

**Spinnaker C++**

2.0.0.0

Generated by Doxygen 1.8.17



---

<b>1 Getting Started</b>	<b>1</b>
<b>2 Programmer's Guide</b>	<b>3</b>
<b>3 Benefits of Spinnaker</b>	<b>5</b>
<b>4 FlyCapture2 Feature Comparison with Spinnaker</b>	<b>7</b>
<b>5 Working with GenICam GenTL Devices</b>	<b>9</b>
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
<b>6 Software Licensing Information</b>	<b>11</b>
<b>7 Module Index</b>	<b>13</b>
7.1 Modules . . . . .	13
<b>8 Namespace Index</b>	<b>17</b>
8.1 Namespace List . . . . .	17
<b>9 Hierarchical Index</b>	<b>19</b>
9.1 Class Hierarchy . . . . .	19
<b>10 Class Index</b>	<b>25</b>
10.1 Class List . . . . .	25
<b>11 File Index</b>	<b>33</b>
11.1 File List . . . . .	33
<b>12 Module Documentation</b>	<b>39</b>
12.1 Spinnaker Classes . . . . .	39
12.2 AVI Recorder Class . . . . .	40
12.3 BasePtr Class . . . . .	41
12.4 Camera Class . . . . .	42
12.5 Camera Base Class . . . . .	43
12.6 CameraDefs Class . . . . .	44
12.7 Camera List Class . . . . .	45
12.8 CameraPtr Class . . . . .	46
12.9 ChunkData Class . . . . .	47
12.10 Chunk Data Inference Class . . . . .	48
12.11 Spinnaker EventHandler Classes . . . . .	49
12.12 DeviceArrivalEventHandler Class . . . . .	50
12.13 DeviceEventHandler Class . . . . .	51

---

---

12.14 DeviceRemovalEventHandler Class . . . . .	52
12.15 EventHandler Class . . . . .	53
12.16 Exception Class . . . . .	54
12.17 Image Class . . . . .	55
12.18 ImageEventHandler Class . . . . .	56
12.19 ImagePtr Class . . . . .	57
12.20 ImageStatistics Class . . . . .	58
12.21 Image Utility Class . . . . .	59
12.22 Image Utility Heatmap Class . . . . .	60
12.23 Image Utility Polarization Class . . . . .	61
12.24 Interface Class . . . . .	62
12.25 InterfaceArrivalEventHandler Class . . . . .	63
12.26 InterfaceEventHandler Class . . . . .	64
12.27 InterfaceList Class . . . . .	65
12.28 InterfacePtr Class . . . . .	66
12.29 InterfaceRemovalEventHandler Class . . . . .	67
12.30 Logging EventHandler Class . . . . .	68
12.31 LoggingEventDataPtr Class . . . . .	69
12.32 LoggingEventHandler Class . . . . .	70
12.33 Spinnaker Headers . . . . .	71
12.34 Spinnaker.h . . . . .	72
12.35 Spinnaker Definitions . . . . .	73
12.36 Spinnaker Platform . . . . .	74
12.37 Spinnaker Video Class . . . . .	75
12.38 Spinnaker Video Definitions . . . . .	76
12.39 System Class . . . . .	77
12.40 SystemEventHandler Class . . . . .	78
12.41 SystemPtr Class . . . . .	79
12.42 Spinnaker QuickSpin Classes . . . . .	80
12.43 TransportLayerDefs Class . . . . .	81
12.44 TransportLayerDevice Class . . . . .	82
12.45 TransportLayerInterface Class . . . . .	83
12.46 TransportLayerStream Class . . . . .	84
12.47 TransportLayerSystem Class . . . . .	85
12.48 Camera Base Interface Class . . . . .	86
12.49 IChunkData Class . . . . .	87
12.50 IImage Class . . . . .	88
12.51 IImageStatistics Class . . . . .	89
12.52 IInterface Class . . . . .	90
12.53 IInterfaceList Class . . . . .	91
12.54 ISystem Class . . . . .	92
12.55 Spinnaker GenApi Classes . . . . .	93

---

---

12.56 AutoVector Class . . . . .	94
12.57 Spinnaker GenApi Interfaces . . . . .	95
12.58 IBase Interface . . . . .	96
12.59 BooleanNode Class . . . . .	97
12.60 CategoryNode Class . . . . .	98
12.61 ChunkAdapter Class . . . . .	99
12.62 ChunkAdapterDcam Class . . . . .	100
12.63 ChunkAdapterGeneric Class . . . . .	101
12.64 ChunkAdapterGEV Class . . . . .	102
12.65 ChunkPort Class . . . . .	103
12.66 CommandNode Class . . . . .	104
12.67 Container Class . . . . .	105
12.68 Counter Class . . . . .	106
12.69 EnumClasses Class . . . . .	107
12.70 EnumEntryNode Class . . . . .	108
12.71 EnumNode Class . . . . .	109
12.72 EnumNodeT Class . . . . .	110
12.73 EventAdapter Class . . . . .	111
12.74 EventAdapter1394 Class . . . . .	112
12.75 EventAdapterGeneric Class . . . . .	113
12.76 EventAdapterGEV Class . . . . .	114
12.77 EventAdapterU3V Class . . . . .	115
12.78 EventPort Class . . . . .	116
12.79 Filestream Class . . . . .	117
12.80 FloatNode Class . . . . .	118
12.81 FloatRegNode Class . . . . .	119
12.82 GCString Class . . . . .	120
12.83 GCSynch Class . . . . .	121
12.84 GCTypes Class . . . . .	122
12.84.1 Detailed Description . . . . .	122
12.84.2 Typedef Documentation . . . . .	122
12.84.2.1 float32_t . . . . .	122
12.84.2.2 float64_t . . . . .	122
12.85 Spinnaker GenApi Utilities . . . . .	123
12.86 GCUtilities Utility . . . . .	124
12.87 IBoolean Interface . . . . .	125
12.88 ICategory Interfaces . . . . .	126
12.89 IChunkPort Interface . . . . .	127
12.90 ICommand Interface . . . . .	128
12.91 IDestroy Interface . . . . .	129
12.92 IDeviceInfo Interface . . . . .	130
12.93 IEnumEntry Interface . . . . .	131

---

12.94 IEnumeration Interface . . . . .	132
12.95 IEnumerationT Interface . . . . .	133
12.96 IFloat Interface . . . . .	134
12.97 IInteger Interface . . . . .	135
12.98 INode Interface . . . . .	136
12.98.1 Detailed Description . . . . .	136
12.98.2 Function Documentation . . . . .	136
12.98.2.1 Combine() [1/3] . . . . .	137
12.98.2.2 Combine() [2/3] . . . . .	137
12.98.2.3 Combine() [3/3] . . . . .	137
12.98.2.4 IsAvailable() [1/3] . . . . .	137
12.98.2.5 IsAvailable() [2/3] . . . . .	137
12.98.2.6 IsAvailable() [3/3] . . . . .	138
12.98.2.7 IsCacheable() . . . . .	138
12.98.2.8 IsImplemented() [1/3] . . . . .	138
12.98.2.9 IsImplemented() [2/3] . . . . .	138
12.98.2.10 IsImplemented() [3/3] . . . . .	138
12.98.2.11 IsReadable() [1/3] . . . . .	138
12.98.2.12 IsReadable() [2/3] . . . . .	139
12.98.2.13 IsReadable() [3/3] . . . . .	139
12.98.2.14 IsVisible() . . . . .	139
12.98.2.15 IsWritable() [1/3] . . . . .	139
12.98.2.16 IsWritable() [2/3] . . . . .	139
12.98.2.17 IsWritable() [3/3] . . . . .	139
12.99 INodeMap Interface . . . . .	140
12.100 INodeMapDyn Interface . . . . .	141
12.101 IntegerNode Class . . . . .	142
12.102 IntRegNode Class . . . . .	143
12.103 IPort Interface . . . . .	144
12.104 IPortConstruct Interface . . . . .	145
12.105 IPortRecorder Interface . . . . .	146
12.106 IRegister Interfaces . . . . .	147
12.107 ISelector Interface . . . . .	148
12.108 ISelectorDigit Interface . . . . .	149
12.109 IString Class . . . . .	150
12.110 IValue Class . . . . .	151
12.111 Node Class . . . . .	152
12.112 NodeCallback Class . . . . .	153
12.113 NodeMap Class . . . . .	154
12.114 NodeMapFactory Class . . . . .	155
12.115 NodeMapRef Class . . . . .	156
12.116 Persistence Class . . . . .	157

---

---

12.117 Pointer Class . . . . .	158
12.117.1 Detailed Description . . . . .	159
12.117.2 Typedef Documentation . . . . .	159
12.117.2.1 CBasePtr . . . . .	159
12.117.2.2 CBooleanPtr . . . . .	160
12.117.2.3 CCategoryPtr . . . . .	160
12.117.2.4 CChunkPortPtr . . . . .	160
12.117.2.5 CCommandPtr . . . . .	160
12.117.2.6 CDeviceInfoPtr . . . . .	160
12.117.2.7 CEnumEntryPtr . . . . .	160
12.117.2.8 CEnumerationPtr . . . . .	161
12.117.2.9 CIIntegerPtr . . . . .	161
12.117.2.10 CNodeMapDynPtr . . . . .	161
12.117.2.11 CNodeMapPtr . . . . .	161
12.117.2.12 CNodePtr . . . . .	161
12.117.2.13 CPortConstructPtr . . . . .	161
12.117.2.14 CPortPtr . . . . .	162
12.117.2.15 CPortRecorderPtr . . . . .	162
12.117.2.16 CPortReplayPtr . . . . .	162
12.117.2.17 CPortWriteListPtr . . . . .	162
12.117.2.18 CRegisterPtr . . . . .	162
12.117.2.19 CSelectorPtr . . . . .	162
12.117.2.20 CStringPtr . . . . .	163
12.117.2.21 CValuePtr . . . . .	163
12.117.3 Function Documentation . . . . .	163
12.117.3.1 GetInterfaceName() . . . . .	163
12.117.3.2 IsAvailable() . . . . .	163
12.117.3.3 IsImplemented() . . . . .	163
12.117.3.4 IsReadable() . . . . .	163
12.117.3.5 IsWritable() . . . . .	163
12.118 PortImpl Class . . . . .	164
12.119 PortNode Class . . . . .	165
12.120 PortRecorder Class . . . . .	166
12.121 PortReplay Class . . . . .	167
12.122 PortWriteList Class . . . . .	168
12.123 Reference Interfaces . . . . .	169
12.124 RegisterNode Class . . . . .	170
12.125 RegisterPortImpl Class . . . . .	171
12.126 SelectorSet Class . . . . .	172
12.127 SpinTestCamera Class . . . . .	173
12.128 StringNode Class . . . . .	174
12.129 StringRegNode Class . . . . .	175

---

---

12.130 StructPort Class . . . . .	176
12.131 Synch Class . . . . .	177
12.132 Spinnaker GenApi Enums . . . . .	178
12.133 Types Enums . . . . .	179
12.134 ValueNode Class . . . . .	180
12.135 ChunkAdapterU3V Class . . . . .	181
<b>13 Namespace Documentation</b> . . . . .	<b>183</b>
13.1 AdapterConfig Namespace Reference . . . . .	183
13.1.1 Enumeration Type Documentation . . . . .	184
13.1.1.1 AdapterConfigErr . . . . .	184
13.1.2 Function Documentation . . . . .	184
13.1.2.1 AutoPopulateAdapterInfo() . . . . .	184
13.1.2.2 AutoPopulateAdvancedProperties() . . . . .	184
13.1.2.3 ConfigureAdapter() . . . . .	185
13.1.2.4 GetAuto10GDesc() . . . . .	185
13.1.2.5 GetAutoGigabitDesc() . . . . .	185
13.1.2.6 GetAutoStartIp() . . . . .	185
13.1.2.7 GetAutoSubnetMask() . . . . .	185
13.1.2.8 GetAutoSubnetMaskLength() . . . . .	185
13.1.2.9 GetConfigLogFileName() . . . . .	185
13.1.2.10 GetEnumerationLogFileName() . . . . .	186
13.1.2.11 GetMaxIpAddress() . . . . .	186
13.1.2.12 GetMinIpAddress() . . . . .	186
13.1.2.13 GetSubnetMaskLength() . . . . .	186
13.1.2.14 IsOnSameSubnet() . . . . .	186
13.1.2.15 IsValidIpAddress() . . . . .	186
13.1.2.16 IsValidSubnetMask() . . . . .	186
13.1.2.17 PopulateAdapterIpInfo() . . . . .	187
13.1.2.18 RetrieveAllAdapters() . . . . .	187
13.1.2.19 ValidateIpAddress() . . . . .	187
13.2 Conversion Namespace Reference . . . . .	187
13.2.1 Function Documentation . . . . .	187
13.2.1.1 NumToCString() [1/3] . . . . .	187
13.2.1.2 NumToCString() [2/3] . . . . .	187
13.2.1.3 NumToCString() [3/3] . . . . .	188
13.3 CpuUtil Namespace Reference . . . . .	188
13.3.1 Function Documentation . . . . .	188
13.3.1.1 GetCpuStats() . . . . .	188
13.3.1.2 StartCpuTracing() . . . . .	188
13.3.1.3 StopCpuTracing() . . . . .	188
13.4 PerformanceCounter Namespace Reference . . . . .	188

