM_SERIALPORTSOURCE, [318](#)
- NUM_SERIALPORTSTOPBITS, [318](#)
- NUM_SOFTWARESIGNALSELECTOR, [319](#)
- NUM_SOURCESELECTOR, [320](#)
- NUM_TESTPATTERN, [322](#)
- NUM_TESTPATTERNGENERATORSELECTOR, [323](#)
- NUM_TIMERSELECTOR, [325](#)
- NUM_TIMERSTATUS, [325](#)
- NUM_TIMERTRIGGERACTIVATION, [325](#)
- NUM_TIMERTRIGGERSOURCE, [327](#)
- NUM_TRANSFERCOMPONENTSELECTOR, [329](#)
- NUM_TRANSFERCONTROLMODE, [329](#)
- NUM_TRANSFEROPERATIONMODE, [329](#)
- NUM_TRANSFERQUEUEMODE, [330](#)
- NUM_TRANSFERSELECTOR, [330](#)
- NUM_TRANSFERSTATUSSELECTOR, [330](#)
- NUM_TRANSFERTRIGGERACTIVATION, [331](#)
- NUM_TRANSFERTRIGGERMODE, [331](#)
- NUM_TRANSFERTRIGGERSELECTOR, [331](#)
- NUM_TRANSFERTRIGGERSOURCE, [332](#)
- NUM_TRIGGERACTIVATION, [333](#)
- NUM_TRIGGERMODE, [333](#)
- NUM_TRIGGEROVERLAP, [333](#)
- NUM_TRIGGERSELECTOR, [334](#)
- NUM_TRIGGERSOURCE, [334](#)
- NUM_USEROUTPUTSELECTOR, [335](#)
- NUM_USERSETDEFAULT, [335](#)
- NUM_USERSETSELECTOR, [335](#)
- NUM_WHITECLIPSELECTOR, [336](#)
- NUMDEVICEACCESSSTATUS, [261](#)
- NUMDEVICECURRENTSPEED, [263](#)
- NUMDEVICEENDIANESSMECHANISM, [263](#)
- NUMDEVICETYPE, [268](#)
- NUMFILTERDRIVERSTATUS, [280](#)
- NUMGENICAMXMLLOCATION, [281](#)
- NUMGEVCCP, [282](#)
- NUMGUIXMLLOCATION, [287](#)
- NUMINTERFACETYPE, [291](#)
- NUMPOESTATUS, [309](#)
- NUMSTREAMBUFFERCOUNTMODE, [321](#)
- NUMSTREAMBUFFERHANDLINGMODE, [321](#)
- NUMSTREAMMODE, [322](#)
- NUMSTREAMTYPE, [322](#)
- NUMTLTYPE, [328](#)
- operator==, [337](#)
- PixelFormat_BayerBG, [296](#)
- PixelFormat_BayerGB, [296](#)
- PixelFormat_BayerGR, [296](#)
- PixelFormat_BayerRG, [296](#)
- PixelFormat_None, [296](#)
- PixelFormatEnums, [296](#)
- PixelFormat_B10, [298](#)
- PixelFormat_B12, [299](#)
- PixelFormat_B12_Jpeg, [302](#)
- PixelFormat_B16, [299](#)
- PixelFormat_B8, [298](#)
- PixelFormat_BayerBG10, [297](#)
- PixelFormat_BayerBG10p, [297](#)
- PixelFormat_BayerBG10Packed, [297](#)
- PixelFormat_BayerBG12, [297](#)
- PixelFormat_BayerBG12p, [297](#)
- PixelFormat_BayerBG12Packed, [297](#)
- PixelFormat_BayerBG16, [297](#)
- PixelFormat_BayerBG8, [296](#)
- PixelFormat_BayerGB10, [297](#)
- PixelFormat_BayerGB10p, [297](#)
- PixelFormat_BayerGB10Packed, [297](#)
- PixelFormat_BayerGB12, [297](#)
- PixelFormat_BayerGB12p, [297](#)
- PixelFormat_BayerGB12Packed, [297](#)
- PixelFormat_BayerGB16, [297](#)
- PixelFormat_BayerGB8, [296](#)
- PixelFormat_BayerGR10, [297](#)
- PixelFormat_BayerGR10p, [297](#)
- PixelFormat_BayerGR10Packed, [297](#)
- PixelFormat_BayerGR12, [297](#)
- PixelFormat_BayerGR12p, [297](#)
- PixelFormat_BayerGR12Packed, [297](#)
- PixelFormat_BayerGR16, [296](#)
- PixelFormat_BayerGR8, [296](#)
- PixelFormat_BayerRG10, [297](#)
- PixelFormat_BayerRG10p, [297](#)
- PixelFormat_BayerRG10Packed, [297](#)
- PixelFormat_BayerRG12, [297](#)
- PixelFormat_BayerRG12p, [297](#)
- PixelFormat_BayerRG12Packed, [297](#)
- PixelFormat_BayerRG16, [296](#)
- PixelFormat_BayerRG8, [296](#)
- PixelFormat_BayerRGPolarized10p, [301](#)
- PixelFormat_BayerRGPolarized12p, [301](#)
- PixelFormat_BayerRGPolarized16, [301](#)
- PixelFormat_BayerRGPolarized8, [301](#)
- PixelFormat_BGR10, [298](#)
- PixelFormat_BGR10p, [298](#)
- PixelFormat_BGR12, [298](#)
- PixelFormat_BGR12p, [298](#)
- PixelFormat_BGR14, [298](#)
- PixelFormat_BGR16, [298](#)
- PixelFormat_BGR565p, [298](#)
- PixelFormat_BGR8, [297](#)

PixelFormat_BGRa10, [298](#)
PixelFormat_BGRa10p, [298](#)
PixelFormat_BGRa12, [298](#)
PixelFormat_BGRa12p, [298](#)
PixelFormat_BGRa14, [298](#)
PixelFormat_BGRa16, [298](#)
PixelFormat_BGRa8, [297](#)
PixelFormat_BiColorBGRG10, [299](#)
PixelFormat_BiColorBGRG10p, [299](#)
PixelFormat_BiColorBGRG12, [299](#)
PixelFormat_BiColorBGRG12p, [299](#)
PixelFormat_BiColorBGRG8, [299](#)
PixelFormat_BiColorRGBG10, [300](#)
PixelFormat_BiColorRGBG10p, [300](#)
PixelFormat_BiColorRGBG12, [300](#)
PixelFormat_BiColorRGBG12p, [300](#)
PixelFormat_BiColorRGBG8, [300](#)
PixelFormat_Confidence1, [299](#)
PixelFormat_Confidence16, [299](#)
PixelFormat_Confidence1p, [299](#)
PixelFormat_Confidence32f, [299](#)
PixelFormat_Confidence8, [299](#)
PixelFormat_Coord3D_A10p, [299](#)
PixelFormat_Coord3D_A12p, [299](#)
PixelFormat_Coord3D_A16, [299](#)
PixelFormat_Coord3D_A32f, [299](#)
PixelFormat_Coord3D_A8, [299](#)
PixelFormat_Coord3D_ABC10p, [299](#)
PixelFormat_Coord3D_ABC10p_Planar, [299](#)
PixelFormat_Coord3D_ABC12p, [299](#)
PixelFormat_Coord3D_ABC12p_Planar, [299](#)
PixelFormat_Coord3D_ABC16, [299](#)
PixelFormat_Coord3D_ABC16_Planar, [299](#)
PixelFormat_Coord3D_ABC32f, [299](#)
PixelFormat_Coord3D_ABC32f_Planar, [299](#)
PixelFormat_Coord3D_ABC8, [299](#)
PixelFormat_Coord3D_ABC8_Planar, [299](#)
PixelFormat_Coord3D_AC10p, [299](#)
PixelFormat_Coord3D_AC10p_Planar, [299](#)
PixelFormat_Coord3D_AC12p, [299](#)
PixelFormat_Coord3D_AC12p_Planar, [299](#)
PixelFormat_Coord3D_AC16, [299](#)
PixelFormat_Coord3D_AC16_Planar, [299](#)
PixelFormat_Coord3D_AC32f, [299](#)
PixelFormat_Coord3D_AC32f_Planar, [299](#)
PixelFormat_Coord3D_AC8, [299](#)
PixelFormat_Coord3D_AC8_Planar, [299](#)
PixelFormat_Coord3D_B10p, [299](#)
PixelFormat_Coord3D_B12p, [299](#)
PixelFormat_Coord3D_B16, [299](#)
PixelFormat_Coord3D_B32f, [299](#)
PixelFormat_Coord3D_B8, [299](#)
PixelFormat_Coord3D_C10p, [299](#)
PixelFormat_Coord3D_C12p, [299](#)
PixelFormat_Coord3D_C16, [299](#)
PixelFormat_Coord3D_C32f, [299](#)
PixelFormat_Coord3D_C8, [299](#)
PixelFormat_G10, [298](#)
PixelFormat_G12, [298](#)
PixelFormat_G16, [298](#)
PixelFormat_G8, [298](#)
PixelFormat_GB12, [302](#)
PixelFormat_GB12_Jpeg, [302](#)
PixelFormat_GR12, [302](#)
PixelFormat_GR12_Jpeg, [302](#)
PixelFormat_JPEGColor8, [302](#)
PixelFormat_JPEGMono8, [302](#)
PixelFormat_LLCBayerRG8, [301](#)
PixelFormat_LLCMono8, [301](#)
PixelFormat_Mono10, [297](#)
PixelFormat_Mono10p, [297](#)
PixelFormat_Mono10Packed, [297](#)
PixelFormat_Mono12, [297](#)
PixelFormat_Mono12p, [297](#)
PixelFormat_Mono12Packed, [297](#)
PixelFormat_Mono14, [297](#)
PixelFormat_Mono16, [296](#)
PixelFormat_Mono16s, [297](#)
PixelFormat_Mono1p, [297](#)
PixelFormat_Mono2p, [297](#)
PixelFormat_Mono32f, [297](#)
PixelFormat_Mono4p, [297](#)
PixelFormat_Mono8, [296](#)
PixelFormat_Mono8s, [297](#)
PixelFormat_Polarized10p, [301](#)
PixelFormat_Polarized12p, [301](#)
PixelFormat_Polarized16, [301](#)
PixelFormat_Polarized8, [301](#)
PixelFormat_R10, [298](#)
PixelFormat_R12, [298](#)
PixelFormat_R12_Jpeg, [302](#)
PixelFormat_R16, [298](#)
PixelFormat_R8, [298](#)
PixelFormat_Raw16, [302](#)
PixelFormat_Raw8, [302](#)
PixelFormat_RGB10, [298](#)
PixelFormat_RGB10_Planar, [298](#)
PixelFormat_RGB10p, [298](#)
PixelFormat_RGB10p32, [298](#)
PixelFormat_RGB12, [298](#)
PixelFormat_RGB12_Planar, [298](#)
PixelFormat_RGB12p, [298](#)
PixelFormat_RGB14, [298](#)
PixelFormat_RGB16, [298](#)
PixelFormat_RGB16_Planar, [298](#)
PixelFormat_RGB16s, [298](#)
PixelFormat_RGB32f, [298](#)
PixelFormat_RGB565p, [298](#)
PixelFormat_RGB8, [298](#)
PixelFormat_RGB8_Planar, [298](#)
PixelFormat_RGB8Packed, [296](#)
PixelFormat_RGBa10, [298](#)
PixelFormat_RGBa10p, [298](#)
PixelFormat_RGBa12, [298](#)
PixelFormat_RGBa12p, [298](#)
PixelFormat_RGBa14, [298](#)

PixelFormat_RGBA16, [298](#)
PixelFormat_RGBA32f, [298](#)
PixelFormat_RGBA8, [298](#)
PixelFormat_SCF1WBWG10, [300](#)
PixelFormat_SCF1WBWG10p, [300](#)
PixelFormat_SCF1WBWG12, [300](#)
PixelFormat_SCF1WBWG12p, [300](#)
PixelFormat_SCF1WBWG14, [300](#)
PixelFormat_SCF1WBWG16, [300](#)
PixelFormat_SCF1WBWG8, [300](#)
PixelFormat_SCF1WGWB10, [300](#)
PixelFormat_SCF1WGWB10p, [300](#)
PixelFormat_SCF1WGWB12, [300](#)
PixelFormat_SCF1WGWB12p, [300](#)
PixelFormat_SCF1WGWB14, [300](#)
PixelFormat_SCF1WGWB16, [300](#)
PixelFormat_SCF1WGWB8, [300](#)
PixelFormat_SCF1WGWR10, [300](#)
PixelFormat_SCF1WGWR10p, [300](#)
PixelFormat_SCF1WGWR12, [300](#)
PixelFormat_SCF1WGWR12p, [300](#)
PixelFormat_SCF1WGWR14, [300](#)
PixelFormat_SCF1WGWR16, [300](#)
PixelFormat_SCF1WGWR8, [300](#)
PixelFormat_SCF1WRWG10, [300](#)
PixelFormat_SCF1WRWG10p, [300](#)
PixelFormat_SCF1WRWG12, [300](#)
PixelFormat_SCF1WRWG12p, [300](#)
PixelFormat_SCF1WRWG14, [300](#)
PixelFormat_SCF1WRWG16, [300](#)
PixelFormat_SCF1WRWG8, [300](#)
PixelFormat_YCbCr10_CbYCr, [300](#)
PixelFormat_YCbCr10p_CbYCr, [300](#)
PixelFormat_YCbCr12_CbYCr, [300](#)
PixelFormat_YCbCr12p_CbYCr, [300](#)
PixelFormat_YCbCr411_8, [297](#)
PixelFormat_YCbCr411_8_CbYYCrYY, [300](#)
PixelFormat_YCbCr422_10, [300](#)
PixelFormat_YCbCr422_10_CbYCrY, [300](#)
PixelFormat_YCbCr422_10p, [300](#)
PixelFormat_YCbCr422_10p_CbYCrY, [300](#)
PixelFormat_YCbCr422_12, [300](#)
PixelFormat_YCbCr422_12_CbYCrY, [300](#)
PixelFormat_YCbCr422_12p, [300](#)
PixelFormat_YCbCr422_12p_CbYCrY, [301](#)
PixelFormat_YCbCr422_8, [297](#)
PixelFormat_YCbCr422_8_CbYCrY, [300](#)
PixelFormat_YCbCr601_10_CbYCr, [301](#)
PixelFormat_YCbCr601_10p_CbYCr, [301](#)
PixelFormat_YCbCr601_12_CbYCr, [301](#)
PixelFormat_YCbCr601_12p_CbYCr, [301](#)
PixelFormat_YCbCr601_411_8_CbYYCrYY, [301](#)
PixelFormat_YCbCr601_422_10, [301](#)
PixelFormat_YCbCr601_422_10_CbYCrY, [301](#)
PixelFormat_YCbCr601_422_10p, [301](#)
PixelFormat_YCbCr601_422_10p_CbYCrY, [301](#)
PixelFormat_YCbCr601_422_12, [301](#)
PixelFormat_YCbCr601_422_12_CbYCrY, [301](#)
PixelFormat_YCbCr601_422_12p, [301](#)
PixelFormat_YCbCr601_422_12p_CbYCrY, [301](#)
PixelFormat_YCbCr601_422_8_CbYYCrYY, [301](#)
PixelFormat_YCbCr709_10_CbYCr, [301](#)
PixelFormat_YCbCr709_10p_CbYCr, [301](#)
PixelFormat_YCbCr709_12_CbYCr, [301](#)
PixelFormat_YCbCr709_12p_CbYCr, [301](#)
PixelFormat_YCbCr709_411_8_CbYYCrYY, [301](#)
PixelFormat_YCbCr709_422_10, [301](#)
PixelFormat_YCbCr709_422_10_CbYCrY, [301](#)
PixelFormat_YCbCr709_422_10p, [301](#)
PixelFormat_YCbCr709_422_10p_CbYCrY, [301](#)
PixelFormat_YCbCr709_422_12, [301](#)
PixelFormat_YCbCr709_422_12_CbYCrY, [301](#)
PixelFormat_YCbCr709_422_12p, [301](#)
PixelFormat_YCbCr709_422_12p_CbYCrY, [301](#)
PixelFormat_YCbCr709_422_8, [301](#)
PixelFormat_YCbCr709_422_8_CbYCrY, [301](#)
PixelFormat_YCbCr709_8_CbYCr, [301](#)
PixelFormat_YCbCr8, [297](#)
PixelFormat_YCbCr8_CbYCr, [300](#)
PixelFormat_YUV411_8_UYYVYY, [301](#)
PixelFormat_YUV411Packed, [297](#)
PixelFormat_YUV422_8, [301](#)
PixelFormat_YUV422_8_UYVY, [301](#)
PixelFormat_YUV422Packed, [297](#)
PixelFormat_YUV444Packed, [297](#)
PixelFormat_YUV8_UYV, [301](#)
PixelFormatEnums, [296](#)
PixelFormatInfoSelector_B10, [304](#)
PixelFormatInfoSelector_B12, [304](#)
PixelFormatInfoSelector_B16, [304](#)
PixelFormatInfoSelector_B8, [304](#)
PixelFormatInfoSelector_BayerBG10, [302](#)
PixelFormatInfoSelector_BayerBG10p, [302](#)
PixelFormatInfoSelector_BayerBG12, [302](#)
PixelFormatInfoSelector_BayerBG12p, [302](#)
PixelFormatInfoSelector_BayerBG16, [302](#)
PixelFormatInfoSelector_BayerBG8, [302](#)
PixelFormatInfoSelector_BayerGB10, [302](#)
PixelFormatInfoSelector_BayerGB10p, [302](#)
PixelFormatInfoSelector_BayerGB12, [302](#)
PixelFormatInfoSelector_BayerGB12p, [302](#)
PixelFormatInfoSelector_BayerGB16, [303](#)
PixelFormatInfoSelector_BayerGB8, [302](#)
PixelFormatInfoSelector_BayerGR10, [303](#)
PixelFormatInfoSelector_BayerGR10p, [303](#)
PixelFormatInfoSelector_BayerGR12, [303](#)
PixelFormatInfoSelector_BayerGR12p, [303](#)
PixelFormatInfoSelector_BayerGR16, [303](#)
PixelFormatInfoSelector_BayerGR8, [303](#)
PixelFormatInfoSelector_BayerRG10, [303](#)
PixelFormatInfoSelector_BayerRG10p, [303](#)
PixelFormatInfoSelector_BayerRG12, [303](#)
PixelFormatInfoSelector_BayerRG12p, [303](#)
PixelFormatInfoSelector_BayerRG16, [303](#)