---

13.4.1 Function Documentation . . . . .	189
13.4.1.1 GetPerformanceCounter() . . . . .	189
13.4.1.2 StartPerformanceCounter() . . . . .	189
13.4.2 Variable Documentation . . . . .	189
13.4.2.1 CounterStart . . . . .	189
13.4.2.2 PCFreq . . . . .	189
13.5 SecondsCounter Namespace Reference . . . . .	189
13.5.1 Function Documentation . . . . .	190
13.5.1.1 GetSecondsCounter() . . . . .	190
13.5.1.2 StartSecondsCounter() . . . . .	190
13.5.2 Variable Documentation . . . . .	190
13.5.2.1 endTime . . . . .	190
13.5.2.2 startTime . . . . .	190
13.5.2.3 timeDiff . . . . .	190
13.6 Spinnaker Namespace Reference . . . . .	191
13.6.1 Enumeration Type Documentation . . . . .	230
13.6.1.1 AcquisitionModeEnums . . . . .	230
13.6.1.2 AcquisitionStatusSelectorEnums . . . . .	230
13.6.1.3 ActionCommandStatus . . . . .	231
13.6.1.4 ActionUnconditionalModeEnums . . . . .	231
13.6.1.5 AdcBitDepthEnums . . . . .	231
13.6.1.6 AutoAlgorithmSelectorEnums . . . . .	232
13.6.1.7 AutoExposureControlPriorityEnums . . . . .	232
13.6.1.8 AutoExposureLightingModeEnums . . . . .	232
13.6.1.9 AutoExposureMeteringModeEnums . . . . .	233
13.6.1.10 AutoExposureTargetGreyValueAutoEnums . . . . .	233
13.6.1.11 BalanceRatioSelectorEnums . . . . .	234
13.6.1.12 BalanceWhiteAutoEnums . . . . .	234
13.6.1.13 BalanceWhiteAutoProfileEnums . . . . .	234
13.6.1.14 BinningHorizontalModeEnums . . . . .	235
13.6.1.15 BinningSelectorEnums . . . . .	235
13.6.1.16 BinningVerticalModeEnums . . . . .	235
13.6.1.17 BlackLevelAutoBalanceEnums . . . . .	236
13.6.1.18 BlackLevelAutoEnums . . . . .	236
13.6.1.19 BlackLevelSelectorEnums . . . . .	236
13.6.1.20 BufferOwnership . . . . .	237
13.6.1.21 ChunkBlackLevelSelectorEnums . . . . .	237
13.6.1.22 ChunkCounterSelectorEnums . . . . .	237
13.6.1.23 ChunkEncoderSelectorEnums . . . . .	237
13.6.1.24 ChunkEncoderStatusEnums . . . . .	238
13.6.1.25 ChunkExposureTimeSelectorEnums . . . . .	238
13.6.1.26 ChunkGainSelectorEnums . . . . .	239

13.6.1.27 ChunkImageComponentEnums . . . . .	239
13.6.1.28 ChunkPixelFormatEnums . . . . .	239
13.6.1.29 ChunkRegionIDEnums . . . . .	240
13.6.1.30 ChunkScan3dCoordinateReferenceSelectorEnums . . . . .	240
13.6.1.31 ChunkScan3dCoordinateSelectorEnums . . . . .	241
13.6.1.32 ChunkScan3dCoordinateSystemEnums . . . . .	241
13.6.1.33 ChunkScan3dCoordinateSystemReferenceEnums . . . . .	241
13.6.1.34 ChunkScan3dCoordinateTransformSelectorEnums . . . . .	242
13.6.1.35 ChunkScan3dDistanceUnitEnums . . . . .	242
13.6.1.36 ChunkScan3dOutputModeEnums . . . . .	242
13.6.1.37 ChunkSelectorEnums . . . . .	243
13.6.1.38 ChunkSourceIDEnums . . . . .	244
13.6.1.39 ChunkTimerSelectorEnums . . . . .	244
13.6.1.40 ChunkTransferStreamIDEnums . . . . .	244
13.6.1.41 CIConfigurationEnums . . . . .	245
13.6.1.42 CITimeSlotsCountEnums . . . . .	245
13.6.1.43 ColorProcessingAlgorithm . . . . .	246
13.6.1.44 ColorTransformationSelectorEnums . . . . .	246
13.6.1.45 ColorTransformationValueSelectorEnums . . . . .	246
13.6.1.46 CounterEventActivationEnums . . . . .	247
13.6.1.47 CounterEventSourceEnums . . . . .	247
13.6.1.48 CounterResetActivationEnums . . . . .	248
13.6.1.49 CounterResetSourceEnums . . . . .	248
13.6.1.50 CounterSelectorEnums . . . . .	249
13.6.1.51 CounterStatusEnums . . . . .	249
13.6.1.52 CounterTriggerActivationEnums . . . . .	250
13.6.1.53 CounterTriggerSourceEnums . . . . .	250
13.6.1.54 CxpConnectionTestModeEnums . . . . .	251
13.6.1.55 CxpLinkConfigurationEnums . . . . .	251
13.6.1.56 CxpLinkConfigurationPreferredEnums . . . . .	252
13.6.1.57 CxpLinkConfigurationStatusEnums . . . . .	253
13.6.1.58 CxpPoCxpStatusEnums . . . . .	254
13.6.1.59 DecimationHorizontalModeEnums . . . . .	254
13.6.1.60 DecimationSelectorEnums . . . . .	254
13.6.1.61 DecimationVerticalModeEnums . . . . .	255
13.6.1.62 DefectCorrectionModeEnums . . . . .	255
13.6.1.63 DeinterlacingEnums . . . . .	255
13.6.1.64 DeviceAccessStatusEnum . . . . .	256
13.6.1.65 DeviceCharacterSetEnums . . . . .	256
13.6.1.66 DeviceClockSelectorEnums . . . . .	256
13.6.1.67 DeviceConnectionStatusEnums . . . . .	257
13.6.1.68 DeviceCurrentSpeedEnum . . . . .	257

---

13.6.1.69 DeviceEndianessMechanismEnum . . . . .	257
13.6.1.70 DeviceIndicatorModeEnums . . . . .	258
13.6.1.71 DeviceLinkHeartbeatModeEnums . . . . .	258
13.6.1.72 DeviceLinkThroughputLimitModeEnums . . . . .	258
13.6.1.73 DevicePowerSupplySelectorEnums . . . . .	259
13.6.1.74 DeviceRegistersEndiannessEnums . . . . .	259
13.6.1.75 DeviceScanTypeEnums . . . . .	259
13.6.1.76 DeviceSerialPortBaudRateEnums . . . . .	260
13.6.1.77 DeviceSerialPortSelectorEnums . . . . .	260
13.6.1.78 DeviceStreamChannelEndiannessEnums . . . . .	260
13.6.1.79 DeviceStreamChannelTypeEnums . . . . .	261
13.6.1.80 DeviceTapGeometryEnums . . . . .	261
13.6.1.81 DeviceTemperatureSelectorEnums . . . . .	262
13.6.1.82 DeviceTLTypeEnums . . . . .	262
13.6.1.83 DeviceTypeEnum . . . . .	263
13.6.1.84 DeviceTypeEnums . . . . .	263
13.6.1.85 EncoderModeEnums . . . . .	264
13.6.1.86 EncoderOutputModeEnums . . . . .	264
13.6.1.87 EncoderResetActivationEnums . . . . .	264
13.6.1.88 EncoderResetSourceEnums . . . . .	265
13.6.1.89 EncoderSelectorEnums . . . . .	266
13.6.1.90 EncoderSourceAEnums . . . . .	266
13.6.1.91 EncoderSourceBEnums . . . . .	267
13.6.1.92 EncoderStatusEnums . . . . .	267
13.6.1.93 Error . . . . .	267
13.6.1.94 EventNotificationEnums . . . . .	268
13.6.1.95 EventSelectorEnums . . . . .	270
13.6.1.96 EventType . . . . .	270
13.6.1.97 ExposureActiveModeEnums . . . . .	270
13.6.1.98 ExposureAutoEnums . . . . .	271
13.6.1.99 ExposureModeEnums . . . . .	271
13.6.1.100 ExposureTimeModeEnums . . . . .	271
13.6.1.101 ExposureTimeSelectorEnums . . . . .	273
13.6.1.102 FileModeEnums . . . . .	273
13.6.1.103 FileOperationSelectorEnums . . . . .	274
13.6.1.104 FileOperationStatusEnums . . . . .	274
13.6.1.105 FileSelectorEnums . . . . .	274
13.6.1.106 FilterDriverStatusEnum . . . . .	275
13.6.1.107 GainAutoBalanceEnums . . . . .	275
13.6.1.108 GainAutoEnums . . . . .	275
13.6.1.109 GainSelectorEnums . . . . .	276
13.6.1.110 GenICamXMLLocationEnum . . . . .	276

---

---

13.6.1.111 GevCCPEnum . . . . .	276
13.6.1.112 GevCCPEnums . . . . .	276
13.6.1.113 GevCurrentPhysicalLinkConfigurationEnums . . . . .	277
13.6.1.114 GevGVCPExtendedStatusCodesSelectorEnums . . . . .	277
13.6.1.115 GevGVSPExtendedIDModeEnums . . . . .	277
13.6.1.116 GevIEEE1588ClockAccuracyEnums . . . . .	278
13.6.1.117 GevIEEE1588ModeEnums . . . . .	278
13.6.1.118 GevIEEE1588StatusEnums . . . . .	278
13.6.1.119 GevIPConfigurationStatusEnums . . . . .	279
13.6.1.120 GevPhysicalLinkConfigurationEnums . . . . .	279
13.6.1.121 GevSupportedOptionSelectorEnums . . . . .	280
13.6.1.122 GUIXMLLocationEnum . . . . .	280
13.6.1.123 ImageComponentSelectorEnums . . . . .	281
13.6.1.124 ImageCompressionJPEGFormatOptionEnums . . . . .	281
13.6.1.125 ImageCompressionModeEnums . . . . .	282
13.6.1.126 ImageCompressionRateOptionEnums . . . . .	282
13.6.1.127 ImageFileFormat . . . . .	282
13.6.1.128 ImageStatus . . . . .	283
13.6.1.129 InferenceBoxType . . . . .	284
13.6.1.130 InterfaceTypeEnum . . . . .	284
13.6.1.131 LineFormatEnums . . . . .	284
13.6.1.132 LineInputFilterSelectorEnums . . . . .	285
13.6.1.133 LineModeEnums . . . . .	285
13.6.1.134 LineSelectorEnums . . . . .	285
13.6.1.135 LineSourceEnums . . . . .	286
13.6.1.136 LogicBlockLUTInputActivationEnums . . . . .	286
13.6.1.137 LogicBlockLUTInputSelectorEnums . . . . .	287
13.6.1.138 LogicBlockLUTInputSourceEnums . . . . .	287
13.6.1.139 LogicBlockLUTSelectorEnums . . . . .	288
13.6.1.140 LogicBlockSelectorEnums . . . . .	288
13.6.1.141 LUTSelectorEnums . . . . .	288
13.6.1.142 PayloadTypeInfoIDs . . . . .	288
13.6.1.143 PixelColorFilterEnums . . . . .	289
13.6.1.144 PixelFormatEnums . . . . .	289
13.6.1.145 PixelFormatInfoSelectorEnums . . . . .	295
13.6.1.146 PixelFormatIntType . . . . .	301
13.6.1.147 PixelFormatNamespaceID . . . . .	301
13.6.1.148 PixelSizeEnums . . . . .	302
13.6.1.149 POEStatusEnum . . . . .	302
13.6.1.150 RegionDestinationEnums . . . . .	303
13.6.1.151 RegionModeEnums . . . . .	303
13.6.1.152 RegionSelectorEnums . . . . .	303

---

13.6.1.153 RgbTransformLightSourceEnums . . . . .	304
13.6.1.154 Scan3dCoordinateReferenceSelectorEnums . . . . .	304
13.6.1.155 Scan3dCoordinateSelectorEnums . . . . .	305
13.6.1.156 Scan3dCoordinateSystemEnums . . . . .	305
13.6.1.157 Scan3dCoordinateSystemReferenceEnums . . . . .	305
13.6.1.158 Scan3dCoordinateTransformSelectorEnums . . . . .	306
13.6.1.159 Scan3dDistanceUnitEnums . . . . .	306
13.6.1.160 Scan3dOutputModeEnums . . . . .	306
13.6.1.161 SensorDigitizationTapsEnums . . . . .	307
13.6.1.162 SensorShutterModeEnums . . . . .	308
13.6.1.163 SensorTapsEnums . . . . .	308
13.6.1.164 SequencerConfigurationModeEnums . . . . .	308
13.6.1.165 SequencerConfigurationValidEnums . . . . .	309
13.6.1.166 SequencerModeEnums . . . . .	309
13.6.1.167 SequencerSetValidEnums . . . . .	309
13.6.1.168 SequencerTriggerActivationEnums . . . . .	309
13.6.1.169 SequencerTriggerSourceEnums . . . . .	310
13.6.1.170 SerialPortBaudRateEnums . . . . .	310
13.6.1.171 SerialPortParityEnums . . . . .	311
13.6.1.172 SerialPortSelectorEnums . . . . .	311
13.6.1.173 SerialPortSourceEnums . . . . .	311
13.6.1.174 SerialPortStopBitsEnums . . . . .	312
13.6.1.175 SoftwareSignalSelectorEnums . . . . .	312
13.6.1.176 SourceSelectorEnums . . . . .	312
13.6.1.177 SpinnakerLogLevel . . . . .	313
13.6.1.178 StatisticsChannel . . . . .	313
13.6.1.179 StreamBufferCountModeEnum . . . . .	313
13.6.1.180 StreamBufferHandlingModeEnum . . . . .	314
13.6.1.181 StreamTypeEnum . . . . .	314
13.6.1.182 TestPatternEnums . . . . .	315
13.6.1.183 TestPatternGeneratorSelectorEnums . . . . .	315
13.6.1.184 TimerSelectorEnums . . . . .	315
13.6.1.185 TimerStatusEnums . . . . .	317
13.6.1.186 TimerTriggerActivationEnums . . . . .	317
13.6.1.187 TimerTriggerSourceEnums . . . . .	317
13.6.1.188 TLTypeEnum . . . . .	319
13.6.1.189 TransferComponentSelectorEnums . . . . .	319
13.6.1.190 TransferControlModeEnums . . . . .	320
13.6.1.191 TransferOperationModeEnums . . . . .	320
13.6.1.192 TransferQueueModeEnums . . . . .	320
13.6.1.193 TransferSelectorEnums . . . . .	321
13.6.1.194 TransferStatusSelectorEnums . . . . .	321