- PixelFormatInfoSelector_BayerRG8, [303](#)
- PixelFormatInfoSelector_BayerRGPolarized10p, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized12p, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized16, [307](#)
- PixelFormatInfoSelector_BayerRGPolarized8, [307](#)
- PixelFormatInfoSelector_BGR10, [303](#)
- PixelFormatInfoSelector_BGR10p, [304](#)
- PixelFormatInfoSelector_BGR12, [304](#)
- PixelFormatInfoSelector_BGR12p, [304](#)
- PixelFormatInfoSelector_BGR14, [304](#)
- PixelFormatInfoSelector_BGR16, [304](#)
- PixelFormatInfoSelector_BGR565p, [304](#)
- PixelFormatInfoSelector_BGR8, [303](#)
- PixelFormatInfoSelector_BGRa10, [303](#)
- PixelFormatInfoSelector_BGRa10p, [303](#)
- PixelFormatInfoSelector_BGRa12, [303](#)
- PixelFormatInfoSelector_BGRa12p, [303](#)
- PixelFormatInfoSelector_BGRa14, [303](#)
- PixelFormatInfoSelector_BGRa16, [303](#)
- PixelFormatInfoSelector_BGRa8, [303](#)
- PixelFormatInfoSelector_BiColorBGRG10, [305](#)
- PixelFormatInfoSelector_BiColorBGRG10p, [305](#)
- PixelFormatInfoSelector_BiColorBGRG12, [305](#)
- PixelFormatInfoSelector_BiColorBGRG12p, [305](#)
- PixelFormatInfoSelector_BiColorBGRG8, [305](#)
- PixelFormatInfoSelector_BiColorRGBG10, [305](#)
- PixelFormatInfoSelector_BiColorRGBG10p, [305](#)
- PixelFormatInfoSelector_BiColorRGBG12, [305](#)
- PixelFormatInfoSelector_BiColorRGBG12p, [305](#)
- PixelFormatInfoSelector_BiColorRGBG8, [305](#)
- PixelFormatInfoSelector_Confidence1, [305](#)
- PixelFormatInfoSelector_Confidence16, [305](#)
- PixelFormatInfoSelector_Confidence1p, [305](#)
- PixelFormatInfoSelector_Confidence32f, [305](#)
- PixelFormatInfoSelector_Confidence8, [305](#)
- PixelFormatInfoSelector_Coord3D_A10p, [304](#)
- PixelFormatInfoSelector_Coord3D_A12p, [304](#)
- PixelFormatInfoSelector_Coord3D_A16, [304](#)
- PixelFormatInfoSelector_Coord3D_A32f, [304](#)
- PixelFormatInfoSelector_Coord3D_A8, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC10p, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC10p_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC12p, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC12p_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC16, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC16_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC32f, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC32f_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC8, [304](#)
- PixelFormatInfoSelector_Coord3D_ABC8_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_AC10p, [304](#)
- PixelFormatInfoSelector_Coord3D_AC10p_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_AC12p, [304](#)
- PixelFormatInfoSelector_Coord3D_AC12p_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_AC16, [304](#)
- PixelFormatInfoSelector_Coord3D_AC16_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_AC32f, [304](#)
- PixelFormatInfoSelector_Coord3D_AC32f_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_AC8, [304](#)
- PixelFormatInfoSelector_Coord3D_AC8_Planar, [304](#)
- PixelFormatInfoSelector_Coord3D_B10p, [304](#)
- PixelFormatInfoSelector_Coord3D_B12p, [304](#)
- PixelFormatInfoSelector_Coord3D_B16, [304](#)
- PixelFormatInfoSelector_Coord3D_B32f, [304](#)
- PixelFormatInfoSelector_Coord3D_B8, [304](#)
- PixelFormatInfoSelector_Coord3D_C10p, [305](#)
- PixelFormatInfoSelector_Coord3D_C12p, [305](#)
- PixelFormatInfoSelector_Coord3D_C16, [305](#)
- PixelFormatInfoSelector_Coord3D_C32f, [305](#)
- PixelFormatInfoSelector_Coord3D_C8, [305](#)
- PixelFormatInfoSelector_G10, [304](#)
- PixelFormatInfoSelector_G12, [304](#)
- PixelFormatInfoSelector_G16, [304](#)
- PixelFormatInfoSelector_G8, [304](#)
- PixelFormatInfoSelector_JPEGColor8, [307](#)
- PixelFormatInfoSelector_JPEGMono8, [307](#)
- PixelFormatInfoSelector_LLCBayerRG8, [307](#)
- PixelFormatInfoSelector_LLCMono8, [307](#)
- PixelFormatInfoSelector_Mono10, [302](#)
- PixelFormatInfoSelector_Mono10p, [302](#)
- PixelFormatInfoSelector_Mono12, [302](#)
- PixelFormatInfoSelector_Mono12p, [302](#)
- PixelFormatInfoSelector_Mono14, [302](#)
- PixelFormatInfoSelector_Mono16, [302](#)
- PixelFormatInfoSelector_Mono16s, [302](#)
- PixelFormatInfoSelector_Mono1p, [302](#)
- PixelFormatInfoSelector_Mono2p, [302](#)
- PixelFormatInfoSelector_Mono32f, [302](#)
- PixelFormatInfoSelector_Mono4p, [302](#)
- PixelFormatInfoSelector_Mono8, [302](#)
- PixelFormatInfoSelector_Mono8s, [302](#)
- PixelFormatInfoSelector_Polarized10p, [307](#)
- PixelFormatInfoSelector_Polarized12p, [307](#)
- PixelFormatInfoSelector_Polarized16, [307](#)
- PixelFormatInfoSelector_Polarized8, [307](#)
- PixelFormatInfoSelector_R10, [304](#)
- PixelFormatInfoSelector_R12, [304](#)
- PixelFormatInfoSelector_R16, [304](#)
- PixelFormatInfoSelector_R8, [304](#)
- PixelFormatInfoSelector_RGB10, [303](#)
- PixelFormatInfoSelector_RGB10_Planar, [303](#)
- PixelFormatInfoSelector_RGB10p, [303](#)
- PixelFormatInfoSelector_RGB10p32, [303](#)

- PixelFormatInfoSelector_RGB12, [303](#)
- PixelFormatInfoSelector_RGB12_Planar, [303](#)
- PixelFormatInfoSelector_RGB12p, [303](#)
- PixelFormatInfoSelector_RGB14, [303](#)
- PixelFormatInfoSelector_RGB16, [303](#)
- PixelFormatInfoSelector_RGB16_Planar, [303](#)
- PixelFormatInfoSelector_RGB16s, [303](#)
- PixelFormatInfoSelector_RGB32f, [303](#)
- PixelFormatInfoSelector_RGB565p, [303](#)
- PixelFormatInfoSelector_RGB8, [303](#)
- PixelFormatInfoSelector_RGB8_Planar, [303](#)
- PixelFormatInfoSelector_RGBa10, [303](#)
- PixelFormatInfoSelector_RGBa10p, [303](#)
- PixelFormatInfoSelector_RGBa12, [303](#)
- PixelFormatInfoSelector_RGBa12p, [303](#)
- PixelFormatInfoSelector_RGBa14, [303](#)
- PixelFormatInfoSelector_RGBa16, [303](#)
- PixelFormatInfoSelector_RGBa32f, [303](#)
- PixelFormatInfoSelector_RGBa8, [303](#)
- PixelFormatInfoSelector_SCF1WBWG10, [305](#)
- PixelFormatInfoSelector_SCF1WBWG10p, [305](#)
- PixelFormatInfoSelector_SCF1WBWG12, [305](#)
- PixelFormatInfoSelector_SCF1WBWG12p, [305](#)
- PixelFormatInfoSelector_SCF1WBWG14, [305](#)
- PixelFormatInfoSelector_SCF1WBWG16, [305](#)
- PixelFormatInfoSelector_SCF1WBWG8, [305](#)
- PixelFormatInfoSelector_SCF1WGWB10, [305](#)
- PixelFormatInfoSelector_SCF1WGWB10p, [305](#)
- PixelFormatInfoSelector_SCF1WGWB12, [305](#)
- PixelFormatInfoSelector_SCF1WGWB12p, [305](#)
- PixelFormatInfoSelector_SCF1WGWB14, [305](#)
- PixelFormatInfoSelector_SCF1WGWB16, [305](#)
- PixelFormatInfoSelector_SCF1WGWB8, [305](#)
- PixelFormatInfoSelector_SCF1WGWR10, [305](#)
- PixelFormatInfoSelector_SCF1WGWR10p, [306](#)
- PixelFormatInfoSelector_SCF1WGWR12, [306](#)
- PixelFormatInfoSelector_SCF1WGWR12p, [306](#)
- PixelFormatInfoSelector_SCF1WGWR14, [306](#)
- PixelFormatInfoSelector_SCF1WGWR16, [306](#)
- PixelFormatInfoSelector_SCF1WGWR8, [305](#)
- PixelFormatInfoSelector_SCF1WRWG10, [306](#)
- PixelFormatInfoSelector_SCF1WRWG10p, [306](#)
- PixelFormatInfoSelector_SCF1WRWG12, [306](#)
- PixelFormatInfoSelector_SCF1WRWG12p, [306](#)
- PixelFormatInfoSelector_SCF1WRWG14, [306](#)
- PixelFormatInfoSelector_SCF1WRWG16, [306](#)
- PixelFormatInfoSelector_SCF1WRWG8, [306](#)
- PixelFormatInfoSelector_YCbCr10_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr10p_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr12_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr12p_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr411_8, [306](#)
- PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY, [306](#)
- PixelFormatInfoSelector_YCbCr422_10, [306](#)
- PixelFormatInfoSelector_YCbCr422_10_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr422_10p, [306](#)
- PixelFormatInfoSelector_YCbCr422_10p_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr422_12, [306](#)
- PixelFormatInfoSelector_YCbCr422_12_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr422_12p, [306](#)
- PixelFormatInfoSelector_YCbCr422_12p_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr422_8, [306](#)
- PixelFormatInfoSelector_YCbCr422_8_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr601_10_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr601_10p_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr601_12_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr601_12p_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_10, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_10p, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12p, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr601_422_8, [306](#)
- PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY, [306](#)
- PixelFormatInfoSelector_YCbCr601_8_CbYCr, [306](#)
- PixelFormatInfoSelector_YCbCr709_10_CbYCr, [307](#)
- PixelFormatInfoSelector_YCbCr709_10p_CbYCr, [307](#)
- PixelFormatInfoSelector_YCbCr709_12_CbYCr, [307](#)
- PixelFormatInfoSelector_YCbCr709_12p_CbYCr, [307](#)
- PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10p, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_12p, [307](#)

- PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_8, [307](#)
- PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY, [307](#)
- PixelFormatInfoSelector_YCbCr709_8_CbYCr, [307](#)
- PixelFormatInfoSelector_YCbCr8, [306](#)
- PixelFormatInfoSelector_YCbCr8_CbYCr, [306](#)
- PixelFormatInfoSelector_YUV411_8_UYVYY, [307](#)
- PixelFormatInfoSelector_YUV422_8, [307](#)
- PixelFormatInfoSelector_YUV422_8_UYVY, [307](#)
- PixelFormatInfoSelector_YUV8_UYV, [307](#)
- PixelFormatInfoSelectorEnums, [302](#)
- PixelFormatIntType, [308](#)
- PixelSize_Bpp1, [308](#)
- PixelSize_Bpp10, [308](#)
- PixelSize_Bpp12, [308](#)
- PixelSize_Bpp14, [308](#)
- PixelSize_Bpp16, [308](#)
- PixelSize_Bpp2, [308](#)
- PixelSize_Bpp20, [308](#)
- PixelSize_Bpp24, [308](#)
- PixelSize_Bpp30, [308](#)
- PixelSize_Bpp32, [308](#)
- PixelSize_Bpp36, [308](#)
- PixelSize_Bpp4, [308](#)
- PixelSize_Bpp48, [308](#)
- PixelSize_Bpp64, [308](#)
- PixelSize_Bpp8, [308](#)
- PixelSize_Bpp96, [308](#)
- PixelSizeEnums, [308](#)
- POEStatus_NotSupported, [309](#)
- POEStatus_PowerOff, [309](#)
- POEStatus_PowerOn, [309](#)
- POEStatusEnum, [309](#)
- PolarizationQuadrant, [309](#)
- RegionDestination_Stream0, [309](#)
- RegionDestination_Stream1, [309](#)
- RegionDestination_Stream2, [309](#)
- RegionDestinationEnums, [309](#)
- RegionMode_Off, [310](#)
- RegionMode_On, [310](#)
- RegionModeEnums, [309](#)
- RegionSelector_All, [310](#)
- RegionSelector_Region0, [310](#)
- RegionSelector_Region1, [310](#)
- RegionSelector_Region2, [310](#)
- RegionSelectorEnums, [310](#)
- RgbTransformLightSource_Cloudy6500K, [311](#)
- RgbTransformLightSource_CoolFluorescent4000K, [310](#)
- RgbTransformLightSource_Custom, [311](#)
- RgbTransformLightSource_Daylight5000K, [311](#)
- RgbTransformLightSource_General, [310](#)
- RgbTransformLightSource_Shade8000K, [311](#)
- RgbTransformLightSource_Tungsten2800K, [310](#)
- RgbTransformLightSource_WarmFluorescent3000K, [310](#)
- RgbTransformLightSourceEnums, [310](#)
- Scan3dCoordinateReferenceSelector_RotationX, [311](#)
- Scan3dCoordinateReferenceSelector_RotationY, [311](#)
- Scan3dCoordinateReferenceSelector_RotationZ, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationX, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationY, [311](#)
- Scan3dCoordinateReferenceSelector_TranslationZ, [311](#)
- Scan3dCoordinateReferenceSelectorEnums, [311](#)
- Scan3dCoordinateSelector_CoordinateA, [311](#)
- Scan3dCoordinateSelector_CoordinateB, [311](#)
- Scan3dCoordinateSelector_CoordinateC, [311](#)
- Scan3dCoordinateSelectorEnums, [311](#)
- Scan3dCoordinateSystem_Cartesian, [312](#)
- Scan3dCoordinateSystem_Cylindrical, [312](#)
- Scan3dCoordinateSystem_Spherical, [312](#)
- Scan3dCoordinateSystemEnums, [311](#)
- Scan3dCoordinateSystemReference_Anchor, [312](#)
- Scan3dCoordinateSystemReference_Transformed, [312](#)
- Scan3dCoordinateSystemReferenceEnums, [312](#)
- Scan3dCoordinateTransformSelector_RotationX, [312](#)
- Scan3dCoordinateTransformSelector_RotationY, [312](#)
- Scan3dCoordinateTransformSelector_RotationZ, [312](#)
- Scan3dCoordinateTransformSelector_TranslationX, [312](#)
- Scan3dCoordinateTransformSelector_TranslationY, [312](#)
- Scan3dCoordinateTransformSelector_TranslationZ, [312](#)
- Scan3dCoordinateTransformSelectorEnums, [312](#)
- Scan3dDistanceUnit_Inch, [313](#)
- Scan3dDistanceUnit_Millimeter, [313](#)
- Scan3dDistanceUnitEnums, [313](#)
- Scan3dOutputMode_CalibratedABC_Grid, [313](#)
- Scan3dOutputMode_CalibratedABC_PointCloud, [313](#)
- Scan3dOutputMode_CalibratedAC, [313](#)
- Scan3dOutputMode_CalibratedAC_Linescan, [313](#)
- Scan3dOutputMode_CalibratedC, [313](#)
- Scan3dOutputMode_CalibratedC_Linescan, [313](#)
- Scan3dOutputMode_DisparityC, [314](#)
- Scan3dOutputMode_DisparityC_Linescan, [314](#)
- Scan3dOutputMode_RectifiedC, [313](#)
- Scan3dOutputMode_RectifiedC_Linescan, [314](#)
- Scan3dOutputMode_UncalibratedC, [313](#)
- Scan3dOutputModeEnums, [313](#)
- SensorDigitizationTaps_Eight, [314](#)

- SensorDigitizationTaps_Four, [314](#)
- SensorDigitizationTaps_One, [314](#)
- SensorDigitizationTaps_Ten, [314](#)
- SensorDigitizationTaps_Three, [314](#)
- SensorDigitizationTaps_Two, [314](#)
- SensorDigitizationTapsEnums, [314](#)
- SensorShutterMode_Global, [314](#)
- SensorShutterMode_GlobalReset, [314](#)
- SensorShutterMode_Rolling, [314](#)
- SensorShutterModeEnums, [314](#)
- SensorTaps_Eight, [315](#)
- SensorTaps_Four, [315](#)
- SensorTaps_One, [315](#)
- SensorTaps_Ten, [315](#)
- SensorTaps_Three, [315](#)
- SensorTaps_Two, [315](#)
- SensorTapsEnums, [315](#)
- SequencerConfigurationMode_Off, [315](#)
- SequencerConfigurationMode_On, [315](#)
- SequencerConfigurationModeEnums, [315](#)
- SequencerConfigurationValid_No, [315](#)
- SequencerConfigurationValid_Yes, [315](#)
- SequencerConfigurationValidEnums, [315](#)
- SequencerMode_Off, [316](#)
- SequencerMode_On, [316](#)
- SequencerModeEnums, [315](#)
- SequencerSetValid_No, [316](#)
- SequencerSetValid_Yes, [316](#)
- SequencerSetValidEnums, [316](#)
- SequencerTriggerActivation_AnyEdge, [316](#)
- SequencerTriggerActivation_FallingEdge, [316](#)
- SequencerTriggerActivation_LevelHigh, [316](#)
- SequencerTriggerActivation_LevelLow, [316](#)
- SequencerTriggerActivation_RisingEdge, [316](#)
- SequencerTriggerActivationEnums, [316](#)
- SequencerTriggerSource_FrameStart, [317](#)
- SequencerTriggerSource_Off, [317](#)
- SequencerTriggerSourceEnums, [316](#)
- SerialPortBaudRate_Baud115200, [317](#)
- SerialPortBaudRate_Baud1200, [317](#)
- SerialPortBaudRate_Baud14400, [317](#)
- SerialPortBaudRate_Baud19200, [317](#)
- SerialPortBaudRate_Baud230400, [317](#)
- SerialPortBaudRate_Baud2400, [317](#)
- SerialPortBaudRate_Baud300, [317](#)
- SerialPortBaudRate_Baud38400, [317](#)
- SerialPortBaudRate_Baud460800, [317](#)
- SerialPortBaudRate_Baud4800, [317](#)
- SerialPortBaudRate_Baud57600, [317](#)
- SerialPortBaudRate_Baud600, [317](#)
- SerialPortBaudRate_Baud921600, [317](#)
- SerialPortBaudRate_Baud9600, [317](#)
- SerialPortBaudRateEnums, [317](#)
- SerialPortParity_Even, [317](#)
- SerialPortParity_Mark, [317](#)
- SerialPortParity_None, [317](#)
- SerialPortParity_Odd, [317](#)
- SerialPortParity_Space, [317](#)
- SerialPortParityEnums, [317](#)
- SerialPortSelector_SerialPort0, [318](#)
- SerialPortSelectorEnums, [318](#)
- SerialPortSource_Line0, [318](#)
- SerialPortSource_Line1, [318](#)
- SerialPortSource_Line2, [318](#)
- SerialPortSource_Line3, [318](#)
- SerialPortSource_Off, [318](#)
- SerialPortSourceEnums, [318](#)
- SerialPortStopBits_Bits1, [318](#)
- SerialPortStopBits_Bits1AndAHalf, [318](#)
- SerialPortStopBits_Bits2, [318](#)
- SerialPortStopBitsEnums, [318](#)
- SoftwareSignalSelector_SoftwareSignal0, [319](#)
- SoftwareSignalSelector_SoftwareSignal1, [319](#)
- SoftwareSignalSelector_SoftwareSignal2, [319](#)
- SoftwareSignalSelectorEnums, [318](#)
- SourceDataRange, [319](#)
- SourceSelector_All, [320](#)
- SourceSelector_Source0, [320](#)
- SourceSelector_Source1, [320](#)
- SourceSelector_Source2, [320](#)
- SourceSelectorEnums, [319](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ERROR, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_OK, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW, [234](#)
- SPINNAKER_BUFFER_OWNERSHIP_SYSTEM, [240](#)
- SPINNAKER_BUFFER_OWNERSHIP_USER, [240](#)
- SPINNAKER_CCM_APPLICATION_GENERIC, [241](#)
- SPINNAKER_CCM_APPLICATION_MICROSCOPY, [241](#)
- SPINNAKER_CCM_COLOR_SPACE_OFF, [241](#)
- SPINNAKER_CCM_COLOR_SPACE_SRGB, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_CLOUDY_6500K, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_COOL_FLUORESCENT_4000K, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_GENERAL, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_SHADE_8000K, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_SUNNY_5000K, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_TUNGSTEN_2800K, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_WARM_FLUORESCENT_3000K, [241](#)
- SPINNAKER_CCM_SENSOR_IMX250, [241](#)
- SPINNAKER_CCM_TYPE_ADVANCED, [242](#)

- SPINNAKER_CCM_TYPE_LINEAR, [242](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECT_FILTER, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_ALPHA, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_TIMEC, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NOISE, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGID, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEAK_DIRECTIONAL_FILTER, [251](#)
- SPINNAKER_ERR_ABORT, [274](#)
- SPINNAKER_ERR_ACCESS_DENIED, [274](#)
- SPINNAKER_ERR_BUFFER_TOO_SMALL, [274](#)
- SPINNAKER_ERR_BUSY, [274](#)
- SPINNAKER_ERR_CUSTOM_ID, [274](#)
- SPINNAKER_ERR_ERROR, [274](#)
- SPINNAKER_ERR_GENICAM_ACCESS, [274](#)
- SPINNAKER_ERR_GENICAM_BAD_ALLOCATION, [274](#)
- SPINNAKER_ERR_GENICAM_DYNAMIC_CAST, [274](#)
- SPINNAKER_ERR_GENICAM_GENERIC, [274](#)
- SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT, [274](#)
- SPINNAKER_ERR_GENICAM_LOGICAL, [274](#)
- SPINNAKER_ERR_GENICAM_OUT_OF_RANGE, [274](#)
- SPINNAKER_ERR_GENICAM_PROPERTY, [274](#)
- SPINNAKER_ERR_GENICAM_RUN_TIME, [274](#)
- SPINNAKER_ERR_GENICAM_TIMEOUT, [274](#)
- SPINNAKER_ERR_IM_COLOR_CONVERSION, [274](#)
- SPINNAKER_ERR_IM_CONVERT, [274](#)
- SPINNAKER_ERR_IM_COPY, [274](#)
- SPINNAKER_ERR_IM_DECOMPRESSION, [274](#)
- SPINNAKER_ERR_IM_HISTOGRAM_MEAN, [274](#)
- SPINNAKER_ERR_IM_HISTOGRAM_RANGE, [274](#)
- SPINNAKER_ERR_IM_MALLOC, [274](#)
- SPINNAKER_ERR_IM_MIN_MAX, [274](#)
- SPINNAKER_ERR_IM_NOT_SUPPORTED, [274](#)
- SPINNAKER_ERR_INVALID_ADDRESS, [274](#)
- SPINNAKER_ERR_INVALID_BUFFER, [274](#)
- SPINNAKER_ERR_INVALID_HANDLE, [274](#)
- SPINNAKER_ERR_INVALID_ID, [274](#)
- SPINNAKER_ERR_INVALID_INDEX, [274](#)
- SPINNAKER_ERR_INVALID_PARAMETER, [274](#)
- SPINNAKER_ERR_INVALID_VALUE, [274](#)
- SPINNAKER_ERR_IO, [274](#)
- SPINNAKER_ERR_NO_DATA, [274](#)
- SPINNAKER_ERR_NOT_AVAILABLE, [274](#)
- SPINNAKER_ERR_NOT_IMPLEMENTED, [274](#)
- SPINNAKER_ERR_NOT_INITIALIZED, [274](#)
- SPINNAKER_ERR_OUT_OF_MEMORY, [274](#)
- SPINNAKER_ERR_PARSING_CHUNK_DATA, [274](#)
- SPINNAKER_ERR_RESOURCE_EXHAUSTED, [274](#)
- SPINNAKER_ERR_RESOURCE_IN_USE, [274](#)
- SPINNAKER_ERR_SUCCESS, [274](#)
- SPINNAKER_ERR_TIMEOUT, [274](#)
- SPINNAKER_EVENT_ARRIVAL_REMOVAL, [275](#)
- SPINNAKER_EVENT_DEVICE, [275](#)
- SPINNAKER_EVENT_DEVICE_SPECIFIC, [275](#)
- SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL, [275](#)
- SPINNAKER_EVENT_INTERFACE_DEVICE, [275](#)
- SPINNAKER_EVENT_NEW_BUFFER, [275](#)
- SPINNAKER_EVENT_NEW_BUFFER_LIST, [275](#)
- SPINNAKER_EVENT_UNKNOWN, [275](#)
- SPINNAKER_HEATMAP_COLOR_BLACK, [287](#)
- SPINNAKER_HEATMAP_COLOR_BLUE, [287](#)
- SPINNAKER_HEATMAP_COLOR_CYAN, [287](#)
- SPINNAKER_HEATMAP_COLOR_GREEN, [287](#)
- SPINNAKER_HEATMAP_COLOR_RED, [287](#)
- SPINNAKER_HEATMAP_COLOR_WHITE, [287](#)
- SPINNAKER_HEATMAP_COLOR_YELLOW, [287](#)
- SPINNAKER_IMAGE_FILE_FORMAT_BMP, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG12_C, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PGM, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PNG, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PPM, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_RAW, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_EXTENDED_CHUNK, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_IMAGE, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSLESS_COMPRESSED, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSY_COMPRESSED, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_UNKNOWN, [290](#)

- [289](#)
- SPINNAKER_IMAGE_SCALING_ALGORITHM_NEAREST_NEIGHBOR, [309](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID, [309](#)
- [291](#)
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED, [319](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE, [319](#)
- [291](#)
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW, [319](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT, [319](#)
- [291](#)
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT, [320](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER, [320](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS, [320](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER, [320](#)
- [291](#)
- SPINNAKER_IMAGE_STATUS_NO_ERROR, [290](#)
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES, [320](#)
- [291](#)
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT, [320](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT, [323](#)
- [290](#)
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR, [323](#)
- [290](#)
- SPINNAKER_INT_TYPE_FLOAT32, [308](#)
- SPINNAKER_INT_TYPE_INT16, [308](#)
- SPINNAKER_INT_TYPE_INT8, [308](#)
- SPINNAKER_INT_TYPE_UINT10, [308](#)
- SPINNAKER_INT_TYPE_UINT10P, [308](#)
- SPINNAKER_INT_TYPE_UINT10p, [308](#)
- SPINNAKER_INT_TYPE_UINT12, [308](#)
- SPINNAKER_INT_TYPE_UINT12P, [308](#)
- SPINNAKER_INT_TYPE_UINT12p, [308](#)
- SPINNAKER_INT_TYPE_UINT14, [308](#)
- SPINNAKER_INT_TYPE_UINT16, [308](#)
- SPINNAKER_INT_TYPE_UINT8, [308](#)
- SPINNAKER_INT_TYPE_UNKNOWN, [308](#)
- SPINNAKER_LOG_LEVEL_ALERT, [320](#)
- SPINNAKER_LOG_LEVEL_CRIT, [320](#)
- SPINNAKER_LOG_LEVEL_DEBUG, [320](#)
- SPINNAKER_LOG_LEVEL_ERROR, [320](#)
- SPINNAKER_LOG_LEVEL_FATAL, [320](#)
- SPINNAKER_LOG_LEVEL_INFO, [320](#)
- SPINNAKER_LOG_LEVEL_NOTICE, [320](#)
- SPINNAKER_LOG_LEVEL_NOTSET, [320](#)
- SPINNAKER_LOG_LEVEL_OFF, [320](#)
- SPINNAKER_LOG_LEVEL_WARN, [320](#)
- SPINNAKER_POLARIZATION_QUADRANT_I0, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I135, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I45, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I90, [309](#)
- SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_DATA_RANGE, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_MIN_IMAGE_I, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_IMAGE_MIN_ABSOLUTE_I, [319](#)
- SPINNAKER_STATISTICS_CHANNEL_BLUE, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_GREEN, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_GREY, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_HUE, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_RED, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_SATURATION, [320](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_JPEG, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS, [323](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_FILE, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_H264, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESS, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED, [327](#)