---

13.6.1.195 TransferTriggerActivationEnums . . . . .	321
13.6.1.196 TransferTriggerModeEnums . . . . .	322
13.6.1.197 TransferTriggerSelectorEnums . . . . .	322
13.6.1.198 TransferTriggerSourceEnums . . . . .	323
13.6.1.199 TriggerActivationEnums . . . . .	324
13.6.1.200 TriggerModeEnums . . . . .	324
13.6.1.201 TriggerOverlapEnums . . . . .	324
13.6.1.202 TriggerSelectorEnums . . . . .	325
13.6.1.203 TriggerSourceEnums . . . . .	325
13.6.1.204 UserOutputSelectorEnums . . . . .	325
13.6.1.205 UserSetDefaultEnums . . . . .	326
13.6.1.206 UserSetSelectorEnums . . . . .	326
13.6.1.207 WhiteClipSelectorEnums . . . . .	326
13.6.2 Function Documentation . . . . .	327
13.6.2.1 DEPRECATED_CLASS() . . . . .	327
13.6.2.2 operator==() . . . . .	328
13.6.3 Variable Documentation . . . . .	329
13.6.3.1 EVENT_TIMEOUT_INFINITE . . . . .	329
13.6.3.2 EVENT_TIMEOUT_NONE . . . . .	329
13.7 Spinnaker::GenApi Namespace Reference . . . . .	329
13.7.1 Typedef Documentation . . . . .	344
13.7.1.1 CallbackHandleType . . . . .	344
13.7.1.2 CBooleanRef . . . . .	344
13.7.1.3 CCategoryRef . . . . .	345
13.7.1.4 CCommandRef . . . . .	345
13.7.1.5 CEnumEntryRef . . . . .	345
13.7.1.6 CEnumerationRef . . . . .	345
13.7.1.7 CFloatRef . . . . .	345
13.7.1.8 CIntegerRef . . . . .	345
13.7.1.9 CNodeMapRef . . . . .	345
13.7.1.10 CNodeRef . . . . .	345
13.7.1.11 CPortRecorderRef . . . . .	346
13.7.1.12 CPortRef . . . . .	346
13.7.1.13 CRegisterRef . . . . .	346
13.7.1.14 CSelectorRef . . . . .	346
13.7.1.15 CStringRef . . . . .	346
13.7.1.16 CValueRef . . . . .	346
13.7.1.17 IDevFileStream . . . . .	346
13.7.1.18 NodeList_t . . . . .	347
13.7.1.19 ODevFileStream . . . . .	347
13.7.1.20 StringList_t . . . . .	347
13.7.2 Enumeration Type Documentation . . . . .	347

---

---

13.7.2.1 _EAccessMode . . . . .	347
13.7.2.2 _ECachingMode . . . . .	347
13.7.2.3 _ECallbackType . . . . .	348
13.7.2.4 _EDisplayNotation . . . . .	348
13.7.2.5 _EEndianess . . . . .	348
13.7.2.6 _EGenApiSchemaVersion . . . . .	349
13.7.2.7 _EIncMode . . . . .	349
13.7.2.8 _EInputDirection . . . . .	349
13.7.2.9 _EInterfaceType . . . . .	350
13.7.2.10 _ELinkType . . . . .	350
13.7.2.11 _ENamespace . . . . .	350
13.7.2.12 _ERepresentation . . . . .	351
13.7.2.13 _ESign . . . . .	351
13.7.2.14 _ESlope . . . . .	351
13.7.2.15 _EStandardNameSpace . . . . .	352
13.7.2.16 _EVisibility . . . . .	352
13.7.2.17 _EXMLValidation . . . . .	352
13.7.2.18 _EYesNo . . . . .	353
13.7.2.19 ECacheUsage_t . . . . .	353
13.7.2.20 EContentType_t . . . . .	354
13.7.2.21 GVCP_MESSAGE_TAGS . . . . .	354
13.7.3 Function Documentation . . . . .	354
13.7.3.1 _ClearXMLCache() . . . . .	354
13.7.3.2 _Connect() [1/2] . . . . .	354
13.7.3.3 _Connect() [2/2] . . . . .	354
13.7.3.4 _GetDeviceName() . . . . .	355
13.7.3.5 _GetNode() . . . . .	355
13.7.3.6 _GetNodes() . . . . .	355
13.7.3.7 _GetSupportedSchemaVersions() . . . . .	355
13.7.3.8 _InvalidateNodes() . . . . .	355
13.7.3.9 _LoadXMLFromFile() . . . . .	355
13.7.3.10 _LoadXMLFromFileInject() . . . . .	355
13.7.3.11 _LoadXMLFromString() . . . . .	356
13.7.3.12 _LoadXMLFromStringInject() . . . . .	356
13.7.3.13 _LoadXMLFromZIPData() . . . . .	356
13.7.3.14 _LoadXMLFromZIPFile() . . . . .	356
13.7.3.15 _Poll() . . . . .	356
13.7.3.16 CacheChunkData() . . . . .	356
13.7.3.17 CastToIDestroy() . . . . .	357
13.7.3.18 Connect() [1/2] . . . . .	357
13.7.3.19 Connect() [2/2] . . . . .	357
13.7.3.20 Deregister() . . . . .	357

13.7.3.21 DeregisterCallback()	357
13.7.3.22 EatComments()	358
13.7.3.23 ExtractIndependentSubtree()	358
13.7.3.24 FromString()	358
13.7.3.25 Get()	358
13.7.3.26 GetAddress()	359
13.7.3.27 GetAlias()	359
13.7.3.28 GetCachingMode()	359
13.7.3.29 GetCastAlias()	359
13.7.3.30 GetChildren()	359
13.7.3.31 GetCookie()	360
13.7.3.32 GetCurrentEntry()	360
13.7.3.33 GetDescription()	360
13.7.3.34 GetDeviceName()	360
13.7.3.35 GetDeviceVersion()	360
13.7.3.36 GetDisplayName()	361
13.7.3.37 GetDisplayNotation()	361
13.7.3.38 GetDisplayPrecision()	361
13.7.3.39 GetDocuURL()	361
13.7.3.40 GetEntries()	361
13.7.3.41 GetEntry() [1/2]	361
13.7.3.42 GetEntry() [2/2]	362
13.7.3.43 GetEntryByName()	362
13.7.3.44 GetEventID()	362
13.7.3.45 GetGenApiVersion()	362
13.7.3.46 GetInc()	362
13.7.3.47 GetIncMode()	362
13.7.3.48 GetIntValue()	362
13.7.3.49 GetLength()	363
13.7.3.50 GetListOfValidValues()	363
13.7.3.51 GetLock()	363
13.7.3.52 GetMax()	363
13.7.3.53 GetMaxLength()	364
13.7.3.54 GetMin()	364
13.7.3.55 GetNameSpace()	364
13.7.3.56 GetNode()	364
13.7.3.57 GetNodeMap()	364
13.7.3.58 GetNumericValue()	364
13.7.3.59 GetNumNodes()	364
13.7.3.60 GetParents()	364
13.7.3.61 GetPollingTime()	365
13.7.3.62 GetPrincipalInterfaceType()	365

---

13.7.3.63 GetProductGuid()	365
13.7.3.64 GetProperty()	365
13.7.3.65 GetPropertyNames()	365
13.7.3.66 GetRepresentation()	366
13.7.3.67 GetSchemaVersion()	366
13.7.3.68 GetSelectedFeatures()	366
13.7.3.69 GetSelectingFeatures()	366
13.7.3.70 GetSelectorList()	366
13.7.3.71 GetStandardNameSpace()	367
13.7.3.72 GetSupportedSchemaVersions()	367
13.7.3.73 GetSwapEndianess()	367
13.7.3.74 GetSymbolic()	367
13.7.3.75 GetToolTip()	367
13.7.3.76 GetUnit()	367
13.7.3.77 GetValue()	367
13.7.3.78 GetVendorName()	368
13.7.3.79 GetVersionGuid()	368
13.7.3.80 GetVisibility()	368
13.7.3.81 HasInc()	368
13.7.3.82 ImposeAccessMode()	368
13.7.3.83 ImposeMax() [1/2]	369
13.7.3.84 ImposeMax() [2/2]	369
13.7.3.85 ImposeMin() [1/2]	369
13.7.3.86 ImposeMin() [2/2]	369
13.7.3.87 ImposeVisibility()	369
13.7.3.88 InvalidateNode()	369
13.7.3.89 InvalidateNodes()	370
13.7.3.90 IsAccessModeCacheable()	370
13.7.3.91 IsCachable()	370
13.7.3.92 IsDeprecated()	370
13.7.3.93 IsDone()	370
13.7.3.94 IsFeature()	371
13.7.3.95 IsSelfClearing()	371
13.7.3.96 IsStreamable()	371
13.7.3.97 IsValueCacheValid()	371
13.7.3.98 LoadXMLFromFile()	371
13.7.3.99 LoadXMLFromFileInject()	371
13.7.3.100 LoadXMLFromString()	372
13.7.3.101 LoadXMLFromStringInject()	372
13.7.3.102 LoadXMLFromZIPData()	372
13.7.3.103 LoadXMLFromZIPFile()	372
13.7.3.104 make_NodeCallback() [1/2]	372

---

13.7.3.105 make_NodeCallback() [2/2] . . . . .	373
13.7.3.106 MergeXMLFiles() . . . . .	373
13.7.3.107 operator"!="() . . . . .	373
13.7.3.108 operator()() . . . . .	373
13.7.3.109 operator*() . . . . .	373
13.7.3.110 operator<<() . . . . .	374
13.7.3.111 operator=() [1/5] . . . . .	374
13.7.3.112 operator=() [2/5] . . . . .	374
13.7.3.113 operator=() [3/5] . . . . .	374
13.7.3.114 operator=() [4/5] . . . . .	374
13.7.3.115 operator=() [5/5] . . . . .	375
13.7.3.116 operator==() . . . . .	375
13.7.3.117 operator>>() . . . . .	375
13.7.3.118 PersistFeature() . . . . .	375
13.7.3.119 Poll() . . . . .	375
13.7.3.120 PreprocessXMLFromFile() . . . . .	375
13.7.3.121 PreprocessXMLFromFileZIPFile() . . . . .	376
13.7.3.122 Register() [1/2] . . . . .	376
13.7.3.123 Register() [2/2] . . . . .	376
13.7.3.124 RegisterCallback() . . . . .	377
13.7.3.125 Replay() . . . . .	377
13.7.3.126 Restore() . . . . .	377
13.7.3.127 SET_GUID() . . . . .	377
13.7.3.128 SetCookie() . . . . .	377
13.7.3.129 SetIntValue() . . . . .	377
13.7.3.130 SetNext() . . . . .	378
13.7.3.131 SetNumEnums() . . . . .	378
13.7.3.132 StopRecording() . . . . .	378
13.7.3.133 ToString() [1/2] . . . . .	378
13.7.3.134 ToString() [2/2] . . . . .	378
13.7.3.135 Write() . . . . .	379
13.7.4 Variable Documentation . . . . .	379
13.7.4.1 Address . . . . .	379
13.7.4.2 COMMAND_MAGIC . . . . .	379
13.7.4.3 GENCP_COMMAND_HEADER_SIZE . . . . .	379
13.7.4.4 GENCP_EVENT_BASIC_SIZE . . . . .	379
13.7.4.5 GENCP_EVENT_CMD_ID . . . . .	380
13.7.4.6 IBase . . . . .	380
13.7.4.7 IBoolean . . . . .	380
13.7.4.8 ICategory . . . . .	380
13.7.4.9 IChunkPort . . . . .	380
13.7.4.10 ICommand . . . . .	380

---

---

13.7.4.11 IDestroy . . . . .	381
13.7.4.12 IDeviceInfo . . . . .	381
13.7.4.13 IEnumEntry . . . . .	381
13.7.4.14 IEnumeration . . . . .	381
13.7.4.15 IEnumerationT . . . . .	381
13.7.4.16 IEnumReference . . . . .	382
13.7.4.17 IFloat . . . . .	382
13.7.4.18 IInteger . . . . .	382
13.7.4.19 INode . . . . .	382
13.7.4.20 INodeMap . . . . .	382
13.7.4.21 INodeMapDyn . . . . .	383
13.7.4.22 Invalidate . . . . .	383
13.7.4.23 IPersistScript . . . . .	383
13.7.4.24 IPort . . . . .	383
13.7.4.25 IPortConstruct . . . . .	383
13.7.4.26 IPortRecorder . . . . .	383
13.7.4.27 IPortReplay . . . . .	384
13.7.4.28 IPortWriteList . . . . .	384
13.7.4.29 IReference . . . . .	384
13.7.4.30 IRegister . . . . .	384
13.7.4.31 ISelector . . . . .	384
13.7.4.32 ISelectorDigit . . . . .	385
13.7.4.33 IString . . . . .	385
13.7.4.34 IValue . . . . .	385
13.7.4.35 Length . . . . .	385
13.7.4.36 U3V_EVENT_PREFIX . . . . .	385
13.7.4.37 Verify . . . . .	385
13.8 Spinnaker::GenICam Namespace Reference . . . . .	386
13.8.1 Function Documentation . . . . .	387
13.8.1.1 DoesEnvironmentVariableExist() . . . . .	387
13.8.1.2 GetFiles() . . . . .	387
13.8.1.3 GetGenICamCacheFolder() . . . . .	388
13.8.1.4 GetGenICamCLProtocolFolder() . . . . .	388
13.8.1.5 GetGenICamLogConfig() . . . . .	388
13.8.1.6 getline() [1/2] . . . . .	388
13.8.1.7 getline() [2/2] . . . . .	388
13.8.1.8 GetModulePathFromFunction() . . . . .	389
13.8.1.9 GetValueOfEnvironmentVariable() [1/2] . . . . .	389
13.8.1.10 GetValueOfEnvironmentVariable() [2/2] . . . . .	389
13.8.1.11 INTEGRAL_CAST() . . . . .	389
13.8.1.12 INTEGRAL_CAST2() . . . . .	390
13.8.1.13 ReplaceEnvironmentVariables() . . . . .	390

---

---

13.8.1.14 SetGenICamCacheFolder() . . . . .	390
13.8.1.15 SetGenICamCLProtocolFolder() . . . . .	390
13.8.1.16 SetGenICamLogConfig() . . . . .	390
13.8.1.17 ThrowBadAlloc() . . . . .	390
13.8.1.18 Tokenize() . . . . .	390
13.8.1.19 UrlDecode() . . . . .	391
13.8.1.20 UrlEncode() . . . . .	391
13.9 Spinnaker::Video Namespace Reference . . . . .	391
<b>14 Class Documentation</b> . . . . .	<b>393</b>
14.1 ActionCommandResult Struct Reference . . . . .	393
14.1.1 Detailed Description . . . . .	393
14.1.2 Member Data Documentation . . . . .	393
14.1.2.1 DeviceAddress . . . . .	393
14.1.2.2 Status . . . . .	393
14.2 AdapterConfigException Class Reference . . . . .	394
14.2.1 Constructor & Destructor Documentation . . . . .	394
14.2.1.1 AdapterConfigException() [1/2] . . . . .	394
14.2.1.2 AdapterConfigException() [2/2] . . . . .	395
14.2.2 Member Function Documentation . . . . .	395
14.2.2.1 ErrCode() . . . . .	395
14.2.2.2 GetParamStr() . . . . .	395
14.3 AdapterInfo Struct Reference . . . . .	395
14.3.1 Constructor & Destructor Documentation . . . . .	396
14.3.1.1 AdapterInfo() . . . . .	396
14.3.2 Member Data Documentation . . . . .	396
14.3.2.1 adapterDescription . . . . .	396
14.3.2.2 adapterGUID . . . . .	396
14.3.2.3 adapterMACAddress . . . . .	397
14.3.2.4 adapterName . . . . .	397
14.3.2.5 dhcpEnabled . . . . .	397
14.3.2.6 ipInfo . . . . .	397
14.3.2.7 jumboPackets . . . . .	397
14.3.2.8 jumboPacketsRegKey . . . . .	397
14.3.2.9 jumboPacketValidValues . . . . .	397
14.3.2.10 receiveBuffers . . . . .	397
14.3.2.11 receiveBuffersMax . . . . .	398
14.3.2.12 receiveBuffersMin . . . . .	398
14.3.2.13 receiveBuffersRegKey . . . . .	398
14.3.2.14 receiveBuffersStep . . . . .	398
14.3.2.15 transmitBuffers . . . . .	398
14.3.2.16 transmitBuffersMax . . . . .	398

---

14.3.2.17 transmitBuffersMin . . . . .	398
14.3.2.18 transmitBuffersRegKey . . . . .	398
14.3.2.19 transmitBuffersStep . . . . .	399
14.4 AttachStatistics_t Struct Reference . . . . .	399
14.4.1 Detailed Description . . . . .	399
14.4.2 Member Data Documentation . . . . .	399
14.4.2.1 NumAttachedChunks . . . . .	399
14.4.2.2 NumChunkPorts . . . . .	399
14.4.2.3 NumChunks . . . . .	400
14.5 AutoLock Class Reference . . . . .	400
14.5.1 Constructor & Destructor Documentation . . . . .	400
14.5.1.1 AutoLock() . . . . .	400
14.5.1.2 ~AutoLock() . . . . .	400
14.6 AutoLock Class Reference . . . . .	400
14.6.1 Constructor & Destructor Documentation . . . . .	401
14.6.1.1 AutoLock() . . . . .	401
14.6.1.2 ~AutoLock() . . . . .	401
14.7 AVIOption Struct Reference . . . . .	401
14.7.1 Detailed Description . . . . .	401
14.7.2 Constructor & Destructor Documentation . . . . .	401
14.7.2.1 AVIOption() . . . . .	402
14.7.3 Member Data Documentation . . . . .	402
14.7.3.1 frameRate . . . . .	402
14.7.3.2 reserved . . . . .	402
14.8 BasePtr< T, B > Class Template Reference . . . . .	402
14.8.1 Detailed Description . . . . .	403
14.8.2 Constructor & Destructor Documentation . . . . .	403
14.8.2.1 BasePtr() [1/2] . . . . .	403
14.8.2.2 ~BasePtr() . . . . .	403
14.8.2.3 BasePtr() [2/2] . . . . .	404
14.8.3 Member Function Documentation . . . . .	404
14.8.3.1 get() . . . . .	404
14.8.3.2 isValid() . . . . .	404
14.8.3.3 operator bool() . . . . .	404
14.8.3.4 operator T*() . . . . .	404
14.8.3.5 operator->() . . . . .	405
14.8.3.6 operator=() [1/4] . . . . .	405
14.8.3.7 operator=() [2/4] . . . . .	405
14.8.3.8 operator=() [3/4] . . . . .	405
14.8.3.9 operator=() [4/4] . . . . .	405
14.8.3.10 operator==() [1/4] . . . . .	405
14.8.3.11 operator==() [2/4] . . . . .	406

---

---

14.8.3.12 operator==( [3/4] . . . . .	406
14.8.3.13 operator==( [4/4] . . . . .	406
14.8.4 Member Data Documentation . . . . .	406
14.8.4.1 m_pT . . . . .	406
14.9 BMPOption Struct Reference . . . . .	406
14.9.1 Detailed Description . . . . .	407
14.9.2 Constructor & Destructor Documentation . . . . .	407
14.9.2.1 BMPOption() . . . . .	407
14.9.3 Member Data Documentation . . . . .	407
14.9.3.1 indexedColor_8bit . . . . .	407
14.9.3.2 reserved . . . . .	407
14.10 BooleanNode Class Reference . . . . .	408
14.10.1 Detailed Description . . . . .	409
14.10.2 Constructor & Destructor Documentation . . . . .	409
14.10.2.1 BooleanNode() [1/2] . . . . .	409
14.10.2.2 BooleanNode() [2/2] . . . . .	409
14.10.2.3 ~BooleanNode() . . . . .	409
14.10.3 Member Function Documentation . . . . .	409
14.10.3.1 GetValue() . . . . .	409
14.10.3.2 operator=() . . . . .	410
14.10.3.3 SetReference() . . . . .	410
14.10.3.4 SetValue() . . . . .	410
14.11 Camera Class Reference . . . . .	411
14.11.1 Detailed Description . . . . .	437
14.11.2 Constructor & Destructor Documentation . . . . .	437
14.11.2.1 ~Camera() . . . . .	437
14.11.2.2 Camera() . . . . .	437
14.11.3 Member Function Documentation . . . . .	437
14.11.3.1 Init() . . . . .	437
14.11.4 Member Data Documentation . . . . .	437
14.11.4.1 AasRoiEnable . . . . .	438
14.11.4.2 AasRoiHeight . . . . .	438
14.11.4.3 AasRoiOffsetX . . . . .	438
14.11.4.4 AasRoiOffsetY . . . . .	438
14.11.4.5 AasRoiWidth . . . . .	439
14.11.4.6 AcquisitionAbort . . . . .	439
14.11.4.7 AcquisitionArm . . . . .	439
14.11.4.8 AcquisitionBurstFrameCount . . . . .	439
14.11.4.9 AcquisitionFrameCount . . . . .	440
14.11.4.10 AcquisitionFrameRate . . . . .	440
14.11.4.11 AcquisitionFrameRateEnable . . . . .	440
14.11.4.12 AcquisitionLineRate . . . . .	440

---

---

14.11.4.13 AcquisitionMode . . . . .	440
14.11.4.14 AcquisitionResultingFrameRate . . . . .	441
14.11.4.15 AcquisitionStart . . . . .	441
14.11.4.16 AcquisitionStatus . . . . .	441
14.11.4.17 AcquisitionStatusSelector . . . . .	441
14.11.4.18 AcquisitionStop . . . . .	441
14.11.4.19 ActionDeviceKey . . . . .	441
14.11.4.20 ActionGroupKey . . . . .	442
14.11.4.21 ActionGroupMask . . . . .	442
14.11.4.22 ActionQueueSize . . . . .	442
14.11.4.23 ActionSelector . . . . .	442
14.11.4.24 ActionUnconditionalMode . . . . .	442
14.11.4.25 AdaptiveCompressionEnable . . . . .	442
14.11.4.26 AdcBitDepth . . . . .	443
14.11.4.27 aPAUSEMACCtrlFramesReceived . . . . .	443
14.11.4.28 aPAUSEMACCtrlFramesTransmitted . . . . .	443
14.11.4.29 AutoAlgorithmSelector . . . . .	443
14.11.4.30 AutoExposureControlLoopDamping . . . . .	443
14.11.4.31 AutoExposureControlPriority . . . . .	444
14.11.4.32 AutoExposureEVCompensation . . . . .	444
14.11.4.33 AutoExposureExposureTimeLowerLimit . . . . .	444
14.11.4.34 AutoExposureExposureTimeUpperLimit . . . . .	444
14.11.4.35 AutoExposureGainLowerLimit . . . . .	445
14.11.4.36 AutoExposureGainUpperLimit . . . . .	445
14.11.4.37 AutoExposureGreyValueLowerLimit . . . . .	445
14.11.4.38 AutoExposureGreyValueUpperLimit . . . . .	445
14.11.4.39 AutoExposureLightingMode . . . . .	446
14.11.4.40 AutoExposureMeteringMode . . . . .	446
14.11.4.41 AutoExposureTargetGreyValue . . . . .	446
14.11.4.42 AutoExposureTargetGreyValueAuto . . . . .	447
14.11.4.43 BalanceRatio . . . . .	447
14.11.4.44 BalanceRatioSelector . . . . .	447
14.11.4.45 BalanceWhiteAuto . . . . .	447
14.11.4.46 BalanceWhiteAutoDamping . . . . .	448
14.11.4.47 BalanceWhiteAutoLowerLimit . . . . .	448
14.11.4.48 BalanceWhiteAutoProfile . . . . .	448
14.11.4.49 BalanceWhiteAutoUpperLimit . . . . .	448
14.11.4.50 BinningHorizontal . . . . .	449
14.11.4.51 BinningHorizontalMode . . . . .	449
14.11.4.52 BinningSelector . . . . .	449
14.11.4.53 BinningVertical . . . . .	449
14.11.4.54 BinningVerticalMode . . . . .	450

---

---

14.11.4.55 BlackLevel . . . . .	450
14.11.4.56 BlackLevelAuto . . . . .	450
14.11.4.57 BlackLevelAutoBalance . . . . .	450
14.11.4.58 BlackLevelClampingEnable . . . . .	450
14.11.4.59 BlackLevelRaw . . . . .	451
14.11.4.60 BlackLevelSelector . . . . .	451
14.11.4.61 ChunkBlackLevel . . . . .	451
14.11.4.62 ChunkBlackLevelSelector . . . . .	451
14.11.4.63 ChunkCounterSelector . . . . .	451
14.11.4.64 ChunkCounterValue . . . . .	452
14.11.4.65 ChunkCRC . . . . .	452
14.11.4.66 ChunkEnable . . . . .	452
14.11.4.67 ChunkEncoderSelector . . . . .	452
14.11.4.68 ChunkEncoderStatus . . . . .	452
14.11.4.69 ChunkEncoderValue . . . . .	452
14.11.4.70 ChunkExposureEndLineStatusAll . . . . .	453
14.11.4.71 ChunkExposureTime . . . . .	453
14.11.4.72 ChunkExposureTimeSelector . . . . .	453
14.11.4.73 ChunkFrameID . . . . .	453
14.11.4.74 ChunkGain . . . . .	453
14.11.4.75 ChunkGainSelector . . . . .	453
14.11.4.76 ChunkHeight . . . . .	454
14.11.4.77 ChunkImage . . . . .	454
14.11.4.78 ChunkImageComponent . . . . .	454
14.11.4.79 ChunkInferenceBoundingBoxResult . . . . .	454
14.11.4.80 ChunkInferenceConfidence . . . . .	454
14.11.4.81 ChunkInferenceFrameID . . . . .	454
14.11.4.82 ChunkInferenceResult . . . . .	455
14.11.4.83 ChunkLinePitch . . . . .	455
14.11.4.84 ChunkLineStatusAll . . . . .	455
14.11.4.85 ChunkModeActive . . . . .	455
14.11.4.86 ChunkOffsetX . . . . .	455
14.11.4.87 ChunkOffsetY . . . . .	455
14.11.4.88 ChunkPartSelector . . . . .	456
14.11.4.89 ChunkPixelDynamicRangeMax . . . . .	456
14.11.4.90 ChunkPixelDynamicRangeMin . . . . .	456
14.11.4.91 ChunkPixelFormat . . . . .	456
14.11.4.92 ChunkRegionID . . . . .	456
14.11.4.93 ChunkScan3dAxisMax . . . . .	456
14.11.4.94 ChunkScan3dAxisMin . . . . .	457
14.11.4.95 ChunkScan3dCoordinateOffset . . . . .	457
14.11.4.96 ChunkScan3dCoordinateReferenceSelector . . . . .	457