- [327](#)
- [SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED](#), [327](#)
- [SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART](#), [327](#)
- [SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA](#), [327](#)
- [SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN](#), [327](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_CUSTOM](#), [328](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV](#), [328](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC](#), [328](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT](#), [328](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT](#), [328](#)
- [SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN](#), [328](#)
- [SpinnakerLogLevel](#), [320](#)
- [StatisticsChannel](#), [320](#)
- [StreamBufferCountMode_Manual](#), [321](#)
- [StreamBufferCountModeEnum](#), [321](#)
- [StreamBufferHandlingMode_NewestFirst](#), [321](#)
- [StreamBufferHandlingMode_NewestOnly](#), [321](#)
- [StreamBufferHandlingMode_OldestFirst](#), [321](#)
- [StreamBufferHandlingMode_OldestFirstOverwrite](#), [321](#)
- [StreamBufferHandlingModeEnum](#), [321](#)
- [StreamMode_LWF](#), [322](#)
- [StreamMode_MVA](#), [322](#)
- [StreamMode_Socket](#), [322](#)
- [StreamModeEnum](#), [321](#)
- [StreamType_CameraLink](#), [322](#)
- [StreamType_CameraLinkHS](#), [322](#)
- [StreamType_CoaXPress](#), [322](#)
- [StreamType_Custom](#), [322](#)
- [StreamType_GigEVision](#), [322](#)
- [StreamType_USB3Vision](#), [322](#)
- [StreamTypeEnum](#), [322](#)
- [TestPattern_Increment](#), [322](#)
- [TestPattern_Off](#), [322](#)
- [TestPattern_SensorTestPattern](#), [322](#)
- [TestPatternEnums](#), [322](#)
- [TestPatternGeneratorSelector_PipelineStart](#), [323](#)
- [TestPatternGeneratorSelector_Sensor](#), [323](#)
- [TestPatternGeneratorSelectorEnums](#), [323](#)
- [TIFFCompressionMethod](#), [323](#)
- [TimerSelector_Timer0](#), [325](#)
- [TimerSelector_Timer1](#), [325](#)
- [TimerSelector_Timer2](#), [325](#)
- [TimerSelectorEnums](#), [323](#)
- [TimerStatus_TimerActive](#), [325](#)
- [TimerStatus_TimerCompleted](#), [325](#)
- [TimerStatus_TimerIdle](#), [325](#)
- [TimerStatus_TimerTriggerWait](#), [325](#)
- [TimerStatusEnums](#), [325](#)
- [TimerTriggerActivation_AnyEdge](#), [325](#)
- [TimerTriggerActivation_FallingEdge](#), [325](#)
- [TimerTriggerActivation_LevelHigh](#), [325](#)
- [TimerTriggerActivation_LevelLow](#), [325](#)
- [TimerTriggerActivation_RisingEdge](#), [325](#)
- [TimerTriggerActivationEnums](#), [325](#)
- [TimerTriggerSource_AcquisitionEnd](#), [326](#)
- [TimerTriggerSource_AcquisitionStart](#), [326](#)
- [TimerTriggerSource_AcquisitionTrigger](#), [326](#)
- [TimerTriggerSource_Action0](#), [326](#)
- [TimerTriggerSource_Action1](#), [327](#)
- [TimerTriggerSource_Action2](#), [327](#)
- [TimerTriggerSource_Counter0End](#), [326](#)
- [TimerTriggerSource_Counter0Start](#), [326](#)
- [TimerTriggerSource_Counter1End](#), [326](#)
- [TimerTriggerSource_Counter1Start](#), [326](#)
- [TimerTriggerSource_Counter2End](#), [326](#)
- [TimerTriggerSource_Counter2Start](#), [326](#)
- [TimerTriggerSource_Encoder0](#), [326](#)
- [TimerTriggerSource_Encoder1](#), [326](#)
- [TimerTriggerSource_Encoder2](#), [326](#)
- [TimerTriggerSource_ExposureEnd](#), [326](#)
- [TimerTriggerSource_ExposureStart](#), [326](#)
- [TimerTriggerSource_FrameBurstEnd](#), [326](#)
- [TimerTriggerSource_FrameBurstStart](#), [326](#)
- [TimerTriggerSource_FrameEnd](#), [326](#)
- [TimerTriggerSource_FrameStart](#), [326](#)
- [TimerTriggerSource_FrameTrigger](#), [326](#)
- [TimerTriggerSource_Line0](#), [326](#)
- [TimerTriggerSource_Line1](#), [326](#)
- [TimerTriggerSource_Line2](#), [326](#)
- [TimerTriggerSource_LineEnd](#), [326](#)
- [TimerTriggerSource_LineStart](#), [326](#)
- [TimerTriggerSource_LineTrigger](#), [326](#)
- [TimerTriggerSource_LinkTrigger0](#), [327](#)
- [TimerTriggerSource_LinkTrigger1](#), [327](#)
- [TimerTriggerSource_LinkTrigger2](#), [327](#)
- [TimerTriggerSource_Off](#), [326](#)
- [TimerTriggerSource_SoftwareSignal0](#), [326](#)
- [TimerTriggerSource_SoftwareSignal1](#), [326](#)
- [TimerTriggerSource_SoftwareSignal2](#), [326](#)
- [TimerTriggerSource_Timer0End](#), [326](#)
- [TimerTriggerSource_Timer0Start](#), [326](#)
- [TimerTriggerSource_Timer1End](#), [326](#)
- [TimerTriggerSource_Timer1Start](#), [326](#)
- [TimerTriggerSource_Timer2End](#), [326](#)
- [TimerTriggerSource_Timer2Start](#), [326](#)
- [TimerTriggerSource_UserOutput0](#), [326](#)
- [TimerTriggerSource_UserOutput1](#), [326](#)
- [TimerTriggerSource_UserOutput2](#), [326](#)
- [TimerTriggerSourceEnums](#), [325](#)
- [TLPayloadType](#), [327](#)
- [TLPixelFormatNamespace](#), [327](#)
- [TLType_CameraLink](#), [328](#)
- [TLType_CameraLinkHS](#), [328](#)
- [TLType_CoaXPress](#), [328](#)
- [TLType_Custom](#), [328](#)

- UserSetSelectorEnums, 335
- WhiteClipSelector_All, 336
- WhiteClipSelector_Blue, 336
- WhiteClipSelector_Green, 336
- WhiteClipSelector_Red, 336
- WhiteClipSelector_Tap1, 336
- WhiteClipSelector_Tap2, 336
- WhiteClipSelector_U, 336
- WhiteClipSelector_V, 336
- WhiteClipSelector_Y, 336
- WhiteClipSelectorEnums, 335
- Spinnaker Classes, 41
- Spinnaker Definitions, 78
- Spinnaker EventHandler Classes, 51
- Spinnaker GenApi Classes, 100
- Spinnaker GenApi Enums, 185
- Spinnaker GenApi Interfaces, 102
- Spinnaker GenApi Utilities, 130
- Spinnaker Headers, 76
- Spinnaker Platform, 80
- Spinnaker QuickSpin Classes, 86
- Spinnaker Video Class, 81
- Spinnaker Video Definitions, 82
- Spinnaker.h, 77
- Spinnaker::GenApi, 338
 - _ClearXMLCache, 363
 - _Connect, 363
 - _CycleDetectAccessMode, 356
 - _EAccessMode, 356
 - _ECachingMode, 356
 - _ECallbackType, 357
 - _EDisplayNotation, 357
 - _EEndianess, 357
 - _EGenApiSchemaVersion, 358
 - _EIncMode, 358
 - _EInputDirection, 358
 - _EInterfaceType, 359
 - _ELinkType, 359
 - _ENamespace, 359
 - _ERepresentation, 360
 - _ESign, 360
 - _ESlope, 360
 - _EStandardNamespace, 361
 - _EVisibility, 361
 - _EXMLValidation, 361
 - _EYesNo, 362
 - _GetDeviceName, 363
 - _GetNode, 364
 - _GetNodes, 364
 - _GetSupportedSchemaVersions, 364
 - _InvalidateNodes, 364
 - _LoadXMLFromFile, 364
 - _LoadXMLFromFileInject, 364
 - _LoadXMLFromString, 364
 - _LoadXMLFromStringInject, 365
 - _LoadXMLFromZIPData, 365
 - _LoadXMLFromZIPFile, 365
 - _Poll, 365
 - _Undefined, 358
 - _UndefinedAccessMode, 356
 - _UndefinedCachingMode, 357
 - _UndefinedEDisplayNotation, 357
 - _UndefinedESlope, 361
 - _UndefinedEXMLValidation, 362
 - _UndefinedEndian, 358
 - _UndefinedNameSpace, 360
 - _UndefinedRepresentation, 360
 - _UndefinedSign, 360
 - _UndefinedStandardNameSpace, 361
 - _UndefinedVisibility, 361
 - _UndefinedYesNo, 362
- Address, 388
- Automatic, 361
- Beginner, 361
- BigEndian, 358
- Boolean, 360
- CacheChunkData, 365
- CacheUsage_Automatic, 362
- CacheUsage_ForceRead, 362
- CacheUsage_ForceWrite, 362
- CacheUsage_Ignore, 362
- CallbackHandleType, 353
- CastToDestroy, 365
- CBooleanRef, 353
- cbPostInsideLock, 357
- cbPostOutsideLock, 357
- CCategoryRef, 353
- CCommandRef, 354
- CEnumEntryRef, 354
- CEnumerationRef, 354
- CFloatRef, 354
- CIntegerRef, 354
- CL, 361
- CNodeMapRef, 354
- CNodeRef, 354
- COMMAND_MAGIC, 388
- Connect, 366
- ContentType_Xml, 363
- ContentType_ZippedXml, 363
- CPortRecorderRef, 354
- CPortRef, 355
- CRegisterRef, 355
- CSelectorRef, 355
- CStringRef, 355
- ctDependingNodes, 359
- ctInvalidatingChildren, 359
- ctParentNodes, 359
- ctReadingChildren, 359
- ctTerminalNodes, 359
- ctWritingChildren, 359
- Custom, 360
- CValueRef, 355
- Decreasing, 361
- Deregister, 366
- DeregisterCallback, 366
- EatComments, 366

- ECacheUsage_t, 362
- EContentType_t, 363
- Expert, 361
- ExtractIndependentSubtree, 367
- fixedIncrement, 358
- fnAutomatic, 357
- fnFixed, 357
- fnScientific, 357
- FromString, 367
- GENCP_COMMAND_HEADER_SIZE, 388
- GENCP_EVENT_BASIC_SIZE, 388
- GENCP_EVENT_CMD_ID, 388
- Get, 367
- GetAddress, 368
- GetAlias, 368
- GetCachingMode, 368
- GetCastAlias, 368
- GetChildren, 368
- GetCookie, 369
- GetCurrentEntry, 369
- GetDescription, 369
- GetDeviceName, 369
- GetDeviceVersion, 369
- GetDisplayName, 369
- GetDisplayNotation, 370
- GetDisplayPrecision, 370
- GetDocuURL, 370
- GetEntries, 370
- GetEntry, 370
- GetEntryByName, 371
- GetEventID, 371
- GetGenApiVersion, 371
- GetInc, 371
- GetIncMode, 371
- GetIntValue, 371
- GetLength, 372
- GetListOfValidValues, 372
- GetLock, 372
- GetMax, 372
- GetMaxLength, 372
- GetMin, 373
- GetNameSpace, 373
- GetNode, 373
- GetNodeMap, 373
- GetNumericValue, 373
- GetNumNodes, 373
- GetParents, 373
- GetPollingTime, 374
- GetPrincipalInterfaceType, 374
- GetProductGuid, 374
- GetProperty, 374
- GetPropertyNames, 374
- GetRepresentation, 374
- GetSchemaVersion, 375
- GetSelectedFeatures, 375
- GetSelectingFeatures, 375
- GetSelectorList, 375
- GetStandardNameSpace, 375
- GetSupportedSchemaVersions, 376
- GetSwapEndianness, 376
- GetSymbolic, 376
- GetToolTip, 376
- GetUnit, 376
- GetValue, 376
- GetVendorName, 377
- GetVersionGuid, 377
- GetVisibility, 377
- GEV, 361
- Guru, 361
- GVCP_MESSAGE_TAGS, 363
- HasInc, 377
- HexNumber, 360
- IBase, 389
- IBoolean, 389
- ICategory, 389
- IChunkPort, 389
- ICommand, 389
- IDestroy, 389
- IDevFileStream, 355
- IDeviceInfo, 390
- idFrom, 358
- idNone, 358
- idTo, 358
- IEnumEntry, 390
- IEnumeration, 390
- IEnumerationT, 390
- IEnumReference, 390
- IFloat, 391
- IIDC, 361
- IInteger, 391
- ImposeAccessMode, 377
- ImposeMax, 377, 378
- ImposeMin, 378
- ImposeVisibility, 378
- Increasing, 361
- INode, 391
- INodeMap, 391
- INodeMapDyn, 391
- intflBase, 359
- intflBoolean, 359
- intflCategory, 359
- intflCommand, 359
- intflEnumEntry, 359
- intflEnumeration, 359
- intflFloat, 359
- intflInteger, 359
- intflIPort, 359
- intflRegister, 359
- intflString, 359
- intflValue, 359
- Invalidate, 392
- InvalidateNode, 378
- InvalidateNodes, 378
- Invisible, 361
- IPersistScript, 392
- IPort, 392

- IPortConstruct, [392](#)
- IPortRecorder, [392](#)
- IPortReplay, [392](#)
- IPortWriteList, [393](#)
- IPv4Address, [360](#)
- IRReference, [393](#)
- IRegister, [393](#)
- IsAccessModeCacheable, [379](#)
- IsCachable, [379](#)
- IsDeprecated, [379](#)
- IsDone, [379](#)
- ISelector, [393](#)
- ISelectorDigit, [393](#)
- IsFeature, [379](#)
- IsSelfClearing, [380](#)
- IsStreamable, [380](#)
- ISString, [394](#)
- IsValueCacheValid, [380](#)
- IValue, [394](#)
- Length, [394](#)
- Linear, [360](#)
- listIncrement, [358](#)
- LittleEndian, [358](#)
- LoadXMLFromFile, [380](#)
- LoadXMLFromFileInject, [380](#)
- LoadXMLFromString, [380](#)
- LoadXMLFromStringInject, [381](#)
- LoadXMLFromZIPData, [381](#)
- LoadXMLFromZIPFile, [381](#)
- Logarithmic, [360](#)
- MACAddress, [360](#)
- make_NodeCallback, [381](#)
- MergeXMLFiles, [382](#)
- NA, [356](#)
- NI, [356](#)
- No, [362](#)
- NoCache, [357](#)
- NodeList_t, [355](#)
- noIncrement, [358](#)
- None, [361](#)
- ODevFileStream, [356](#)
- operator!=, [382](#)
- operator<<, [382](#)
- operator>>, [384](#)
- operator*, [382](#)
- operator(), [382](#)
- operator=, [383](#)
- operator==, [384](#)
- PersistFeature, [384](#)
- Poll, [384](#)
- PreprocessXMLFromFile, [384](#)
- PreprocessXMLFromZIPFile, [385](#)
- PureNumber, [360](#)
- Register, [385](#)
- RegisterCallback, [385](#)
- Replay, [386](#)
- Restore, [386](#)
- RO, [356](#)
- RW, [356](#)
- SET_GUID, [386](#)
- SetCookie, [386](#)
- SetIntValue, [386](#)
- SetNext, [387](#)
- SetNumEnums, [387](#)
- Signed, [360](#)
- Standard, [360](#)
- StopRecording, [387](#)
- StringList_t, [356](#)
- TAG_EVENT_CMD, [363](#)
- TAG_EVENTDATA_CMD, [363](#)
- ToString, [387](#)
- U3V_EVENT_PREFIX, [394](#)
- Unsigned, [360](#)
- USB, [361](#)
- v1_0, [358](#)
- v1_1, [358](#)
- Varying, [361](#)
- Verify, [394](#)
- WO, [356](#)
- Write, [388](#)
- WriteAround, [357](#)
- WriteThrough, [357](#)
- xvAll, [362](#)
- xvCycles, [362](#)
- xvDefault, [362](#)
- xvLoad, [362](#)
- xvSFNC, [362](#)
- Yes, [362](#)
- Spinnaker::GenICam, [395](#)
 - DoesEnvironmentVariableExist, [396](#)
 - GetFiles, [396](#)
 - GetGenICamCacheFolder, [396](#)
 - GetGenICamCLProtocolFolder, [397](#)
 - GetGenICamLogConfig, [397](#)
 - getline, [397](#)
 - GetModulePathFromFunction, [397](#)
 - GetValueOfEnvironmentVariable, [398](#)
 - INTEGRAL_CAST, [398](#)
 - INTEGRAL_CAST2, [398](#)
 - ReplaceEnvironmentVariables, [399](#)
 - SetGenICamCacheFolder, [399](#)
 - SetGenICamCLProtocolFolder, [399](#)
 - SetGenICamLogConfig, [399](#)
 - ThrowBadAlloc, [399](#)
 - Tokenize, [399](#)
 - UrlDecode, [400](#)
 - UrlEncode, [400](#)
- Spinnaker::Video, [400](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ACTION_LATE
 - Spinnaker, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_ERROR
 - Spinnaker, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_NO_REF_TIME
 - Spinnaker, [234](#)
- SPINNAKER_ACTION_COMMAND_STATUS_OK
 - Spinnaker, [234](#)