---

---

14.11.4.97 ChunkScan3dCoordinateReferenceValue . . . . .	457
14.11.4.98 ChunkScan3dCoordinateScale . . . . .	457
14.11.4.99 ChunkScan3dCoordinateSelector . . . . .	457
14.11.4.100 ChunkScan3dCoordinateSystem . . . . .	458
14.11.4.101 ChunkScan3dCoordinateSystemReference . . . . .	458
14.11.4.102 ChunkScan3dCoordinateTransformSelector . . . . .	458
14.11.4.103 ChunkScan3dDistanceUnit . . . . .	458
14.11.4.104 ChunkScan3dInvalidDataFlag . . . . .	458
14.11.4.105 ChunkScan3dInvalidDataValue . . . . .	458
14.11.4.106 ChunkScan3dOutputMode . . . . .	459
14.11.4.107 ChunkScan3dTransformValue . . . . .	459
14.11.4.108 ChunkScanLineSelector . . . . .	459
14.11.4.109 ChunkSelector . . . . .	459
14.11.4.110 ChunkSequencerSetActive . . . . .	459
14.11.4.111 ChunkSerialData . . . . .	459
14.11.4.112 ChunkSerialDataLength . . . . .	460
14.11.4.113 ChunkSerialReceiveOverflow . . . . .	460
14.11.4.114 ChunkSourceID . . . . .	460
14.11.4.115 ChunkStreamChannelID . . . . .	460
14.11.4.116 ChunkTimerSelector . . . . .	460
14.11.4.117 ChunkTimerValue . . . . .	460
14.11.4.118 ChunkTimestamp . . . . .	461
14.11.4.119 ChunkTimestampLatchValue . . . . .	461
14.11.4.120 ChunkTransferBlockID . . . . .	461
14.11.4.121 ChunkTransferQueueCurrentBlockCount . . . . .	461
14.11.4.122 ChunkTransferStreamID . . . . .	461
14.11.4.123 ChunkWidth . . . . .	461
14.11.4.124 CIConfiguration . . . . .	462
14.11.4.125 CITimeSlotsCount . . . . .	462
14.11.4.126 ColorTransformationEnable . . . . .	462
14.11.4.127 ColorTransformationSelector . . . . .	462
14.11.4.128 ColorTransformationValue . . . . .	462
14.11.4.129 ColorTransformationValueSelector . . . . .	463
14.11.4.130 CompressionRatio . . . . .	463
14.11.4.131 CounterDelay . . . . .	463
14.11.4.132 CounterDuration . . . . .	463
14.11.4.133 CounterEventActivation . . . . .	463
14.11.4.134 CounterEventSource . . . . .	464
14.11.4.135 CounterReset . . . . .	464
14.11.4.136 CounterResetActivation . . . . .	464
14.11.4.137 CounterResetSource . . . . .	464
14.11.4.138 CounterSelector . . . . .	464

---

---

14.11.4.139 CounterStatus . . . . .	464
14.11.4.140 CounterTriggerActivation . . . . .	465
14.11.4.141 CounterTriggerSource . . . . .	465
14.11.4.142 CounterValue . . . . .	465
14.11.4.143 CounterValueAtReset . . . . .	465
14.11.4.144 CxpConnectionSelector . . . . .	465
14.11.4.145 CxpConnectionTestErrorCount . . . . .	465
14.11.4.146 CxpConnectionTestMode . . . . .	466
14.11.4.147 CxpConnectionTestPacketCount . . . . .	466
14.11.4.148 CxpLinkConfiguration . . . . .	466
14.11.4.149 CxpLinkConfigurationPreferred . . . . .	466
14.11.4.150 CxpLinkConfigurationStatus . . . . .	466
14.11.4.151 CxpPoCxpAuto . . . . .	466
14.11.4.152 CxpPoCxpStatus . . . . .	467
14.11.4.153 CxpPoCxpTripReset . . . . .	467
14.11.4.154 CxpPoCxpTurnOff . . . . .	467
14.11.4.155 DecimationHorizontal . . . . .	467
14.11.4.156 DecimationHorizontalMode . . . . .	467
14.11.4.157 DecimationSelector . . . . .	468
14.11.4.158 DecimationVertical . . . . .	468
14.11.4.159 DecimationVerticalMode . . . . .	468
14.11.4.160 DefectCorrectionMode . . . . .	468
14.11.4.161 DefectCorrectStaticEnable . . . . .	469
14.11.4.162 DefectTableApply . . . . .	469
14.11.4.163 DefectTableCoordinateX . . . . .	469
14.11.4.164 DefectTableCoordinateY . . . . .	469
14.11.4.165 DefectTableFactoryRestore . . . . .	470
14.11.4.166 DefectTableIndex . . . . .	470
14.11.4.167 DefectTablePixelCount . . . . .	470
14.11.4.168 DefectTableSave . . . . .	470
14.11.4.169 Deinterlacing . . . . .	471
14.11.4.170 DeviceCharacterSet . . . . .	471
14.11.4.171 DeviceClockFrequency . . . . .	471
14.11.4.172 DeviceClockSelector . . . . .	471
14.11.4.173 DeviceConnectionSelector . . . . .	471
14.11.4.174 DeviceConnectionSpeed . . . . .	472
14.11.4.175 DeviceConnectionStatus . . . . .	472
14.11.4.176 DeviceEventChannelCount . . . . .	472
14.11.4.177 DeviceFamilyName . . . . .	472
14.11.4.178 DeviceFeaturePersistenceEnd . . . . .	472
14.11.4.179 DeviceFeaturePersistenceStart . . . . .	473
14.11.4.180 DeviceFirmwareVersion . . . . .	473

---

---

14.11.4.181 DeviceGenCPVersionMajor . . . . .	473
14.11.4.182 DeviceGenCPVersionMinor . . . . .	473
14.11.4.183 DeviceID . . . . .	473
14.11.4.184 DeviceIndicatorMode . . . . .	473
14.11.4.185 DeviceLinkBandwidthReserve . . . . .	474
14.11.4.186 DeviceLinkCommandTimeout . . . . .	474
14.11.4.187 DeviceLinkConnectionCount . . . . .	474
14.11.4.188 DeviceLinkCurrentThroughput . . . . .	474
14.11.4.189 DeviceLinkHeartbeatMode . . . . .	474
14.11.4.190 DeviceLinkHeartbeatTimeout . . . . .	475
14.11.4.191 DeviceLinkSelector . . . . .	475
14.11.4.192 DeviceLinkSpeed . . . . .	475
14.11.4.193 DeviceLinkThroughputLimit . . . . .	475
14.11.4.194 DeviceLinkThroughputLimitMode . . . . .	475
14.11.4.195 DeviceManifestEntrySelector . . . . .	476
14.11.4.196 DeviceManifestPrimaryURL . . . . .	476
14.11.4.197 DeviceManifestSchemaMajorVersion . . . . .	476
14.11.4.198 DeviceManifestSchemaMinorVersion . . . . .	476
14.11.4.199 DeviceManifestSecondaryURL . . . . .	476
14.11.4.200 DeviceManifestXMLMajorVersion . . . . .	476
14.11.4.201 DeviceManifestXMLMinorVersion . . . . .	477
14.11.4.202 DeviceManifestXMLSubMinorVersion . . . . .	477
14.11.4.203 DeviceManufacturerInfo . . . . .	477
14.11.4.204 DeviceMaxThroughput . . . . .	477
14.11.4.205 DeviceModelName . . . . .	477
14.11.4.206 DevicePowerSupplySelector . . . . .	478
14.11.4.207 DeviceRegistersCheck . . . . .	478
14.11.4.208 DeviceRegistersEndianness . . . . .	478
14.11.4.209 DeviceRegistersStreamingEnd . . . . .	478
14.11.4.210 DeviceRegistersStreamingStart . . . . .	478
14.11.4.211 DeviceRegistersValid . . . . .	479
14.11.4.212 DeviceReset . . . . .	479
14.11.4.213 DeviceScanType . . . . .	479
14.11.4.214 DeviceSerialNumber . . . . .	479
14.11.4.215 DeviceSerialPortBaudRate . . . . .	479
14.11.4.216 DeviceSerialPortSelector . . . . .	480
14.11.4.217 DeviceSFNCVersionMajor . . . . .	480
14.11.4.218 DeviceSFNCVersionMinor . . . . .	480
14.11.4.219 DeviceSFNCVersionSubMinor . . . . .	480
14.11.4.220 DeviceStreamChannelCount . . . . .	480
14.11.4.221 DeviceStreamChannelEndianness . . . . .	481
14.11.4.222 DeviceStreamChannelLink . . . . .	481

---

---

14.11.4.223 DeviceStreamChannelPacketSize . . . . .	481
14.11.4.224 DeviceStreamChannelSelector . . . . .	481
14.11.4.225 DeviceStreamChannelType . . . . .	481
14.11.4.226 DeviceTapGeometry . . . . .	481
14.11.4.227 DeviceTemperature . . . . .	482
14.11.4.228 DeviceTemperatureSelector . . . . .	482
14.11.4.229 DeviceTLType . . . . .	482
14.11.4.230 DeviceTLVersionMajor . . . . .	482
14.11.4.231 DeviceTLVersionMinor . . . . .	482
14.11.4.232 DeviceTLVersionSubMinor . . . . .	483
14.11.4.233 DeviceType . . . . .	483
14.11.4.234 DeviceUptime . . . . .	483
14.11.4.235 DeviceUserID . . . . .	483
14.11.4.236 DeviceVendorName . . . . .	483
14.11.4.237 DeviceVersion . . . . .	483
14.11.4.238 EncoderDivider . . . . .	484
14.11.4.239 EncoderMode . . . . .	484
14.11.4.240 EncoderOutputMode . . . . .	484
14.11.4.241 EncoderReset . . . . .	484
14.11.4.242 EncoderResetActivation . . . . .	484
14.11.4.243 EncoderResetSource . . . . .	484
14.11.4.244 EncoderSelector . . . . .	485
14.11.4.245 EncoderSourceA . . . . .	485
14.11.4.246 EncoderSourceB . . . . .	485
14.11.4.247 EncoderStatus . . . . .	485
14.11.4.248 EncoderTimeout . . . . .	485
14.11.4.249 EncoderValue . . . . .	485
14.11.4.250 EncoderValueAtReset . . . . .	486
14.11.4.251 EnumerationCount . . . . .	486
14.11.4.252 EventAcquisitionEnd . . . . .	486
14.11.4.253 EventAcquisitionEndFrameID . . . . .	486
14.11.4.254 EventAcquisitionEndTimestamp . . . . .	486
14.11.4.255 EventAcquisitionError . . . . .	486
14.11.4.256 EventAcquisitionErrorFrameID . . . . .	487
14.11.4.257 EventAcquisitionErrorTimestamp . . . . .	487
14.11.4.258 EventAcquisitionStart . . . . .	487
14.11.4.259 EventAcquisitionStartFrameID . . . . .	487
14.11.4.260 EventAcquisitionStartTimestamp . . . . .	487
14.11.4.261 EventAcquisitionTransferEnd . . . . .	487
14.11.4.262 EventAcquisitionTransferEndFrameID . . . . .	488
14.11.4.263 EventAcquisitionTransferEndTimestamp . . . . .	488
14.11.4.264 EventAcquisitionTransferStart . . . . .	488

---

---

14.11.4.265 EventAcquisitionTransferStartFrameID . . . . .	488
14.11.4.266 EventAcquisitionTransferStartTimestamp . . . . .	488
14.11.4.267 EventAcquisitionTrigger . . . . .	488
14.11.4.268 EventAcquisitionTriggerFrameID . . . . .	489
14.11.4.269 EventAcquisitionTriggerTimestamp . . . . .	489
14.11.4.270 EventActionLate . . . . .	489
14.11.4.271 EventActionLateFrameID . . . . .	489
14.11.4.272 EventActionLateTimestamp . . . . .	489
14.11.4.273 EventCounter0End . . . . .	489
14.11.4.274 EventCounter0EndFrameID . . . . .	490
14.11.4.275 EventCounter0EndTimestamp . . . . .	490
14.11.4.276 EventCounter0Start . . . . .	490
14.11.4.277 EventCounter0StartFrameID . . . . .	490
14.11.4.278 EventCounter0StartTimestamp . . . . .	490
14.11.4.279 EventCounter1End . . . . .	490
14.11.4.280 EventCounter1EndFrameID . . . . .	491
14.11.4.281 EventCounter1EndTimestamp . . . . .	491
14.11.4.282 EventCounter1Start . . . . .	491
14.11.4.283 EventCounter1StartFrameID . . . . .	491
14.11.4.284 EventCounter1StartTimestamp . . . . .	491
14.11.4.285 EventEncoder0Restarted . . . . .	491
14.11.4.286 EventEncoder0RestartedFrameID . . . . .	492
14.11.4.287 EventEncoder0RestartedTimestamp . . . . .	492
14.11.4.288 EventEncoder0Stopped . . . . .	492
14.11.4.289 EventEncoder0StoppedFrameID . . . . .	492
14.11.4.290 EventEncoder0StoppedTimestamp . . . . .	492
14.11.4.291 EventEncoder1Restarted . . . . .	492
14.11.4.292 EventEncoder1RestartedFrameID . . . . .	493
14.11.4.293 EventEncoder1RestartedTimestamp . . . . .	493
14.11.4.294 EventEncoder1Stopped . . . . .	493
14.11.4.295 EventEncoder1StoppedFrameID . . . . .	493
14.11.4.296 EventEncoder1StoppedTimestamp . . . . .	493
14.11.4.297 EventError . . . . .	493
14.11.4.298 EventErrorCode . . . . .	494
14.11.4.299 EventErrorFrameID . . . . .	494
14.11.4.300 EventErrorTimestamp . . . . .	494
14.11.4.301 EventExposureEnd . . . . .	494
14.11.4.302 EventExposureEndFrameID . . . . .	494
14.11.4.303 EventExposureEndTimestamp . . . . .	494
14.11.4.304 EventExposureStart . . . . .	495
14.11.4.305 EventExposureStartFrameID . . . . .	495
14.11.4.306 EventExposureStartTimestamp . . . . .	495