- SPINNAKER_ACTION_COMMAND_STATUS_OVERFLOW
Spinnaker, [234](#)
- SPINNAKER_API
SpinnakerPlatform.h, [1352](#)
- SPINNAKER_API_ABSTRACT
SpinnakerPlatform.h, [1352](#)
- SPINNAKER_BUFFER_OWNERSHIP_SYSTEM
Spinnaker, [240](#)
- SPINNAKER_BUFFER_OWNERSHIP_USER
Spinnaker, [240](#)
- SPINNAKER_CCM_APPLICATION_GENERIC
Spinnaker, [241](#)
- SPINNAKER_CCM_APPLICATION_MICROSCOPY
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_SPACE_OFF
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_SPACE_SRGB
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_CLOUDY_6500K
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_COOL_FLUORESCENCE_4000K
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_GENERAL
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_SHADE_8000K
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_SUNNY_5000K
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_TUNGSTEN_2800K
Spinnaker, [241](#)
- SPINNAKER_CCM_COLOR_TEMP_WARM_FLUORESCENCE_3000K
Spinnaker, [241](#)
- SPINNAKER_CCM_SENSOR_IMX250
Spinnaker, [241](#)
- SPINNAKER_CCM_TYPE_ADVANCED
Spinnaker, [242](#)
- SPINNAKER_CCM_TYPE_LINEAR
Spinnaker, [242](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_BILINEAR
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_DIRECT_FILTER
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_EDGE_SENSING
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_HQ_SPLINE
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_IPP
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR_INVALID
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_NONSPONSOR
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_RIGID
Spinnaker, [251](#)
- SPINNAKER_COLOR_PROCESSING_ALGORITHM_WEAK_DIRECTIONAL_FILTER
Spinnaker, [251](#)
- SPINNAKER_ERR_ABORT
Spinnaker, [274](#)
- SPINNAKER_ERR_ACCESS_DENIED
Spinnaker, [274](#)
- SPINNAKER_ERR_BUFFER_TOO_SMALL
Spinnaker, [274](#)
- SPINNAKER_ERR_BUSY
Spinnaker, [274](#)
- SPINNAKER_ERR_CUSTOM_ID
Spinnaker, [274](#)
- SPINNAKER_ERR_ERROR
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_ACCESS
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_BAD_ALLOCATION
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_DYNAMIC_CAST
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_GENERIC
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_INVALID_ARGUMENT
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_LOGICAL
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_OUT_OF_RANGE
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_PROPERTY
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_RUN_TIME
Spinnaker, [274](#)
- SPINNAKER_ERR_GENICAM_TIMEOUT
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_COLOR_CONVERSION
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_CONVERT
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_COPY
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_DECOMPRESSION
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_HISTOGRAM_MEAN
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_HISTOGRAM_RANGE
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_MALLOC
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_MIN_MAX
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_NOT_SUPPORTED
Spinnaker, [274](#)
- SPINNAKER_ERR_IM_INVALID_ADDRESS
Spinnaker, [274](#)
- SPINNAKER_ERR_INVALID_BUFFER
Spinnaker, [274](#)
- SPINNAKER_ERR_INVALID_HANDLE
Spinnaker, [274](#)
- SPINNAKER_ERR_INVALID_FILTER
Spinnaker, [274](#)

- SPINNAKER_ERR_INVALID_INDEX
 - Spinnaker, [274](#)
- SPINNAKER_ERR_INVALID_PARAMETER
 - Spinnaker, [274](#)
- SPINNAKER_ERR_INVALID_VALUE
 - Spinnaker, [274](#)
- SPINNAKER_ERR_IO
 - Spinnaker, [274](#)
- SPINNAKER_ERR_NO_DATA
 - Spinnaker, [274](#)
- SPINNAKER_ERR_NOT_AVAILABLE
 - Spinnaker, [274](#)
- SPINNAKER_ERR_NOT_IMPLEMENTED
 - Spinnaker, [274](#)
- SPINNAKER_ERR_NOT_INITIALIZED
 - Spinnaker, [274](#)
- SPINNAKER_ERR_OUT_OF_MEMORY
 - Spinnaker, [274](#)
- SPINNAKER_ERR_PARSING_CHUNK_DATA
 - Spinnaker, [274](#)
- SPINNAKER_ERR_RESOURCE_EXHAUSTED
 - Spinnaker, [274](#)
- SPINNAKER_ERR_RESOURCE_IN_USE
 - Spinnaker, [274](#)
- SPINNAKER_ERR_SUCCESS
 - Spinnaker, [274](#)
- SPINNAKER_ERR_TIMEOUT
 - Spinnaker, [274](#)
- SPINNAKER_EVENT_ARRIVAL_REMOVAL
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_DEVICE
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_DEVICE_SPECIFIC
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_LOGGING_EVENT
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_NEW_BUFFER
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_NEW_BUFFER_LIST
 - Spinnaker, [275](#)
- SPINNAKER_EVENT_UNKNOWN
 - Spinnaker, [275](#)
- SPINNAKER_EXCEPTION
 - ExceptionHandler.cpp, [1431](#)
- SPINNAKER_HEATMAP_COLOR_BLACK
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_BLUE
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_CYAN
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_GREEN
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_RED
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_WHITE
 - Spinnaker, [287](#)
- SPINNAKER_HEATMAP_COLOR_YELLOW
 - Spinnaker, [287](#)
- SPINNAKER_IMAGE_FILE_FORMAT_BMP
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FORCE_32BITS
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG12_C
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_JPEG2000
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PGM
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PNG
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_PPM
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_RAW
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_FILE_FORMAT_TIFF
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_EXTENDED_CHUNK
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_IMAGE
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSLESS_COMPRESSED
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_LOSSY_COMPRESSED
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_PAYLOAD_TYPE_UNKNOWN
 - Spinnaker, [289](#)
- SPINNAKER_IMAGE_SCALING_ALGORITHM_NEAREST_NEIGHBOR
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_CHUNK_DATA_INVALID
 - Spinnaker, [291](#)
- SPINNAKER_IMAGE_STATUS_CRC_CHECK_FAILED
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_DATA_INCOMPLETE
 - Spinnaker, [291](#)
- SPINNAKER_IMAGE_STATUS_DATA_OVERFLOW
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_INFO_INCONSISTENT
 - Spinnaker, [291](#)
- SPINNAKER_IMAGE_STATUS_LEADER_BUFFER_SIZE_INCONSISTENT
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_LEADER
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_PACKETS
 - Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_MISSING_TRAILER
 - Spinnaker, [291](#)

- SPINNAKER_IMAGE_STATUS_NO_ERROR
Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_NO_SYSTEM_RESOURCES
Spinnaker, [291](#)
- SPINNAKER_IMAGE_STATUS_PACKETID_INCONSISTENT
Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_TRAILER_BUFFER_SIZE_INCONSISTENT
Spinnaker, [290](#)
- SPINNAKER_IMAGE_STATUS_UNKNOWN_ERROR
Spinnaker, [290](#)
- SPINNAKER_INT_TYPE_FLOAT32
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_INT16
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_INT8
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT10
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT10P
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT10p
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT12
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT12P
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT12p
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT14
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT16
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UINT8
Spinnaker, [308](#)
- SPINNAKER_INT_TYPE_UNKNOWN
Spinnaker, [308](#)
- SPINNAKER_LOCAL
SpinnakerPlatform.h, [1353](#)
- SPINNAKER_LOG_LEVEL_ALERT
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_CRIT
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_DEBUG
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_ERROR
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_FATAL
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_INFO
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_NOTICE
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_NOTSET
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_OFF
Spinnaker, [320](#)
- SPINNAKER_LOG_LEVEL_WARN
Spinnaker, [320](#)
- SPINNAKER_POLARIZATION_QUADRANT_I0
Spinnaker, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I135
Spinnaker, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I45
Spinnaker, [309](#)
- SPINNAKER_POLARIZATION_QUADRANT_I90
Spinnaker, [309](#)
- SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_DATA_RANGE
Spinnaker, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_ABSOLUTE_MIN_IMAGE_MAX
Spinnaker, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE
Spinnaker, [319](#)
- SPINNAKER_SOURCE_DATA_RANGE_IMAGE_MIN_ABSOLUTE_MAX
Spinnaker, [319](#)
- SPINNAKER_STATISTICS_CHANNEL_BLUE
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_GREEN
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_GREY
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_HUE
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_LIGHTNESS
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_NUM_CHANNELS
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_RED
Spinnaker, [320](#)
- SPINNAKER_STATISTICS_CHANNEL_SATURATION
Spinnaker, [320](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_ADOBE_DEFLATE
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX3
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_CCITTFAX4
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_DEFLATE
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_JPEG
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_LZW
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_NONE
Spinnaker, [323](#)
- SPINNAKER_TIFF_COMPRESS_METHOD_PACKBITS
Spinnaker, [323](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_DATA
Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_CHUNK_ONLY
Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_CUSTOM_ID
Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_DEVICE_SPECIFIC
Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_FILE
Spinnaker, [327](#)

- SPINNAKER_TLPAYLOAD_TYPE_H264
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_IMAGE
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG2000
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_JPEG_LOSSLESS_COMPRESSED
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSLESS_COMPRESSED
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_LOSSY_COMPRESSED
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_MULTI_PART
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_RAW_DATA
 - Spinnaker, [327](#)
- SPINNAKER_TLPAYLOAD_TYPE_UNKNOWN
 - Spinnaker, [327](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_CUSTOM_ID
 - Spinnaker, [328](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_GEV
 - Spinnaker, [328](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_IIDC
 - Spinnaker, [328](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_16BIT
 - Spinnaker, [328](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_PFNC_32BIT
 - Spinnaker, [328](#)
- SPINNAKER_TLPIXELFORMAT_NAMESPACE_UNKNOWN
 - Spinnaker, [328](#)
- SpinnakerDirectShow.h, [79](#)
 - BeginAcquisition, [1336](#)
 - BooleanGetValue, [1336](#)
 - BooleanSetValue, [1338](#)
 - bufferSize, [1351](#)
 - CommandExecute, [1338](#)
 - EndAcquisition, [1338](#)
 - entryExist, [1351](#)
 - entryIndex, [1351](#)
 - enumerationEntryName, [1351](#)
 - EnumerationGetEntry, [1339](#)
 - EnumerationGetNumEntries, [1339](#)
 - enumerationName, [1351](#)
 - EnumerationSetEntry, [1340](#)
 - FloatGetInc, [1340](#)
 - FloatGetIncMode, [1340](#)
 - FloatGetMax, [1341](#)
 - FloatGetMin, [1341](#)
 - FloatGetValue, [1341](#)
 - FloatSetValue, [1342](#)
 - GetCameraInfo, [1342](#)
 - GetSelectedCameraIndex, [1343](#)
 - IID_ISpinnakerInterface, [1352](#)
 - IntegerGetInc, [1343](#)
 - IntegerGetIncMode, [1343](#)
 - IntegerGetMax, [1344](#)
 - IntegerGetMin, [1344](#)
 - IntegerGetValue, [1345](#)
 - IntegerSetValue, [1345](#)
 - IsStreaming, [1345](#)
 - MAX_LENGTH, [1352](#)
 - NodeGetDisplayName, [1346](#)
 - NodeGetType, [1346](#)
 - NodesAvailable, [1347](#)
 - NodesImplemented, [1347](#)
 - NodesReadable, [1347](#)
 - NodesWritable, [1348](#)
 - NodeMapGetNodeAtIndex, [1348](#)
 - NodeMapGetNumNodes, [1349](#)
 - NodeToString, [1349](#)
 - SetSelectedCameraIndex, [1349](#)
 - StringGetValue, [1350](#)
 - StringSetValue, [1350](#)
 - SpinnakerLogLevel
 - Spinnaker, [320](#)
 - SpinnakerPlatform.h
 - SPINNAKER_API, [1352](#)
 - SPINNAKER_API_ABSTRACT, [1352](#)
 - SPINNAKER_LOCAL, [1353](#)
 - SpinTestCamera, [1088](#)
 - SpinTestCamera Class, [180](#)
 - SpinUpdate.h
 - GetErrorMessage, [1354](#)
 - SetMessageCallback, [1354](#)
 - SetProgressCallback, [1354](#)
 - SPINUPDATE_API, [1354](#)
 - UpdateFirmware, [1354](#)
 - UpdateFirmwareConsole, [1354](#)
 - UpdateFirmwareGUI, [1355](#)
 - UpdaterMessageCallback, [1355](#)
 - UpdaterProgressCallback, [1355](#)
 - SPINUPDATE_API
 - SpinUpdate.h, [1354](#)
 - SpinVideo, [1088](#)
 - ~SpinVideo, [1089](#)
 - Append, [1089](#)
 - Close, [1090](#)
 - Open, [1090](#), [1092](#)
 - SetMaximumFileSize, [1092](#)
 - SpinVideo, [1089](#)
 - src/Acquisition/Acquisition.cpp, [1365](#)
 - src/Acquisition/resource.h, [1368](#)
 - src/Acquisition/stdafx.cpp, [1368](#)
 - src/Acquisition/stdafx.h, [1377](#)
 - src/Acquisition/targetver.h, [1393](#)
 - src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.h, [1407](#)
 - src/AcquisitionMultipleCameraRecovery/resource.h, [1368](#)
 - src/AcquisitionMultipleCamerasWriteToFile/AcquisitionMultipleCamerasWriteToFile.h, [1410](#)
 - src/AcquisitionMultipleCamerasWriteToFile/resource.h, [1368](#)

src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp, 1412
 src/AcquisitionMultipleThread/resource.h, 1368
 src/ActionCommand/ActionCommand.cpp, 1414
 src/ActionCommand/resource.h, 1368
 src/ActionCommand/stdafx.cpp, 1369
 src/ActionCommand/stdafx.h, 1378
 src/ActionCommand/targetver.h, 1394
 src/BufferHandling/BufferHandling.cpp, 1416
 src/BufferHandling/resource.h, 1368
 src/BufferHandling/stdafx.cpp, 1369
 src/BufferHandling/stdafx.h, 1379
 src/BufferHandling/targetver.h, 1395
 src/ChunkData/ChunkData.cpp, 1418
 src/ChunkData/resource.h, 1368
 src/Compression/Compression.cpp, 1421
 src/Compression/resource.h, 1368
 src/CounterAndTimer/CounterAndTimer.cpp, 1424
 src/CounterAndTimer/resource.h, 1368
 src/CounterAndTimer/stdafx.cpp, 1370
 src/CounterAndTimer/stdafx.h, 1380
 src/CounterAndTimer/targetver.h, 1396
 src/DeviceEvents/DeviceEvents.cpp, 1425
 src/DeviceEvents/resource.h, 1368
 src/DeviceEvents/stdafx.cpp, 1370
 src/DeviceEvents/stdafx.h, 1381
 src/DeviceEvents/targetver.h, 1397
 src/Enumeration/Enumeration.cpp, 1428
 src/Enumeration/resource.h, 1368
 src/Enumeration/stdafx.cpp, 1371
 src/Enumeration/stdafx.h, 1382
 src/Enumeration/targetver.h, 1398
 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp, 1429
 src/Enumeration_QuickSpin/resource.h, 1368
 src/Enumeration_QuickSpin/stdafx.cpp, 1371
 src/Enumeration_QuickSpin/stdafx.h, 1383
 src/Enumeration_QuickSpin/targetver.h, 1399
 src/EnumerationEvents/EnumerationEvents.cpp, 1429
 src/EnumerationEvents/resource.h, 1368
 src/ExceptionHandling/ExceptionHandling.cpp, 1430
 src/ExceptionHandling/resource.h, 1368
 src/ExceptionHandling/stdafx.cpp, 1372
 src/ExceptionHandling/stdafx.h, 1384
 src/ExceptionHandling/targetver.h, 1400
 src/Exposure/Exposure.cpp, 1432
 src/Exposure/resource.h, 1368
 src/Exposure/stdafx.cpp, 1372
 src/Exposure/stdafx.h, 1385
 src/Exposure/targetver.h, 1401
 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp, 1433
 src/Exposure_QuickSpin/resource.h, 1368
 src/Exposure_QuickSpin/stdafx.cpp, 1373
 src/Exposure_QuickSpin/stdafx.h, 1386
 src/Exposure_QuickSpin/targetver.h, 1402
 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp, 1435
 src/FileAccess_QuickSpin/resource.h, 1368
 src/FileAccess_QuickSpin/stdafx.cpp, 1374
 src/FileAccess_QuickSpin/stdafx.h, 1387
 src/FileAccess_QuickSpin/targetver.h, 1403
 src/GigEVisionPerformance/CpuUtil.cpp, 1438
 src/GigEVisionPerformance/CpuUtil.h, 1439
 src/GigEVisionPerformance/GigEVisionPerformance.cpp, 1440
 src/GigEVisionPerformance/GigEVisionPerformance.h, 1445
 src/GigEVisionPerformance/resource.h, 1368
 src/GigEVisionPerformance/stdafx.cpp, 1375
 src/GigEVisionPerformance/stdafx.h, 1388
 src/GigEVisionPerformance/targetver.h, 1404
 src/ImageEvents/ImageEvents.cpp, 1446
 src/ImageEvents/resource.h, 1368
 src/ImageFormatControl/ImageFormatControl.cpp, 1448
 src/ImageFormatControl/resource.h, 1368
 src/ImageFormatControl/stdafx.h, 1389
 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp, 1449
 src/ImageFormatControl_QuickSpin/resource.h, 1368
 src/ImageFormatControl_QuickSpin/stdafx.h, 1389
 src/Inference/Inference.cpp, 1450
 src/Inference/resource.h, 1368
 src/Logging/Logging.cpp, 1457
 src/Logging/resource.h, 1368
 src/LogicBlock/LogicBlock.cpp, 1458
 src/LogicBlock/resource.h, 1368
 src/LookupTable/LookupTable.cpp, 1460
 src/LookupTable/resource.h, 1368
 src/NodeMapCallback/NodeMapCallback.cpp, 1461
 src/NodeMapCallback/resource.h, 1368
 src/NodeMapInfo/NodeMapInfo.cpp, 1463
 src/NodeMapInfo/resource.h, 1368
 src/NodeMapInfo/stdafx.cpp, 1375
 src/NodeMapInfo/stdafx.h, 1390
 src/NodeMapInfo/targetver.h, 1405
 src/Polarization/Polarization.cpp, 1467
 src/Polarization/resource.h, 1368
 src/Polarization/stdafx.h, 1391
 src/SaveToAvi/resource.h, 1368
 src/SaveToAvi/SaveToAvi.cpp, 1470
 src/Sequencer/resource.h, 1368
 src/Sequencer/Sequencer.cpp, 1472
 src/Sequencer/stdafx.cpp, 1376
 src/Sequencer/stdafx.h, 1391
 src/Sequencer/targetver.h, 1406
 src/SerialRxTx/resource.h, 1368
 src/SerialRxTx/SerialRxTx.cpp, 1474
 src/SerialRxTx/stdafx.cpp, 1376
 src/SerialRxTx/stdafx.h, 1392
 src/SerialRxTx/targetver.h, 1407
 src/Trigger/resource.h, 1368
 src/Trigger/Trigger.cpp, 1477
 src/Trigger_QuickSpin/resource.h, 1368
 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp, 1480

- Standard
 - Spinnaker::GenApi, [360](#)
- STANDARD_CAST_TO_SPINNAKER
 - ExceptionHandling.cpp, [1431](#)
- STANDARD_EXCEPTION
 - ExceptionHandling.cpp, [1431](#)
- StartCpuTracing
 - CpuUtil, [190](#)
- StartPerformanceCounter
 - PerformanceCounter, [191](#)
- StartRecording
 - PortNode, [1071](#)
 - PortRecorder, [1074](#)
- StartSecondsCounter
 - SecondsCounter, [192](#)
- StartStream
 - IDataStream, [827](#)
- startTime
 - SecondsCounter, [192](#)
- StatisticsChannel
 - Spinnaker, [320](#)
- Status
 - ActionCommandResult, [401](#)
- StopCpuTracing
 - CpuUtil, [190](#)
- StopRecording
 - PortNode, [1071](#)
 - PortRecorder, [1075](#)
 - Spinnaker::GenApi, [387](#)
- StopStream
 - IDataStream, [827](#)
- StoreToBag
 - CFeatureBag, [635](#)
- Stream
 - EventHandler, [752](#)
 - Image, [851](#)
 - ImageEventHandler, [854](#)
 - Image, [912](#)
- StreamAnnounceBufferMinimum
 - TransportLayerStream, [1144](#)
- StreamAnnouncedBufferCount
 - TransportLayerStream, [1144](#)
- StreamBlockTransferSize
 - TransportLayerStream, [1145](#)
- StreamBufferAlignment
 - TransportLayerStream, [1145](#)
- StreamBufferCountManual
 - TransportLayerStream, [1145](#)
- StreamBufferCountMax
 - TransportLayerStream, [1145](#)
- StreamBufferCountMode
 - TransportLayerStream, [1145](#)
- StreamBufferCountMode_Manual
 - Spinnaker, [321](#)
- StreamBufferCountModeEnum
 - Spinnaker, [321](#)
- StreamBufferCountResult
 - TransportLayerStream, [1145](#)
- StreamBufferHandlingMode
 - TransportLayerStream, [1146](#)
- StreamBufferHandlingMode_NewestFirst
 - Spinnaker, [321](#)
- StreamBufferHandlingMode_NewestOnly
 - Spinnaker, [321](#)
- StreamBufferHandlingMode_OldestFirst
 - Spinnaker, [321](#)
- StreamBufferHandlingMode_OldestFirstOverwrite
 - Spinnaker, [321](#)
- StreamBufferHandlingModeEnum
 - Spinnaker, [321](#)
- StreamChannelId
 - GVCP_EVENT_ITEM, [788](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
- StreamChunkCountMaximum
 - TransportLayerStream, [1146](#)
- StreamCRCCheckEnable
 - TransportLayerStream, [1146](#)
- StreamDeliveredFrameCount
 - TransportLayerStream, [1146](#)
- StreamDroppedFrameCount
 - TransportLayerStream, [1146](#)
- StreamID
 - TransportLayerStream, [1146](#)
- StreamIncompleteFrameCount
 - TransportLayerStream, [1147](#)
- StreamInputBufferCount
 - TransportLayerStream, [1147](#)
- StreamIsGrabbing
 - TransportLayerStream, [1147](#)
- StreamLostFrameCount
 - TransportLayerStream, [1147](#)
- StreamMissedPacketCount
 - TransportLayerStream, [1147](#)
- StreamMode
 - TransportLayerStream, [1147](#)
- StreamMode_LWF
 - Spinnaker, [322](#)
- StreamMode_MVA
 - Spinnaker, [322](#)
- StreamMode_Socket
 - Spinnaker, [322](#)
- StreamModeEnum
 - Spinnaker, [321](#)
- StreamOutputBufferCount
 - TransportLayerStream, [1148](#)
- StreamPacketResendEnable
 - TransportLayerStream, [1148](#)
- StreamPacketResendMaxRequests
 - TransportLayerStream, [1148](#)
- StreamPacketResendReceivedPacketCount
 - TransportLayerStream, [1148](#)
- StreamPacketResendRequestCount
 - TransportLayerStream, [1148](#)
- StreamPacketResendRequestedPacketCount
 - TransportLayerStream, [1148](#)
- StreamPacketResendRequestSuccessCount

- TransportLayerStream, 1149
- StreamPacketResendTimeout
 - TransportLayerStream, 1149
- StreamReceivedFrameCount
 - TransportLayerStream, 1149
- StreamReceivedPacketCount
 - TransportLayerStream, 1149
- StreamStartedFrameCount
 - TransportLayerStream, 1149
- StreamType
 - TransportLayerStream, 1149
- StreamType_CameraLink
 - Spinnaker, 322
- StreamType_CameraLinkHS
 - Spinnaker, 322
- StreamType_CoaXPress
 - Spinnaker, 322
- StreamType_Custom
 - Spinnaker, 322
- StreamType_GigEVision
 - Spinnaker, 322
- StreamType_USB3Vision
 - Spinnaker, 322
- StreamTypeEnum
 - Spinnaker, 322
- StringGetValue
 - SpinnakerDirectShow.h, 1350
- StringList_t
 - Spinnaker::GenApi, 356
- StringNode, 1093
 - ~StringNode, 1095
 - GetMaxLength, 1095
 - GetValue, 1095
 - operator*, 1096
 - operator(), 1096
 - operator=, 1096
 - SetReference, 1096
 - SetValue, 1096
 - StringNode, 1095
- StringNode Class, 181
- StringRegNode, 1097
 - ~StringRegNode, 1099
 - SetReference, 1099
 - StringRegNode, 1098, 1099
- StringRegNode Class, 182
- StringSetValue
 - SpinnakerDirectShow.h, 1350
- StructPort Class, 183
- SubMinor
 - Version_t, 1165
- substr
 - gcstring, 784
- swap
 - gcstring, 784
- sync
 - ODevFileStreamBuf< CharType, Traits >, 1063
- Synch Class, 184
- System, 1100
 - ~System, 1101
 - GetCameras, 1102
 - GetInstance, 1102
 - GetInterfaces, 1103
 - GetLibraryVersion, 1103
 - GetLoggingEventPriorityLevel, 1104
 - GetTLNodeMap, 1104
 - IsInUse, 1104
 - RegisterEventHandler, 1105
 - RegisterLoggingEventHandler, 1105
 - ReleaseInstance, 1105
 - SendActionCommand, 1106
 - SetLoggingEventPriorityLevel, 1106
 - System, 1102
 - TransportLayerSystem, 1152
 - UnregisterAllLoggingEventHandlers, 1107
 - UnregisterEventHandler, 1107
 - UnregisterLoggingEventHandler, 1108
 - UpdateCameras, 1108
 - UpdateInterfaceList, 1108
- System Class, 83
- System.h
 - FLIR_SPINNAKER_VERSION_BUILD, 1358
 - FLIR_SPINNAKER_VERSION_MAJOR, 1358
 - FLIR_SPINNAKER_VERSION_MINOR, 1359
 - FLIR_SPINNAKER_VERSION_TYPE, 1359
- SystemEventHandler, 1109
 - ~SystemEventHandler, 1111
 - OnInterfaceArrival, 1111
 - OnInterfaceRemoval, 1111
 - operator=, 1112
 - SystemEventHandler, 1110
- SystemEventHandler Class, 84
- SystemEventHandlerImpl, 1112
 - ~SystemEventHandlerImpl, 1113
 - LockEventHandlerMutex, 1114
 - OnInterfaceArrival, 1114
 - OnInterfaceRemoval, 1114
 - RegisterAllInterfaceEvents, 1114
 - RegisterInterfaceEventToSystem, 1115
 - SystemEventHandlerImpl, 1113
 - UnlockEventHandlerMutex, 1115
 - UnregisterAllInterfaceEvents, 1115
 - UnregisterInterfaceEventFromSystem, 1115
- SystemImpl
 - InterfaceList, 1004
 - LoggingEventData, 1028
- SystemPtr, 1116
 - ~SystemPtr, 1117
 - SystemPtr, 1117
- SystemPtr Class, 85
- SystemPtrInternal
 - ISystem, 1016
 - TransportLayerSystem, 1152
- TAG_EVENT_CMD
 - Spinnaker::GenApi, 363
- TAG_EVENTDATA_CMD
 - Spinnaker::GenApi, 363

- Test0001
 - Camera, [553](#)
- TestDuration
 - GigEVisionPerformance.cpp, [1445](#)
- TestEventGenerate
 - Camera, [553](#)
- TestPattern
 - Camera, [553](#)
- TestPattern_Increment
 - Spinnaker, [322](#)
- TestPattern_Off
 - Spinnaker, [322](#)
- TestPattern_SensorTestPattern
 - Spinnaker, [322](#)
- TestPatternEnums
 - Spinnaker, [322](#)
- TestPatternGeneratorSelector
 - Camera, [553](#)
- TestPatternGeneratorSelector_PipelineStart
 - Spinnaker, [323](#)
- TestPatternGeneratorSelector_Sensor
 - Spinnaker, [323](#)
- TestPatternGeneratorSelectorEnums
 - Spinnaker, [323](#)
- TestPendingAck
 - Camera, [554](#)
- ThrowBadAlloc
 - Spinnaker::GenICam, [399](#)
- TIFFCompressionMethod
 - Spinnaker, [323](#)
- TIFFOption, [1118](#)
 - compression, [1118](#)
 - reserved, [1119](#)
- TIFFOption, [1118](#)
- timeDiff
 - SecondsCounter, [192](#)
- TimerDelay
 - Camera, [554](#)
- TimerDuration
 - Camera, [554](#)
- TimerReset
 - Camera, [554](#)
- TimerSelector
 - Camera, [554](#)
- TimerSelector_Timer0
 - Spinnaker, [325](#)
- TimerSelector_Timer1
 - Spinnaker, [325](#)
- TimerSelector_Timer2
 - Spinnaker, [325](#)
- TimerSelectorEnums
 - Spinnaker, [323](#)
- TimerStatus
 - Camera, [555](#)
- TimerStatus_TimerActive
 - Spinnaker, [325](#)
- TimerStatus_TimerCompleted
 - Spinnaker, [325](#)
- TimerStatus_TimerIdle
 - Spinnaker, [325](#)
- TimerStatus_TimerTriggerWait
 - Spinnaker, [325](#)
- TimerStatusEnums
 - Spinnaker, [325](#)
- TimerTriggerActivation
 - Camera, [555](#)
- TimerTriggerActivation_AnyEdge
 - Spinnaker, [325](#)
- TimerTriggerActivation_FallingEdge
 - Spinnaker, [325](#)
- TimerTriggerActivation_LevelHigh
 - Spinnaker, [325](#)
- TimerTriggerActivation_LevelLow
 - Spinnaker, [325](#)
- TimerTriggerActivation_RisingEdge
 - Spinnaker, [325](#)
- TimerTriggerActivationEnums
 - Spinnaker, [325](#)
- TimerTriggerSource
 - Camera, [555](#)
- TimerTriggerSource_AcquisitionEnd
 - Spinnaker, [326](#)
- TimerTriggerSource_AcquisitionStart
 - Spinnaker, [326](#)
- TimerTriggerSource_AcquisitionTrigger
 - Spinnaker, [326](#)
- TimerTriggerSource_Action0
 - Spinnaker, [326](#)
- TimerTriggerSource_Action1
 - Spinnaker, [327](#)
- TimerTriggerSource_Action2
 - Spinnaker, [327](#)
- TimerTriggerSource_Counter0End
 - Spinnaker, [326](#)
- TimerTriggerSource_Counter0Start
 - Spinnaker, [326](#)
- TimerTriggerSource_Counter1End
 - Spinnaker, [326](#)
- TimerTriggerSource_Counter1Start
 - Spinnaker, [326](#)
- TimerTriggerSource_Counter2End
 - Spinnaker, [326](#)
- TimerTriggerSource_Counter2Start
 - Spinnaker, [326](#)
- TimerTriggerSource_Encoder0
 - Spinnaker, [326](#)
- TimerTriggerSource_Encoder1
 - Spinnaker, [326](#)
- TimerTriggerSource_Encoder2
 - Spinnaker, [326](#)
- TimerTriggerSource_ExposureEnd
 - Spinnaker, [326](#)
- TimerTriggerSource_ExposureStart
 - Spinnaker, [326](#)
- TimerTriggerSource_FrameBurstEnd
 - Spinnaker, [326](#)