---

---

14.11.4.307 EventFrameBurstEnd . . . . .	495
14.11.4.308 EventFrameBurstEndFrameID . . . . .	495
14.11.4.309 EventFrameBurstEndTimestamp . . . . .	495
14.11.4.310 EventFrameBurstStart . . . . .	496
14.11.4.311 EventFrameBurstStartFrameID . . . . .	496
14.11.4.312 EventFrameBurstStartTimestamp . . . . .	496
14.11.4.313 EventFrameEnd . . . . .	496
14.11.4.314 EventFrameEndFrameID . . . . .	496
14.11.4.315 EventFrameEndTimestamp . . . . .	496
14.11.4.316 EventFrameStart . . . . .	497
14.11.4.317 EventFrameStartFrameID . . . . .	497
14.11.4.318 EventFrameStartTimestamp . . . . .	497
14.11.4.319 EventFrameTransferEnd . . . . .	497
14.11.4.320 EventFrameTransferEndFrameID . . . . .	497
14.11.4.321 EventFrameTransferEndTimestamp . . . . .	497
14.11.4.322 EventFrameTransferStart . . . . .	498
14.11.4.323 EventFrameTransferStartFrameID . . . . .	498
14.11.4.324 EventFrameTransferStartTimestamp . . . . .	498
14.11.4.325 EventFrameTrigger . . . . .	498
14.11.4.326 EventFrameTriggerFrameID . . . . .	498
14.11.4.327 EventFrameTriggerTimestamp . . . . .	498
14.11.4.328 EventLine0AnyEdge . . . . .	499
14.11.4.329 EventLine0AnyEdgeFrameID . . . . .	499
14.11.4.330 EventLine0AnyEdgeTimestamp . . . . .	499
14.11.4.331 EventLine0FallingEdge . . . . .	499
14.11.4.332 EventLine0FallingEdgeFrameID . . . . .	499
14.11.4.333 EventLine0FallingEdgeTimestamp . . . . .	499
14.11.4.334 EventLine0RisingEdge . . . . .	500
14.11.4.335 EventLine0RisingEdgeFrameID . . . . .	500
14.11.4.336 EventLine0RisingEdgeTimestamp . . . . .	500
14.11.4.337 EventLine1AnyEdge . . . . .	500
14.11.4.338 EventLine1AnyEdgeFrameID . . . . .	500
14.11.4.339 EventLine1AnyEdgeTimestamp . . . . .	500
14.11.4.340 EventLine1FallingEdge . . . . .	501
14.11.4.341 EventLine1FallingEdgeFrameID . . . . .	501
14.11.4.342 EventLine1FallingEdgeTimestamp . . . . .	501
14.11.4.343 EventLine1RisingEdge . . . . .	501
14.11.4.344 EventLine1RisingEdgeFrameID . . . . .	501
14.11.4.345 EventLine1RisingEdgeTimestamp . . . . .	501
14.11.4.346 EventLinkSpeedChange . . . . .	502
14.11.4.347 EventLinkSpeedChangeFrameID . . . . .	502
14.11.4.348 EventLinkSpeedChangeTimestamp . . . . .	502

---

---

14.11.4.349 EventLinkTrigger0 . . . . .	502
14.11.4.350 EventLinkTrigger0FrameID . . . . .	502
14.11.4.351 EventLinkTrigger0Timestamp . . . . .	502
14.11.4.352 EventLinkTrigger1 . . . . .	503
14.11.4.353 EventLinkTrigger1FrameID . . . . .	503
14.11.4.354 EventLinkTrigger1Timestamp . . . . .	503
14.11.4.355 EventNotification . . . . .	503
14.11.4.356 EventSelector . . . . .	503
14.11.4.357 EventSequencerSetChange . . . . .	503
14.11.4.358 EventSequencerSetChangeFrameID . . . . .	504
14.11.4.359 EventSequencerSetChangeTimestamp . . . . .	504
14.11.4.360 EventSerialData . . . . .	504
14.11.4.361 EventSerialDataLength . . . . .	504
14.11.4.362 EventSerialPortReceive . . . . .	504
14.11.4.363 EventSerialPortReceiveTimestamp . . . . .	504
14.11.4.364 EventSerialReceiveOverflow . . . . .	505
14.11.4.365 EventStream0TransferBlockEnd . . . . .	505
14.11.4.366 EventStream0TransferBlockEndFrameID . . . . .	505
14.11.4.367 EventStream0TransferBlockEndTimestamp . . . . .	505
14.11.4.368 EventStream0TransferBlockStart . . . . .	505
14.11.4.369 EventStream0TransferBlockStartFrameID . . . . .	505
14.11.4.370 EventStream0TransferBlockStartTimestamp . . . . .	506
14.11.4.371 EventStream0TransferBlockTrigger . . . . .	506
14.11.4.372 EventStream0TransferBlockTriggerFrameID . . . . .	506
14.11.4.373 EventStream0TransferBlockTriggerTimestamp . . . . .	506
14.11.4.374 EventStream0TransferBurstEnd . . . . .	506
14.11.4.375 EventStream0TransferBurstEndFrameID . . . . .	506
14.11.4.376 EventStream0TransferBurstEndTimestamp . . . . .	507
14.11.4.377 EventStream0TransferBurstStart . . . . .	507
14.11.4.378 EventStream0TransferBurstStartFrameID . . . . .	507
14.11.4.379 EventStream0TransferBurstStartTimestamp . . . . .	507
14.11.4.380 EventStream0TransferEnd . . . . .	507
14.11.4.381 EventStream0TransferEndFrameID . . . . .	507
14.11.4.382 EventStream0TransferEndTimestamp . . . . .	508
14.11.4.383 EventStream0TransferOverflow . . . . .	508
14.11.4.384 EventStream0TransferOverflowFrameID . . . . .	508
14.11.4.385 EventStream0TransferOverflowTimestamp . . . . .	508
14.11.4.386 EventStream0TransferPause . . . . .	508
14.11.4.387 EventStream0TransferPauseFrameID . . . . .	508
14.11.4.388 EventStream0TransferPauseTimestamp . . . . .	509
14.11.4.389 EventStream0TransferResume . . . . .	509
14.11.4.390 EventStream0TransferResumeFrameID . . . . .	509

---

14.11.4.391 EventStream0TransferResumeTimestamp	509
14.11.4.392 EventStream0TransferStart	509
14.11.4.393 EventStream0TransferStartFrameID	509
14.11.4.394 EventStream0TransferStartTimestamp	510
14.11.4.395 EventTest	510
14.11.4.396 EventTestTimestamp	510
14.11.4.397 EventTimer0End	510
14.11.4.398 EventTimer0EndFrameID	510
14.11.4.399 EventTimer0EndTimestamp	510
14.11.4.400 EventTimer0Start	511
14.11.4.401 EventTimer0StartFrameID	511
14.11.4.402 EventTimer0StartTimestamp	511
14.11.4.403 EventTimer1End	511
14.11.4.404 EventTimer1EndFrameID	511
14.11.4.405 EventTimer1EndTimestamp	511
14.11.4.406 EventTimer1Start	512
14.11.4.407 EventTimer1StartFrameID	512
14.11.4.408 EventTimer1StartTimestamp	512
14.11.4.409 ExposureActiveMode	512
14.11.4.410 ExposureAuto	512
14.11.4.411 ExposureMode	512
14.11.4.412 ExposureTime	513
14.11.4.413 ExposureTimeMode	513
14.11.4.414 ExposureTimeSelector	513
14.11.4.415 FactoryReset	513
14.11.4.416 FileAccessBuffer	513
14.11.4.417 FileAccessLength	514
14.11.4.418 FileAccessOffset	514
14.11.4.419 FileMode	514
14.11.4.420 FileOperationExecute	514
14.11.4.421 FileOperationResult	514
14.11.4.422 FileOperationSelector	515
14.11.4.423 FileOperationStatus	515
14.11.4.424 FileSelector	515
14.11.4.425 FileSize	515
14.11.4.426 Gain	515
14.11.4.427 GainAuto	516
14.11.4.428 GainAutoBalance	516
14.11.4.429 GainSelector	516
14.11.4.430 Gamma	516
14.11.4.431 GammaEnable	516
14.11.4.432 GevActiveLinkCount	517

---

14.11.4.433 GevCCP . . . . .	517
14.11.4.434 GevCurrentDefaultGateway . . . . .	517
14.11.4.435 GevCurrentIPAddress . . . . .	517
14.11.4.436 GevCurrentIPConfigurationDHCP . . . . .	517
14.11.4.437 GevCurrentIPConfigurationLLA . . . . .	517
14.11.4.438 GevCurrentIPConfigurationPersistentIP . . . . .	518
14.11.4.439 GevCurrentPhysicalLinkConfiguration . . . . .	518
14.11.4.440 GevCurrentSubnetMask . . . . .	518
14.11.4.441 GevDiscoveryAckDelay . . . . .	518
14.11.4.442 GevFirstURL . . . . .	518
14.11.4.443 GevGVCPExtendedStatusCodes . . . . .	518
14.11.4.444 GevGVCPExtendedStatusCodesSelector . . . . .	519
14.11.4.445 GevGVCPHeartbeatDisable . . . . .	519
14.11.4.446 GevGVCPPendingAck . . . . .	519
14.11.4.447 GevGVCPPendingTimeout . . . . .	519
14.11.4.448 GevGVSPExtendedIDMode . . . . .	519
14.11.4.449 GevHeartbeatTimeout . . . . .	519
14.11.4.450 GevIEEE1588 . . . . .	520
14.11.4.451 GevIEEE1588ClockAccuracy . . . . .	520
14.11.4.452 GevIEEE1588Mode . . . . .	520
14.11.4.453 GevIEEE1588Status . . . . .	520
14.11.4.454 GevInterfaceSelector . . . . .	520
14.11.4.455 GevIPConfigurationStatus . . . . .	520
14.11.4.456 GevMACAddress . . . . .	521
14.11.4.457 GevMCDA . . . . .	521
14.11.4.458 GevMCPHostPort . . . . .	521
14.11.4.459 GevMCRC . . . . .	521
14.11.4.460 GevMCSP . . . . .	521
14.11.4.461 GevMCTT . . . . .	521
14.11.4.462 GevNumberOfInterfaces . . . . .	522
14.11.4.463 GevPAUSEFrameReception . . . . .	522
14.11.4.464 GevPAUSEFrameTransmission . . . . .	522
14.11.4.465 GevPersistentDefaultGateway . . . . .	522
14.11.4.466 GevPersistentIPAddress . . . . .	522
14.11.4.467 GevPersistentSubnetMask . . . . .	522
14.11.4.468 GevPhysicalLinkConfiguration . . . . .	523
14.11.4.469 GevPrimaryApplicationIPAddress . . . . .	523
14.11.4.470 GevPrimaryApplicationSocket . . . . .	523
14.11.4.471 GevPrimaryApplicationSwitchoverKey . . . . .	523
14.11.4.472 GevSCCFGAllInTransmission . . . . .	523
14.11.4.473 GevSCCFGExtendedChunkData . . . . .	523
14.11.4.474 GevSCCFGPacketResendDestination . . . . .	524

---

---

14.11.4.475 GevSCCFGUnconditionalStreaming . . . . .	524
14.11.4.476 GevSCDA . . . . .	524
14.11.4.477 GevSCPDir . . . . .	524
14.11.4.478 GevSCPDir . . . . .	524
14.11.4.479 GevSCPHostPort . . . . .	524
14.11.4.480 GevSCPIfaceIndex . . . . .	525
14.11.4.481 GevSCPSBigEndian . . . . .	525
14.11.4.482 GevSCPSDoNotFragment . . . . .	525
14.11.4.483 GevSCPSFireTestPacket . . . . .	525
14.11.4.484 GevSCPSPacketSize . . . . .	525
14.11.4.485 GevSCSP . . . . .	525
14.11.4.486 GevSCZoneConfigLock . . . . .	526
14.11.4.487 GevSCZoneCount . . . . .	526
14.11.4.488 GevSCZoneDirectionAll . . . . .	526
14.11.4.489 GevSecondURL . . . . .	526
14.11.4.490 GevStreamChannelSelector . . . . .	526
14.11.4.491 GevSupportedOption . . . . .	526
14.11.4.492 GevSupportedOptionSelector . . . . .	527
14.11.4.493 GevTimestampTickFrequency . . . . .	527
14.11.4.494 GuiXmlManifestAddress . . . . .	527
14.11.4.495 Height . . . . .	527
14.11.4.496 HeightMax . . . . .	527
14.11.4.497 ImageComponentEnable . . . . .	528
14.11.4.498 ImageComponentSelector . . . . .	528
14.11.4.499 ImageCompressionBitrate . . . . .	528
14.11.4.500 ImageCompressionJPEGFormatOption . . . . .	528
14.11.4.501 ImageCompressionMode . . . . .	528
14.11.4.502 ImageCompressionQuality . . . . .	528
14.11.4.503 ImageCompressionRateOption . . . . .	529
14.11.4.504 IspEnable . . . . .	529
14.11.4.505 LineFilterWidth . . . . .	529
14.11.4.506 LineFormat . . . . .	529
14.11.4.507 LineInputFilterSelector . . . . .	529
14.11.4.508 LineInverter . . . . .	530
14.11.4.509 LineMode . . . . .	530
14.11.4.510 LinePitch . . . . .	530
14.11.4.511 LineSelector . . . . .	530
14.11.4.512 LineSource . . . . .	530
14.11.4.513 LineStatus . . . . .	530
14.11.4.514 LineStatusAll . . . . .	531
14.11.4.515 LinkErrorCount . . . . .	531
14.11.4.516 LinkUptime . . . . .	531

---

---

14.11.4.517 LogicBlockLUTInputActivation . . . . .	531
14.11.4.518 LogicBlockLUTInputSelector . . . . .	531
14.11.4.519 LogicBlockLUTInputSource . . . . .	531
14.11.4.520 LogicBlockLUTOutputValue . . . . .	532
14.11.4.521 LogicBlockLUTOutputValueAll . . . . .	532
14.11.4.522 LogicBlockLUTRowIndex . . . . .	532
14.11.4.523 LogicBlockLUTSelector . . . . .	532
14.11.4.524 LogicBlockSelector . . . . .	532
14.11.4.525 LUTEnable . . . . .	532
14.11.4.526 LUTIndex . . . . .	533
14.11.4.527 LUTSelector . . . . .	533
14.11.4.528 LUTValue . . . . .	533
14.11.4.529 LUTValueAll . . . . .	533
14.11.4.530 MaxDeviceResetTime . . . . .	534
14.11.4.531 OffsetX . . . . .	534
14.11.4.532 OffsetY . . . . .	534
14.11.4.533 PacketResendRequestCount . . . . .	534
14.11.4.534 PayloadSize . . . . .	534
14.11.4.535 PixelColorFilter . . . . .	535
14.11.4.536 PixelDynamicRangeMax . . . . .	535
14.11.4.537 PixelDynamicRangeMin . . . . .	535
14.11.4.538 PixelFormat . . . . .	535
14.11.4.539 PixelFormatInfoID . . . . .	535
14.11.4.540 PixelFormatInfoSelector . . . . .	536
14.11.4.541 PixelSize . . . . .	536
14.11.4.542 PowerSupplyCurrent . . . . .	536
14.11.4.543 PowerSupplyVoltage . . . . .	536
14.11.4.544 RegionDestination . . . . .	536
14.11.4.545 RegionMode . . . . .	537
14.11.4.546 RegionSelector . . . . .	537
14.11.4.547 ReverseX . . . . .	537
14.11.4.548 ReverseY . . . . .	537
14.11.4.549 RgbTransformLightSource . . . . .	537
14.11.4.550 Saturation . . . . .	538
14.11.4.551 SaturationEnable . . . . .	538
14.11.4.552 Scan3dAxisMax . . . . .	538
14.11.4.553 Scan3dAxisMin . . . . .	538
14.11.4.554 Scan3dCoordinateOffset . . . . .	538
14.11.4.555 Scan3dCoordinateReferenceSelector . . . . .	538
14.11.4.556 Scan3dCoordinateReferenceValue . . . . .	539
14.11.4.557 Scan3dCoordinateScale . . . . .	539
14.11.4.558 Scan3dCoordinateSelector . . . . .	539