- TimerTriggerSource_FrameBurstStart
 - Spinnaker, [326](#)
- TimerTriggerSource_FrameEnd
 - Spinnaker, [326](#)
- TimerTriggerSource_FrameStart
 - Spinnaker, [326](#)
- TimerTriggerSource_FrameTrigger
 - Spinnaker, [326](#)
- TimerTriggerSource_Line0
 - Spinnaker, [326](#)
- TimerTriggerSource_Line1
 - Spinnaker, [326](#)
- TimerTriggerSource_Line2
 - Spinnaker, [326](#)
- TimerTriggerSource_LineEnd
 - Spinnaker, [326](#)
- TimerTriggerSource_LineStart
 - Spinnaker, [326](#)
- TimerTriggerSource_LineTrigger
 - Spinnaker, [326](#)
- TimerTriggerSource_LinkTrigger0
 - Spinnaker, [327](#)
- TimerTriggerSource_LinkTrigger1
 - Spinnaker, [327](#)
- TimerTriggerSource_LinkTrigger2
 - Spinnaker, [327](#)
- TimerTriggerSource_Off
 - Spinnaker, [326](#)
- TimerTriggerSource_SoftwareSignal0
 - Spinnaker, [326](#)
- TimerTriggerSource_SoftwareSignal1
 - Spinnaker, [326](#)
- TimerTriggerSource_SoftwareSignal2
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer0End
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer0Start
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer1End
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer1Start
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer2End
 - Spinnaker, [326](#)
- TimerTriggerSource_Timer2Start
 - Spinnaker, [326](#)
- TimerTriggerSource_UserOutput0
 - Spinnaker, [326](#)
- TimerTriggerSource_UserOutput1
 - Spinnaker, [326](#)
- TimerTriggerSource_UserOutput2
 - Spinnaker, [326](#)
- TimerTriggerSourceEnums
 - Spinnaker, [325](#)
- TimerValue
 - Camera, [555](#)
- Timestamp
 - Camera, [555](#)
- U3V_EVENT_DATA, [1159](#)
- TimestampHigh
 - GVCP_EVENT_ITEM, [788](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
- TimestampLatch
 - Camera, [555](#)
- TimestampLatchValue
 - Camera, [556](#)
- TimestampLow
 - GVCP_EVENT_ITEM, [788](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [790](#)
- TimestampReset
 - Camera, [556](#)
- TLDevice
 - ICameraBase, [806](#)
- TLDisplayName
 - TransportLayerSystem, [1155](#)
- TLFileName
 - TransportLayerSystem, [1156](#)
- TLID
 - TransportLayerSystem, [1156](#)
- TLInterface
 - Interface, [873](#)
- TLModelName
 - TransportLayerSystem, [1156](#)
- TLParamsLocked
 - Camera, [556](#)
- TLPath
 - TransportLayerSystem, [1156](#)
- TLPayloadType
 - Spinnaker, [327](#)
- TLPixelFormatNamespace
 - Spinnaker, [327](#)
- TLStream
 - ICameraBase, [807](#)
- TLSystem
 - ISystem, [1016](#)
- TLType
 - TransportLayerSystem, [1156](#)
- TLType_CameraLink
 - Spinnaker, [328](#)
- TLType_CameraLinkHS
 - Spinnaker, [328](#)
- TLType_CoaXPRESS
 - Spinnaker, [328](#)
- TLType_Custom
 - Spinnaker, [328](#)
- TLType_GigEVision
 - Spinnaker, [328](#)
- TLType_Mixed
 - Spinnaker, [328](#)
- TLType_USB3Vision
 - Spinnaker, [328](#)
- TLTypeEnum
 - Spinnaker, [328](#)
- TLVendorName
 - TransportLayerSystem, [1156](#)
- TLVersion

- TransportLayerSystem, [1157](#)
- Tokenize
 - Spinnaker::GenlCam, [399](#)
- topLeftXCoord
 - InferenceBoxRect, [973](#)
 - InferenceBoxRotatedRect, [974](#)
- topLeftYCoord
 - InferenceBoxRect, [973](#)
 - InferenceBoxRotatedRect, [974](#)
- ToString
 - CNodeMapFactory, [674](#)
 - CSelectorSet, [707](#)
 - EAccessModeClass, [729](#)
 - ECachingModeClass, [730](#)
 - EDisplayNotationClass, [731](#)
 - EEndianessClass, [732](#)
 - EGenApiSchemaVersionClass, [733](#)
 - EInputDirectionClass, [734](#)
 - ENameSpaceClass, [735](#)
 - ERepresentationClass, [745](#)
 - ESignClass, [746](#)
 - ESlopeClass, [747](#)
 - EStandardNameSpaceClass, [748](#)
 - EVisibilityClass, [753](#)
 - EYesNoClass, [759](#)
 - Spinnaker::GenApi, [387](#)
 - ValueNode, [1163](#)
- ToXml
 - CNodeMapFactory, [674](#)
- TransferAbort
 - Camera, [556](#)
- TransferBlockCount
 - Camera, [556](#)
- TransferBurstCount
 - Camera, [556](#)
- TransferComponentSelector
 - Camera, [557](#)
- TransferComponentSelector_All
 - Spinnaker, [329](#)
- TransferComponentSelector_Blue
 - Spinnaker, [328](#)
- TransferComponentSelector_Green
 - Spinnaker, [328](#)
- TransferComponentSelector_Red
 - Spinnaker, [328](#)
- TransferComponentSelectorEnums
 - Spinnaker, [328](#)
- TransferControlMode
 - Camera, [557](#)
- TransferControlMode_Automatic
 - Spinnaker, [329](#)
- TransferControlMode_Basic
 - Spinnaker, [329](#)
- TransferControlMode_UserControlled
 - Spinnaker, [329](#)
- TransferControlModeEnums
 - Spinnaker, [329](#)
- TransferOperationMode
 - Camera, [557](#)
- TransferOperationMode_Continuous
 - Spinnaker, [329](#)
- TransferOperationMode_MultiBlock
 - Spinnaker, [329](#)
- TransferOperationModeEnums
 - Spinnaker, [329](#)
- TransferPause
 - Camera, [557](#)
- TransferQueueCurrentBlockCount
 - Camera, [557](#)
- TransferQueueMaxBlockCount
 - Camera, [557](#)
- TransferQueueMode
 - Camera, [558](#)
- TransferQueueMode_FirstInFirstOut
 - Spinnaker, [330](#)
- TransferQueueModeEnums
 - Spinnaker, [329](#)
- TransferQueueOverflowCount
 - Camera, [558](#)
- TransferResume
 - Camera, [558](#)
- TransferSelector
 - Camera, [558](#)
- TransferSelector_All
 - Spinnaker, [330](#)
- TransferSelector_Stream0
 - Spinnaker, [330](#)
- TransferSelector_Stream1
 - Spinnaker, [330](#)
- TransferSelector_Stream2
 - Spinnaker, [330](#)
- TransferSelectorEnums
 - Spinnaker, [330](#)
- TransferStart
 - Camera, [558](#)
- TransferStatus
 - Camera, [558](#)
- TransferStatusSelector
 - Camera, [559](#)
- TransferStatusSelector_Paused
 - Spinnaker, [330](#)
- TransferStatusSelector_QueueOverflow
 - Spinnaker, [330](#)
- TransferStatusSelector_Stopped
 - Spinnaker, [330](#)
- TransferStatusSelector_Stopping
 - Spinnaker, [330](#)
- TransferStatusSelector_Streaming
 - Spinnaker, [330](#)
- TransferStatusSelectorEnums
 - Spinnaker, [330](#)
- TransferStop
 - Camera, [559](#)
- TransferStreamChannel
 - Camera, [559](#)
- TransferTriggerActivation

- Camera, [559](#)
- TransferTriggerActivation_AnyEdge
 - Spinnaker, [331](#)
- TransferTriggerActivation_FallingEdge
 - Spinnaker, [331](#)
- TransferTriggerActivation_LevelHigh
 - Spinnaker, [331](#)
- TransferTriggerActivation_LevelLow
 - Spinnaker, [331](#)
- TransferTriggerActivation_RisingEdge
 - Spinnaker, [331](#)
- TransferTriggerActivationEnums
 - Spinnaker, [330](#)
- TransferTriggerMode
 - Camera, [559](#)
- TransferTriggerMode_Off
 - Spinnaker, [331](#)
- TransferTriggerMode_On
 - Spinnaker, [331](#)
- TransferTriggerModeEnums
 - Spinnaker, [331](#)
- TransferTriggerSelector
 - Camera, [559](#)
- TransferTriggerSelector_TransferAbort
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferActive
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferBurstStart
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferBurstStop
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferPause
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferResume
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferStart
 - Spinnaker, [331](#)
- TransferTriggerSelector_TransferStop
 - Spinnaker, [331](#)
- TransferTriggerSelectorEnums
 - Spinnaker, [331](#)
- TransferTriggerSource
 - Camera, [560](#)
- TransferTriggerSource_Action0
 - Spinnaker, [332](#)
- TransferTriggerSource_Action1
 - Spinnaker, [332](#)
- TransferTriggerSource_Action2
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter0End
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter0Start
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter1End
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter1Start
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter2End
 - Spinnaker, [332](#)
- TransferTriggerSource_Counter2Start
 - Spinnaker, [332](#)
- TransferTriggerSource_Line0
 - Spinnaker, [332](#)
- TransferTriggerSource_Line1
 - Spinnaker, [332](#)
- TransferTriggerSource_Line2
 - Spinnaker, [332](#)
- TransferTriggerSource_SoftwareSignal0
 - Spinnaker, [332](#)
- TransferTriggerSource_SoftwareSignal1
 - Spinnaker, [332](#)
- TransferTriggerSource_SoftwareSignal2
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer0End
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer0Start
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer1End
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer1Start
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer2End
 - Spinnaker, [332](#)
- TransferTriggerSource_Timer2Start
 - Spinnaker, [332](#)
- TransferTriggerSourceEnums
 - Spinnaker, [332](#)
- TransportLayerDefs Class, [87](#)
- TransportLayerDevice, [1119](#)
 - ~TransportLayerDevice, [1121](#)
 - CameraBase, [1122](#)
 - CameraInternal, [1122](#)
 - DeviceAccessStatus, [1122](#)
 - DeviceBootloaderVersion, [1122](#)
 - DeviceCurrentSpeed, [1123](#)
 - DeviceDisplayName, [1123](#)
 - DeviceDriverVersion, [1123](#)
 - DeviceEndiannessMechanism, [1123](#)
 - DeviceID, [1123](#)
 - DeviceInstanceId, [1123](#)
 - DevicesUpdater, [1124](#)
 - DeviceLinkSpeed, [1124](#)
 - DeviceLocation, [1124](#)
 - DeviceModelName, [1124](#)
 - DeviceMulticastMonitorMode, [1124](#)
 - DevicePortId, [1124](#)
 - DeviceReset, [1125](#)
 - DeviceSerialNumber, [1125](#)
 - DeviceType, [1125](#)
 - DeviceU3VProtocol, [1125](#)
 - DeviceUserID, [1125](#)
 - DeviceVendorName, [1125](#)
 - DeviceVersion, [1126](#)
 - GenICamXMLLocation, [1126](#)
 - GenICamXMLPath, [1126](#)
 - GevCCP, [1126](#)

- GevDeviceAutoForceIP, 1126
- GevDeviceDiscoverMaximumPacketSize, 1126
- GevDeviceForceGateway, 1127
- GevDeviceForceIP, 1127
- GevDeviceForceIPAddress, 1127
- GevDeviceForceSubnetMask, 1127
- GevDeviceGateway, 1127
- GevDeviceIPAddress, 1127
- GevDevicesWrongSubnet, 1128
- GevDeviceMACAddress, 1128
- GevDeviceMaximumPacketSize, 1128
- GevDeviceMaximumRetryCount, 1128
- GevDeviceModelsBigEndian, 1128
- GevDevicePort, 1128
- GevDeviceReadAndWriteTimeout, 1129
- GevDeviceSubnetMask, 1129
- GevVersionMajor, 1129
- GevVersionMinor, 1129
- GUXMLLocation, 1129
- GUXMLPath, 1129
- ICameraBase, 1122
- TransportLayerDevice, 1121, 1122
- TransportLayerDevice Class, 88
- TransportLayerInterface, 1130
 - ~TransportLayerInterface, 1132
 - ActionCommand, 1133
 - DeviceAccessStatus, 1133
 - DeviceCount, 1134
 - DeviceID, 1134
 - DeviceModelName, 1134
 - DeviceSelector, 1134
 - DeviceSerialNumber, 1134
 - DeviceUnlock, 1134
 - DeviceUpdateList, 1135
 - DeviceVendorName, 1135
 - FilterDriverStatus, 1135
 - GevActionDeviceKey, 1135
 - GevActionGroupKey, 1135
 - GevActionGroupMask, 1135
 - GevActionTime, 1136
 - GevDeviceAutoForceIP, 1136
 - GevDeviceForceGateway, 1136
 - GevDeviceForceIP, 1136
 - GevDeviceForceIPAddress, 1136
 - GevDeviceForceSubnetMask, 1136
 - GevDeviceGateway, 1137
 - GevDeviceIPAddress, 1137
 - GevDeviceMACAddress, 1137
 - GevDeviceSubnetMask, 1137
 - GevInterfaceGateway, 1137
 - GevInterfaceGatewaySelector, 1137
 - GevInterfaceMACAddress, 1138
 - GevInterfaceMTU, 1138
 - GevInterfaceReceiveLinkSpeed, 1138
 - GevInterfaceSubnetIPAddress, 1138
 - GevInterfaceSubnetMask, 1138
 - GevInterfaceSubnetSelector, 1138
 - GevInterfaceTransmitLinkSpeed, 1139
 - HostAdapterDriverVersion, 1139
 - HostAdapterName, 1139
 - HostAdapterVendor, 1139
 - IInterface, 1133
 - IncompatibleDeviceCount, 1139
 - IncompatibleDeviceID, 1139
 - IncompatibleDeviceModelName, 1140
 - IncompatibleDeviceSelector, 1140
 - IncompatibleDeviceVendorName, 1140
 - IncompatibleGevDeviceIPAddress, 1140
 - IncompatibleGevDeviceMACAddress, 1140
 - IncompatibleGevDeviceSubnetMask, 1140
 - Interface, 1133
 - InterfaceDisplayName, 1141
 - InterfaceID, 1141
 - InterfaceInternal, 1133
 - InterfaceType, 1141
 - POEStatus, 1141
 - TransportLayerInterface, 1132, 1133
- TransportLayerInterface Class, 89
- TransportLayerStream, 1141
 - ~TransportLayerStream, 1143
 - CameraBase, 1144
 - CameraInternal, 1144
 - ICameraBase, 1144
 - StreamAnnounceBufferMinimum, 1144
 - StreamAnnouncedBufferCount, 1144
 - StreamBlockTransferSize, 1145
 - StreamBufferAlignment, 1145
 - StreamBufferCountManual, 1145
 - StreamBufferCountMax, 1145
 - StreamBufferCountMode, 1145
 - StreamBufferCountResult, 1145
 - StreamBufferHandlingMode, 1146
 - StreamChunkCountMaximum, 1146
 - StreamCRCCheckEnable, 1146
 - StreamDeliveredFrameCount, 1146
 - StreamDroppedFrameCount, 1146
 - StreamID, 1146
 - StreamIncompleteFrameCount, 1147
 - StreamInputBufferCount, 1147
 - StreamIsGrabbing, 1147
 - StreamLostFrameCount, 1147
 - StreamMissedPacketCount, 1147
 - StreamMode, 1147
 - StreamOutputBufferCount, 1148
 - StreamPacketResendEnable, 1148
 - StreamPacketResendMaxRequests, 1148
 - StreamPacketResendReceivedPacketCount, 1148
 - StreamPacketResendRequestCount, 1148
 - StreamPacketResendRequestedPacketCount, 1148
 - StreamPacketResendRequestSuccessCount, 1149
 - StreamPacketResendTimeout, 1149
 - StreamReceivedFrameCount, 1149
 - StreamReceivedPacketCount, 1149
 - StreamStartedFrameCount, 1149

- StreamType, 1149
- TransportLayerStream, 1143, 1144
- TransportLayerStream Class, 90
- TransportLayerStreamInfo
 - IDataStream, 827
- TransportLayerSystem, 1150
 - ~TransportLayerSystem, 1152
 - EnumerateGen2Cameras, 1152
 - EnumerateGEVInterfaces, 1153
 - EnumerateUSBInterfaces, 1153
 - GenTLSFNCVersionMajor, 1153
 - GenTLSFNCVersionMinor, 1153
 - GenTLSFNCVersionSubMinor, 1153
 - GenTLVersionMajor, 1153
 - GenTLVersionMinor, 1154
 - GevInterfaceDefaultGateway, 1154
 - GevInterfaceDefaultIPAddress, 1154
 - GevInterfaceDefaultSubnetMask, 1154
 - GevInterfaceMACAddress, 1154
 - GevVersionMajor, 1154
 - GevVersionMinor, 1155
 - InterfaceDisplayName, 1155
 - InterfaceID, 1155
 - InterfaceSelector, 1155
 - InterfaceUpdateList, 1155
 - ISystem, 1152
 - System, 1152
 - SystemPtrInternal, 1152
 - TLDisplayName, 1155
 - TLFileName, 1156
 - TLID, 1156
 - TLModelName, 1156
 - TLPath, 1156
 - TLType, 1156
 - TLVendorName, 1156
 - TLVersion, 1157
 - TransportLayerSystem, 1151, 1152
- TransportLayerSystem Class, 91
- Trigger.cpp
 - AcquireImages, 1479
 - chosenTrigger, 1480
 - ConfigureTrigger, 1479
 - GrabNextImageByTrigger, 1479
 - HARDWARE, 1479
 - main, 1479
 - PrintDeviceInfo, 1479
 - ResetTrigger, 1479
 - RunSingleCamera, 1480
 - SOFTWARE, 1479
 - triggerType, 1478
- Trigger_QuickSpin.cpp
 - AcquireImages, 1481
 - chosenTrigger, 1482
 - ConfigureTrigger, 1481
 - GrabNextImageByTrigger, 1481
 - HARDWARE, 1481
 - main, 1481
 - PrintDeviceInfo, 1482
 - ResetTrigger, 1482
 - RunSingleCamera, 1482
 - SOFTWARE, 1481
 - triggerType, 1481
 - TriggerActivation
 - Camera, 560
 - TriggerActivation_AnyEdge
 - Spinnaker, 333
 - TriggerActivation_FallingEdge
 - Spinnaker, 333
 - TriggerActivation_LevelHigh
 - Spinnaker, 333
 - TriggerActivation_LevelLow
 - Spinnaker, 333
 - TriggerActivation_RisingEdge
 - Spinnaker, 333
 - TriggerActivationEnums
 - Spinnaker, 333
 - TriggerDelay
 - Camera, 560
 - TriggerDivider
 - Camera, 560
 - TriggerEventTest
 - Camera, 560
 - TriggerMode
 - Camera, 561
 - TriggerMode_Off
 - Spinnaker, 333
 - TriggerMode_On
 - Spinnaker, 333
 - TriggerModeEnums
 - Spinnaker, 333
 - TriggerMultiplier
 - Camera, 561
 - TriggerOverlap
 - Camera, 561
 - TriggerOverlap_Off
 - Spinnaker, 333
 - TriggerOverlap_PreviousFrame
 - Spinnaker, 333
 - TriggerOverlap_ReadOut
 - Spinnaker, 333
 - TriggerOverlapEnums
 - Spinnaker, 333
 - TriggerSelector
 - Camera, 561
 - TriggerSelector_AcquisitionStart
 - Spinnaker, 334
 - TriggerSelector_FrameBurstStart
 - Spinnaker, 334
 - TriggerSelector_FrameStart
 - Spinnaker, 334
 - TriggerSelectorEnums
 - Spinnaker, 333
 - TriggerSoftware
 - Camera, 561
 - TriggerSource
 - Camera, 562

- TriggerSource_Action0
 - Spinnaker, [334](#)
- TriggerSource_Counter0End
 - Spinnaker, [334](#)
- TriggerSource_Counter0Start
 - Spinnaker, [334](#)
- TriggerSource_Counter1End
 - Spinnaker, [334](#)
- TriggerSource_Counter1Start
 - Spinnaker, [334](#)
- TriggerSource_Line0
 - Spinnaker, [334](#)
- TriggerSource_Line1
 - Spinnaker, [334](#)
- TriggerSource_Line2
 - Spinnaker, [334](#)
- TriggerSource_Line3
 - Spinnaker, [334](#)
- TriggerSource_LogicBlock0
 - Spinnaker, [334](#)
- TriggerSource_LogicBlock1
 - Spinnaker, [334](#)
- TriggerSource_Software
 - Spinnaker, [334](#)
- TriggerSource_UserOutput0
 - Spinnaker, [334](#)
- TriggerSource_UserOutput1
 - Spinnaker, [334](#)
- TriggerSource_UserOutput2
 - Spinnaker, [334](#)
- TriggerSource_UserOutput3
 - Spinnaker, [334](#)
- TriggerSourceEnums
 - Spinnaker, [334](#)
- triggerType
 - Trigger.cpp, [1478](#)
 - Trigger_QuickSpin.cpp, [1481](#)
- TryLock
 - CGlobalLock, [640](#)
 - CLock, [658](#), [660](#)
- TWO_SECOND_DELAY
 - SerialRxTx.cpp, [1475](#)
- Type
 - CCMSettings, [611](#)
- type
 - LibraryVersion, [1022](#)
- Types Enums, [186](#)
- Types.h
 - _UndefinedRepresentation, [1327](#)
 - interface, [1327](#)
- U3V_CHUNK_TRAILER, [1157](#)
 - ChunkID, [1157](#)
 - ChunkLength, [1157](#)
- U3V_COMMAND_HEADER, [1158](#)
 - CommandId, [1158](#)
 - Flags, [1158](#)
 - Length, [1158](#)
 - Prefix, [1158](#)
 - ReqId, [1159](#)
- U3V_EVENT_DATA, [1159](#)
 - EventId, [1159](#)
 - Reserved, [1159](#)
 - Timestamp, [1159](#)
- U3V_EVENT_MESSAGE, [1160](#)
 - CommandHeader, [1160](#)
 - EventData, [1160](#)
- U3V_EVENT_PREFIX
 - Spinnaker::GenApi, [394](#)
- UNCOMPRESSED
 - SaveToAvi.cpp, [1470](#)
- underflow
 - IDevFileStreamBuf< CharType, Traits >, [832](#)
- UNKNOWN_PIXELFORMAT
 - Spinnaker, [302](#)
- Unlock
 - CGlobalLock, [641](#)
 - CLock, [658](#), [660](#)
- UnlockEarly
 - CGlobalLockUnlocker, [642](#)
- UnlockEventHandlerMutex
 - SystemEventHandlerImpl, [1115](#)
- UnregisterAllInterfaceEvents
 - SystemEventHandlerImpl, [1115](#)
- UnregisterAllLoggingEventHandlers
 - ISystem, [1015](#)
 - System, [1107](#)
- UnregisterEventHandler
 - CameraBase, [580](#)
 - ICameraBase, [805](#)
 - IInterface, [872](#)
 - Interface, [988](#)
 - ISystem, [1015](#)
 - System, [1107](#)
- UnregisterImageEventHandler
 - IDataStream, [827](#)
- UnregisterInterfaceEventFromSystem
 - SystemEventHandlerImpl, [1115](#)
- UnregisterLoggingEventHandler
 - ISystem, [1015](#)
 - System, [1108](#)
- Unsigned
 - Spinnaker::GenApi, [360](#)
- UpdateBuffer
 - CChunkAdapter, [595](#)
 - CChunkPort, [609](#)
- UpdateCameras
 - IInterface, [872](#)
 - Interface, [988](#)
 - ISystem, [1015](#)
 - System, [1108](#)
- UpdateFirmware
 - SpinUpdate.h, [1354](#)
- UpdateFirmwareConsole
 - SpinUpdate.h, [1354](#)
- UpdateFirmwareGUI
 - SpinUpdate.h, [1355](#)

- UpdateInterfaceList
 - ISystem, 1015
 - System, 1108
- UpdatorMessageCallback
 - SpinUpdate.h, 1355
- UpdatorProgressCallback
 - SpinUpdate.h, 1355
- UploadFileToCamera
 - Inference.cpp, 1455
- UploadImage
 - FileAccess_QuickSpin.cpp, 1437
- UrlDecode
 - Spinnaker::GenICam, 400
- UrlEncode
 - Spinnaker::GenICam, 400
- USB
 - Spinnaker::GenApi, 361
- USE_TEMP_CACHE_FILE
 - GCUtilities.h, 1274
- UseDuration
 - GigEVisionPerformance.cpp, 1445
- UseMaxFramerate
 - GigEVisionPerformance.cpp, 1445
- UserOutputSelector
 - Camera, 562
- UserOutputSelector_UserOutput0
 - Spinnaker, 335
- UserOutputSelector_UserOutput1
 - Spinnaker, 335
- UserOutputSelector_UserOutput2
 - Spinnaker, 335
- UserOutputSelector_UserOutput3
 - Spinnaker, 335
- UserOutputSelectorEnums
 - Spinnaker, 334
- UserOutputValue
 - Camera, 562
- UserOutputValueAll
 - Camera, 562
- UserOutputValueAllMask
 - Camera, 562
- UserSetDefault
 - Camera, 563
- UserSetDefault_Default
 - Spinnaker, 335
- UserSetDefault_UserSet0
 - Spinnaker, 335
- UserSetDefault_UserSet1
 - Spinnaker, 335
- UserSetDefaultEnums
 - Spinnaker, 335
- UserSetFeatureEnable
 - Camera, 563
- UserSetFramerate
 - GigEVisionPerformance.cpp, 1445
- UserSetLoad
 - Camera, 563
- UserSetSave
 - Camera, 563
- UserSetSelector
 - Camera, 564
- UserSetSelector_Default
 - Spinnaker, 335
- UserSetSelector_UserSet0
 - Spinnaker, 335
- UserSetSelector_UserSet1
 - Spinnaker, 335
- UserSetSelectorEnums
 - Spinnaker, 335
- v1_0
 - Spinnaker::GenApi, 358
- v1_1
 - Spinnaker::GenApi, 358
- V3_3Enable
 - Camera, 564
- VALUE
 - NodeMapInfo.cpp, 1464
- ValueNode, 1161
 - ~ValueNode, 1162
 - FromString, 1163
 - GetNode, 1163
 - IsValueCacheValid, 1163
 - SetReference, 1163
 - ToString, 1163
 - ValueNode, 1162
- ValueNode Class, 187
- Varying
 - Spinnaker::GenApi, 361
- Verify
 - Spinnaker::GenApi, 394
- Version_t, 1164
 - Major, 1164
 - Minor, 1164
 - SubMinor, 1165
- videoType
 - SaveToAvi.cpp, 1470
- WaitForImages
 - ImageEvents.cpp, 1447
- WaitOnImageEvent
 - IDataStream, 828
- what
 - Exception, 758
- WhiteClip
 - Camera, 564
- WhiteClipSelector
 - Camera, 564
- WhiteClipSelector_All
 - Spinnaker, 336
- WhiteClipSelector_Blue
 - Spinnaker, 336
- WhiteClipSelector_Green
 - Spinnaker, 336
- WhiteClipSelector_Red
 - Spinnaker, 336
- WhiteClipSelector_Tap1

- Spinnaker, [336](#)
- WhiteClipSelector_Tap2
 - Spinnaker, [336](#)
- WhiteClipSelector_U
 - Spinnaker, [336](#)
- WhiteClipSelector_V
 - Spinnaker, [336](#)
- WhiteClipSelector_Y
 - Spinnaker, [336](#)
- WhiteClipSelectorEnums
 - Spinnaker, [335](#)
- Width
 - Camera, [564](#)
- width
 - AVIOption, [405](#)
 - H264Option, [797](#)
 - MJPEGOption, [1037](#)
- WidthMax
 - Camera, [565](#)
- WO
 - Spinnaker::GenApi, [356](#)
- Write
 - CChunkPort, [609](#)
 - CEventPort, [632](#)
 - CPortImpl, [697](#)
 - CPortWriteList, [700](#)
 - CRegisterPortImpl, [704](#)
 - CTestPortStruct< CDataStruct >, [710](#)
 - PortNode, [1071](#)
 - PortRecorder, [1075](#)
 - PortReplay, [1079](#)
 - Spinnaker::GenApi, [388](#)
- write
 - FileProtocolAdapter, [764](#)
- WriteAround
 - Spinnaker::GenApi, [357](#)
- WritePort
 - CameraBase, [581](#)
 - ICameraBase, [806](#)
- WriteRegister
 - CRegisterPortImpl, [704](#)
- WriteThrough
 - Spinnaker::GenApi, [357](#)
- xsputn
 - ODevFileStreamBuf< CharType, Traits >, [1063](#)
- xvAll
 - Spinnaker::GenApi, [362](#)
- xvCycles
 - Spinnaker::GenApi, [362](#)
- xvDefault
 - Spinnaker::GenApi, [362](#)
- xvLoad
 - Spinnaker::GenApi, [362](#)
- xvSFNC
 - Spinnaker::GenApi, [362](#)
- Yes
 - Spinnaker::GenApi, [362](#)
- z_numTriggers
 - BufferHandling.cpp, [1417](#)