---

14.11.4.559 Scan3dCoordinateSystem . . . . .	539
14.11.4.560 Scan3dCoordinateSystemReference . . . . .	539
14.11.4.561 Scan3dCoordinateTransformSelector . . . . .	539
14.11.4.562 Scan3dDistanceUnit . . . . .	540
14.11.4.563 Scan3dInvalidDataFlag . . . . .	540
14.11.4.564 Scan3dInvalidDataValue . . . . .	540
14.11.4.565 Scan3dOutputMode . . . . .	540
14.11.4.566 Scan3dTransformValue . . . . .	540
14.11.4.567 SensorDescription . . . . .	540
14.11.4.568 SensorDigitizationTaps . . . . .	541
14.11.4.569 SensorHeight . . . . .	541
14.11.4.570 SensorShutterMode . . . . .	541
14.11.4.571 SensorTaps . . . . .	541
14.11.4.572 SensorWidth . . . . .	541
14.11.4.573 SequencerConfigurationMode . . . . .	541
14.11.4.574 SequencerConfigurationValid . . . . .	542
14.11.4.575 SequencerFeatureEnable . . . . .	542
14.11.4.576 SequencerMode . . . . .	542
14.11.4.577 SequencerPathSelector . . . . .	542
14.11.4.578 SequencerSetActive . . . . .	543
14.11.4.579 SequencerSetLoad . . . . .	543
14.11.4.580 SequencerSetNext . . . . .	543
14.11.4.581 SequencerSetSave . . . . .	543
14.11.4.582 SequencerSetSelector . . . . .	543
14.11.4.583 SequencerSetStart . . . . .	544
14.11.4.584 SequencerSetValid . . . . .	544
14.11.4.585 SequencerTriggerActivation . . . . .	544
14.11.4.586 SequencerTriggerSource . . . . .	544
14.11.4.587 SerialPortBaudRate . . . . .	545
14.11.4.588 SerialPortDataBits . . . . .	545
14.11.4.589 SerialPortParity . . . . .	545
14.11.4.590 SerialPortSelector . . . . .	545
14.11.4.591 SerialPortSource . . . . .	545
14.11.4.592 SerialPortStopBits . . . . .	545
14.11.4.593 SerialReceiveFramingErrorCount . . . . .	546
14.11.4.594 SerialReceiveParityErrorCount . . . . .	546
14.11.4.595 SerialReceiveQueueClear . . . . .	546
14.11.4.596 SerialReceiveQueueCurrentCharacterCount . . . . .	546
14.11.4.597 SerialReceiveQueueMaxCharacterCount . . . . .	546
14.11.4.598 SerialTransmitQueueCurrentCharacterCount . . . . .	546
14.11.4.599 SerialTransmitQueueMaxCharacterCount . . . . .	547
14.11.4.600 Sharpening . . . . .	547

---

14.11.4.601 SharpeningAuto . . . . .	547
14.11.4.602 SharpeningEnable . . . . .	547
14.11.4.603 SharpeningThreshold . . . . .	548
14.11.4.604 SoftwareSignalPulse . . . . .	548
14.11.4.605 SoftwareSignalSelector . . . . .	548
14.11.4.606 SourceCount . . . . .	548
14.11.4.607 SourceSelector . . . . .	548
14.11.4.608 Test0001 . . . . .	549
14.11.4.609 TestEventGenerate . . . . .	549
14.11.4.610 TestPattern . . . . .	549
14.11.4.611 TestPatternGeneratorSelector . . . . .	549
14.11.4.612 TestPendingAck . . . . .	549
14.11.4.613 TimerDelay . . . . .	550
14.11.4.614 TimerDuration . . . . .	550
14.11.4.615 TimerReset . . . . .	550
14.11.4.616 TimerSelector . . . . .	550
14.11.4.617 TimerStatus . . . . .	550
14.11.4.618 TimerTriggerActivation . . . . .	550
14.11.4.619 TimerTriggerSource . . . . .	551
14.11.4.620 TimerValue . . . . .	551
14.11.4.621 Timestamp . . . . .	551
14.11.4.622 TimestampLatch . . . . .	551
14.11.4.623 TimestampLatchValue . . . . .	551
14.11.4.624 TimestampReset . . . . .	551
14.11.4.625 TLParamsLocked . . . . .	552
14.11.4.626 TransferAbort . . . . .	552
14.11.4.627 TransferBlockCount . . . . .	552
14.11.4.628 TransferBurstCount . . . . .	552
14.11.4.629 TransferComponentSelector . . . . .	552
14.11.4.630 TransferControlMode . . . . .	552
14.11.4.631 TransferOperationMode . . . . .	553
14.11.4.632 TransferPause . . . . .	553
14.11.4.633 TransferQueueCurrentBlockCount . . . . .	553
14.11.4.634 TransferQueueMaxBlockCount . . . . .	553
14.11.4.635 TransferQueueMode . . . . .	553
14.11.4.636 TransferQueueOverflowCount . . . . .	553
14.11.4.637 TransferResume . . . . .	554
14.11.4.638 TransferSelector . . . . .	554
14.11.4.639 TransferStart . . . . .	554
14.11.4.640 TransferStatus . . . . .	554
14.11.4.641 TransferStatusSelector . . . . .	554
14.11.4.642 TransferStop . . . . .	554

---

---

14.11.4.643 TransferStreamChannel . . . . .	555
14.11.4.644 TransferTriggerActivation . . . . .	555
14.11.4.645 TransferTriggerMode . . . . .	555
14.11.4.646 TransferTriggerSelector . . . . .	555
14.11.4.647 TransferTriggerSource . . . . .	555
14.11.4.648 TriggerActivation . . . . .	555
14.11.4.649 TriggerDelay . . . . .	556
14.11.4.650 TriggerDivider . . . . .	556
14.11.4.651 TriggerEventTest . . . . .	556
14.11.4.652 TriggerMode . . . . .	556
14.11.4.653 TriggerMultiplier . . . . .	556
14.11.4.654 TriggerOverlap . . . . .	557
14.11.4.655 TriggerSelector . . . . .	557
14.11.4.656 TriggerSoftware . . . . .	557
14.11.4.657 TriggerSource . . . . .	557
14.11.4.658 UserOutputSelector . . . . .	557
14.11.4.659 UserOutputValue . . . . .	558
14.11.4.660 UserOutputValueAll . . . . .	558
14.11.4.661 UserOutputValueAllMask . . . . .	558
14.11.4.662 UserSetDefault . . . . .	558
14.11.4.663 UserSetFeatureEnable . . . . .	558
14.11.4.664 UserSetLoad . . . . .	559
14.11.4.665 UserSetSave . . . . .	559
14.11.4.666 UserSetSelector . . . . .	559
14.11.4.667 V3_3Enable . . . . .	559
14.11.4.668 WhiteClip . . . . .	560
14.11.4.669 WhiteClipSelector . . . . .	560
14.11.4.670 Width . . . . .	560
14.11.4.671 WidthMax . . . . .	560
<b>14.12 CameraBase Class Reference . . . . .</b>	<b>561</b>
<b>14.12.1 Detailed Description . . . . .</b>	<b>563</b>
<b>14.12.2 Constructor &amp; Destructor Documentation . . . . .</b>	<b>563</b>
<b>14.12.2.1 ~CameraBase() . . . . .</b>	<b>563</b>
<b>14.12.2.2 CameraBase() [1/2] . . . . .</b>	<b>564</b>
<b>14.12.2.3 CameraBase() [2/2] . . . . .</b>	<b>564</b>
<b>14.12.3 Member Function Documentation . . . . .</b>	<b>564</b>
<b>14.12.3.1 BeginAcquisition() . . . . .</b>	<b>564</b>
<b>14.12.3.2 DelInit() . . . . .</b>	<b>564</b>
<b>14.12.3.3 DiscoverMaxPacketSize() . . . . .</b>	<b>565</b>
<b>14.12.3.4 EndAcquisition() . . . . .</b>	<b>565</b>
<b>14.12.3.5 ForceIP() . . . . .</b>	<b>565</b>
<b>14.12.3.6 GetAccessMode() . . . . .</b>	<b>566</b>

---

---

14.12.3.7 GetBufferOwnership()	566
14.12.3.8 GetGuiXml()	566
14.12.3.9 GetNextImage()	567
14.12.3.10 GetNodeMap()	567
14.12.3.11 GetNumDataStreams()	568
14.12.3.12 GetNumImagesInUse()	568
14.12.3.13 GetTLDeviceNodeMap()	568
14.12.3.14 GetTLStreamNodeMap()	569
14.12.3.15 GetUniqueId()	569
14.12.3.16 GetUserBufferCount()	569
14.12.3.17 GetUserBufferSize()	570
14.12.3.18 GetUserBufferTotalSize()	570
14.12.3.19 Init()	571
14.12.3.20 IsInitialized()	571
14.12.3.21 IsStreaming()	571
14.12.3.22 IsValid()	572
14.12.3.23 operator=()	572
14.12.3.24 ReadPort()	572
14.12.3.25 RegisterEventHandler() [1/2]	572
14.12.3.26 RegisterEventHandler() [2/2]	573
14.12.3.27 SetBufferOwnership()	573
14.12.3.28 SetUserBuffers() [1/2]	574
14.12.3.29 SetUserBuffers() [2/2]	574
14.12.3.30 UnregisterEventHandler()	575
14.12.3.31 WritePort()	575
14.12.4 Friends And Related Function Documentation	575
14.12.4.1 InterfacImpl	576
14.13 CameraList Class Reference	576
14.13.1 Detailed Description	577
14.13.2 Constructor & Destructor Documentation	577
14.13.2.1 CameraList() [1/2]	577
14.13.2.2 ~CameraList()	578
14.13.2.3 CameraList() [2/2]	578
14.13.3 Member Function Documentation	578
14.13.3.1 Append()	578
14.13.3.2 Clear()	578
14.13.3.3 GetByDeviceID()	579
14.13.3.4 GetByIndex()	579
14.13.3.5 GetBySerial()	579
14.13.3.6 GetSize()	580
14.13.3.7 operator=()	580
14.13.3.8 operator[]()	580

---

---

14.13.3.9 RemoveByDeviceID()	580
14.13.3.10 RemoveByIndex()	581
14.13.3.11 RemoveBySerial()	581
14.14 CameraPtr Class Reference	581
14.14.1 Detailed Description	582
14.14.2 Constructor & Destructor Documentation	583
14.14.2.1 CameraPtr() [1/4]	583
14.14.2.2 CameraPtr() [2/4]	583
14.14.2.3 CameraPtr() [3/4]	583
14.14.2.4 CameraPtr() [4/4]	583
14.15 CategoryNode Class Reference	584
14.15.1 Detailed Description	585
14.15.2 Constructor & Destructor Documentation	585
14.15.2.1 CategoryNode() [1/2]	585
14.15.2.2 CategoryNode() [2/2]	585
14.15.2.3 ~CategoryNode()	585
14.15.3 Member Function Documentation	585
14.15.3.1 GetFeatures()	586
14.15.3.2 SetReference()	586
14.16 CChunkAdapter Class Reference	586
14.16.1 Detailed Description	587
14.16.2 Constructor & Destructor Documentation	587
14.16.2.1 ~CChunkAdapter()	587
14.16.2.2 CChunkAdapter()	587
14.16.3 Member Function Documentation	587
14.16.3.1 AttachBuffer()	587
14.16.3.2 AttachNodeMap()	588
14.16.3.3 CheckBufferLayout()	588
14.16.3.4 ClearCaches()	588
14.16.3.5 DetachBuffer()	588
14.16.3.6 DetachNodeMap()	588
14.16.3.7 UpdateBuffer()	588
14.16.4 Member Data Documentation	589
14.16.4.1 m_pChunkAdapter	589
14.17 CChunkAdapterDcam Class Reference	589
14.17.1 Detailed Description	590
14.17.2 Constructor & Destructor Documentation	590
14.17.2.1 CChunkAdapterDcam()	590
14.17.2.2 ~CChunkAdapterDcam()	590
14.17.3 Member Function Documentation	590
14.17.3.1 AttachBuffer()	591
14.17.3.2 CheckBufferLayout()	591

---

---

14.17.3.3 CheckCRC() . . . . .	591
14.17.3.4 HasCRC() . . . . .	591
14.18 CChunkAdapterGeneric Class Reference . . . . .	592
14.18.1 Constructor & Destructor Documentation . . . . .	592
14.18.1.1 CChunkAdapterGeneric() . . . . .	593
14.18.1.2 ~CChunkAdapterGeneric() . . . . .	593
14.18.2 Member Function Documentation . . . . .	593
14.18.2.1 AttachBuffer() [1/3] . . . . .	593
14.18.2.2 AttachBuffer() [2/3] . . . . .	593
14.18.2.3 AttachBuffer() [3/3] . . . . .	593
14.18.2.4 CheckBufferLayout() . . . . .	594
14.19 CChunkAdapterGEV Class Reference . . . . .	594
14.19.1 Detailed Description . . . . .	595
14.19.2 Constructor & Destructor Documentation . . . . .	595
14.19.2.1 CChunkAdapterGEV() . . . . .	595
14.19.2.2 ~CChunkAdapterGEV() . . . . .	595
14.19.3 Member Function Documentation . . . . .	595
14.19.3.1 AttachBuffer() . . . . .	596
14.19.3.2 CheckBufferLayout() . . . . .	596
14.20 CChunkAdapterU3V Class Reference . . . . .	596
14.20.1 Detailed Description . . . . .	597
14.20.2 Constructor & Destructor Documentation . . . . .	597
14.20.2.1 CChunkAdapterU3V() . . . . .	597
14.20.2.2 ~CChunkAdapterU3V() . . . . .	598
14.20.3 Member Function Documentation . . . . .	598
14.20.3.1 AttachBuffer() . . . . .	598
14.20.3.2 CheckBufferLayout() . . . . .	598
14.21 CChunkPort Class Reference . . . . .	599
14.21.1 Detailed Description . . . . .	600
14.21.2 Constructor & Destructor Documentation . . . . .	600
14.21.2.1 CChunkPort() . . . . .	600
14.21.2.2 ~CChunkPort() . . . . .	601
14.21.3 Member Function Documentation . . . . .	601
14.21.3.1 AttachChunk() . . . . .	601
14.21.3.2 AttachPort() . . . . .	601
14.21.3.3 CheckChunkID() [1/2] . . . . .	601
14.21.3.4 CheckChunkID() [2/2] . . . . .	601
14.21.3.5 ClearCache() . . . . .	602
14.21.3.6 DetachChunk() . . . . .	602
14.21.3.7 DetachPort() . . . . .	602
14.21.3.8 GetAccessMode() . . . . .	602
14.21.3.9 GetChunkIDLength() . . . . .	602

---

---

14.21.3.10 GetPrincipalInterfaceType()	602
14.21.3.11 GetSwapEndianess()	603
14.21.3.12 InvalidateNode()	603
14.21.3.13 Read()	603
14.21.3.14 SetPortImpl()	603
14.21.3.15 UpdateBuffer()	603
14.21.3.16 Write()	603
14.21.4 Member Data Documentation	604
14.21.4.1 m_pChunkPort	604
14.21.4.2 m_pPort	604
14.21.4.3 m_pPortAdapter	604
14.22 CEnumerationTRef< EnumT > Class Template Reference	604
14.22.1 Detailed Description	606
14.22.2 Constructor & Destructor Documentation	606
14.22.2.1 CEnumerationTRef() [1/2]	606
14.22.2.2 CEnumerationTRef() [2/2]	606
14.22.2.3 ~CEnumerationTRef()	606
14.22.3 Member Function Documentation	606
14.22.3.1 GetCurrentEntry()	607
14.22.3.2 GetEntry() [1/2]	607
14.22.3.3 GetEntry() [2/2]	607
14.22.3.4 GetValue()	607
14.22.3.5 operator()	608
14.22.3.6 operator=() [1/2]	608
14.22.3.7 operator=() [2/2]	608
14.22.3.8 SetEnumReference()	608
14.22.3.9 SetNumEnums()	608
14.22.3.10 SetReference()	609
14.22.3.11 SetValue()	609
14.23 CEventAdapter Class Reference	609
14.23.1 Detailed Description	610
14.23.2 Constructor & Destructor Documentation	610
14.23.2.1 CEventAdapter()	610
14.23.2.2 ~CEventAdapter()	610
14.23.3 Member Function Documentation	610
14.23.3.1 AttachNodeMap()	610
14.23.3.2 DeliverMessage()	611
14.23.3.3 DetachNodeMap()	611
14.23.4 Member Data Documentation	611
14.23.4.1 m_pEventAdapter	611
14.24 CEventAdapter1394 Class Reference	611
14.24.1 Detailed Description	612

---

---

14.24.2 Constructor & Destructor Documentation . . . . .	612
14.24.2.1 CEventAdapter1394() . . . . .	612
14.24.2.2 ~CEventAdapter1394() . . . . .	612
14.24.3 Member Function Documentation . . . . .	613
14.24.3.1 DeliverEventMessage() . . . . .	613
14.24.3.2 DeliverMessage() . . . . .	613
14.25 CEventAdapterGeneric Class Reference . . . . .	613
14.25.1 Detailed Description . . . . .	614
14.25.2 Constructor & Destructor Documentation . . . . .	614
14.25.2.1 CEventAdapterGeneric() . . . . .	614
14.25.2.2 ~CEventAdapterGeneric() . . . . .	615
14.25.3 Member Function Documentation . . . . .	615
14.25.3.1 DeliverMessage() [1/3] . . . . .	615
14.25.3.2 DeliverMessage() [2/3] . . . . .	615
14.25.3.3 DeliverMessage() [3/3] . . . . .	615
14.26 CEventAdapterGEV Class Reference . . . . .	616
14.26.1 Detailed Description . . . . .	617
14.26.2 Constructor & Destructor Documentation . . . . .	617
14.26.2.1 CEventAdapterGEV() . . . . .	617
14.26.2.2 ~CEventAdapterGEV() . . . . .	617
14.26.3 Member Function Documentation . . . . .	617
14.26.3.1 DeliverEventMessage() [1/2] . . . . .	617
14.26.3.2 DeliverEventMessage() [2/2] . . . . .	617
14.26.3.3 DeliverMessage() . . . . .	618
14.27 CEventAdapterU3V Class Reference . . . . .	618
14.27.1 Detailed Description . . . . .	619
14.27.2 Constructor & Destructor Documentation . . . . .	619
14.27.2.1 CEventAdapterU3V() . . . . .	619
14.27.2.2 ~CEventAdapterU3V() . . . . .	619
14.27.3 Member Function Documentation . . . . .	619
14.27.3.1 DeliverEventMessage() . . . . .	619
14.27.3.2 DeliverMessage() . . . . .	620
14.28 CEventPort Class Reference . . . . .	620
14.28.1 Detailed Description . . . . .	621
14.28.2 Constructor & Destructor Documentation . . . . .	621
14.28.2.1 CEventPort() . . . . .	622
14.28.2.2 ~CEventPort() . . . . .	622
14.28.3 Member Function Documentation . . . . .	622
14.28.3.1 AttachEvent() . . . . .	622
14.28.3.2 AttachNode() . . . . .	622
14.28.3.3 CheckEventID() [1/2] . . . . .	622
14.28.3.4 CheckEventID() [2/2] . . . . .	623

---

---

14.28.3.5 DetachEvent()	623
14.28.3.6 DetachNode()	623
14.28.3.7 GetAccessMode()	623
14.28.3.8 GetEventIDLength()	623
14.28.3.9 GetPrincipalInterfaceType()	623
14.28.3.10 GetSwapEndianess()	624
14.28.3.11 InvalidateNode()	624
14.28.3.12 Read()	624
14.28.3.13 SetPortImpl()	624
14.28.3.14 Write()	624
14.28.4 Member Data Documentation	624
14.28.4.1 m_pEventPort	625
14.28.4.2 m_pNode	625
14.28.4.3 m_pPortAdapter	625
14.29 CFeatureBag Class Reference	625
14.29.1 Detailed Description	626
14.29.2 Constructor & Destructor Documentation	626
14.29.2.1 CFeatureBag()	626
14.29.2.2 ~CFeatureBag()	626
14.29.3 Member Function Documentation	626
14.29.3.1 GetFeatureBagHandle()	626
14.29.3.2 LoadFromBag()	626
14.29.3.3 operator==()	627
14.29.3.4 PersistFeature()	627
14.29.3.5 SetInfo()	627
14.29.3.6 StoreToBag()	627
14.30 CFloatPtr Class Reference	628
14.30.1 Detailed Description	629
14.30.2 Constructor & Destructor Documentation	629
14.30.2.1 CFloatPtr() [1/2]	629
14.30.2.2 CFloatPtr() [2/2]	629
14.30.3 Member Function Documentation	629
14.30.3.1 GetEnumAlias()	629
14.30.3.2 GetIntAlias()	629
14.30.3.3 operator=()	630
14.31 CGeneric_XMLLoaderParams Class Reference	630
14.31.1 Detailed Description	630
14.31.2 Member Function Documentation	630
14.31.2.1 _Initialize()	631
14.32 CGlobalLock Class Reference	631
14.32.1 Detailed Description	631
14.32.2 Constructor & Destructor Documentation	631

---

---

14.32.2.1 <code>CGlobalLock()</code> [1/2] . . . . .	632
14.32.2.2 <code>CGlobalLock()</code> [2/2] . . . . .	632
14.32.2.3 <code>~CGlobalLock()</code> . . . . .	632
14.32.3 Member Function Documentation . . . . .	632
14.32.3.1 <code>IsValid()</code> . . . . .	632
14.32.3.2 <code>Lock()</code> . . . . .	632
14.32.3.3 <code>TryLock()</code> . . . . .	633
14.32.3.4 <code>Unlock()</code> . . . . .	633
14.32.4 Member Data Documentation . . . . .	633
14.32.4.1 <code>m_DebugCount</code> . . . . .	633
14.33 <code>CGlobalLockUnlocker</code> Class Reference . . . . .	633
14.33.1 Detailed Description . . . . .	634
14.33.2 Constructor & Destructor Documentation . . . . .	634
14.33.2.1 <code>CGlobalLockUnlocker()</code> . . . . .	634
14.33.2.2 <code>~CGlobalLockUnlocker()</code> . . . . .	634
14.33.3 Member Function Documentation . . . . .	634
14.33.3.1 <code>UnlockEarly()</code> . . . . .	634
14.33.4 Member Data Documentation . . . . .	635
14.33.4.1 <code>m_enabled</code> . . . . .	635
14.33.4.2 <code>m_Lock</code> . . . . .	635
14.34 <code>ChunkData</code> Class Reference . . . . .	635
14.34.1 Detailed Description . . . . .	637
14.34.2 Constructor & Destructor Documentation . . . . .	637
14.34.2.1 <code>ChunkData()</code> [1/2] . . . . .	637
14.34.2.2 <code>ChunkData()</code> [2/2] . . . . .	638
14.34.2.3 <code>~ChunkData()</code> . . . . .	638
14.34.3 Member Function Documentation . . . . .	638
14.34.3.1 <code>GetBlackLevel()</code> . . . . .	638
14.34.3.2 <code>GetCounterValue()</code> . . . . .	638
14.34.3.3 <code>GetCRC()</code> . . . . .	638
14.34.3.4 <code>GetEncoderValue()</code> . . . . .	639
14.34.3.5 <code>GetExposureEndLineStatusAll()</code> . . . . .	639
14.34.3.6 <code>GetExposureTime()</code> . . . . .	639
14.34.3.7 <code>GetFrameID()</code> . . . . .	639
14.34.3.8 <code>GetGain()</code> . . . . .	640
14.34.3.9 <code>GetHeight()</code> . . . . .	640
14.34.3.10 <code>GetImage()</code> . . . . .	640
14.34.3.11 <code>GetInferenceBoundingBoxResult()</code> . . . . .	640
14.34.3.12 <code>GetInferenceConfidence()</code> . . . . .	641
14.34.3.13 <code>GetInferenceFrameId()</code> . . . . .	641
14.34.3.14 <code>GetInferenceResult()</code> . . . . .	641
14.34.3.15 <code>GetLinePitch()</code> . . . . .	641

---

---

14.34.3.16 GetLineStatusAll()	642
14.34.3.17 GetOffsetX()	642
14.34.3.18 GetOffsetY()	642
14.34.3.19 GetPartSelector()	642
14.34.3.20 GetPixelDynamicRangeMax()	643
14.34.3.21 GetPixelDynamicRangeMin()	643
14.34.3.22 GetScan3dAxisMax()	643
14.34.3.23 GetScan3dAxisMin()	643
14.34.3.24 GetScan3dCoordinateOffset()	644
14.34.3.25 GetScan3dCoordinateReferenceValue()	644
14.34.3.26 GetScan3dCoordinateScale()	644
14.34.3.27 GetScan3dInvalidDataValue()	644
14.34.3.28 GetScan3dTransformValue()	645
14.34.3.29 GetScanLineSelector()	645
14.34.3.30 GetSequencerSetActive()	645
14.34.3.31 GetSerialDataLength()	645
14.34.3.32 GetStreamChannelID()	646
14.34.3.33 GetTimerValue()	646
14.34.3.34 GetTimestamp()	646
14.34.3.35 GetTimestampLatchValue()	646
14.34.3.36 GetTransferBlockID()	647
14.34.3.37 GetTransferQueueCurrentBlockCount()	647
14.34.3.38 GetWidth()	647
14.34.3.39 SetChunks()	647
14.35 CLock Class Reference	648
14.35.1 Detailed Description	648
14.35.2 Constructor & Destructor Documentation	648
14.35.2.1 CLock()	648
14.35.2.2 ~CLock()	649
14.35.3 Member Function Documentation	649
14.35.3.1 Lock()	649
14.35.3.2 TryLock()	649
14.35.3.3 Unlock()	649
14.36 CLock Class Reference	649
14.36.1 Detailed Description	650
14.36.2 Constructor & Destructor Documentation	650
14.36.2.1 CLock() [1/2]	650
14.36.2.2 CLock() [2/2]	650
14.36.2.3 ~CLock()	651
14.36.3 Member Function Documentation	651
14.36.3.1 Lock()	651
14.36.3.2 TryLock()	651

---

---

14.36.3.3 <code>Unlock()</code>	651
14.36.4 Friends And Related Function Documentation	651
14.36.4.1 <code>NodeMap</code>	651
14.36.5 Member Data Documentation	651
14.36.5.1 <code>m_bOwnLock</code>	652
14.36.5.2 <code>m_lock</code>	652
14.37 <code>CLockEx</code> Class Reference	652
14.37.1 Detailed Description	653
14.38 <code>CLockEx</code> Class Reference	653
14.38.1 Detailed Description	654
14.38.2 Member Data Documentation	654
14.38.2.1 <code>m_lockEx</code>	654
14.39 <code>CNodeCallback</code> Class Reference	654
14.39.1 Detailed Description	655
14.39.2 Constructor & Destructor Documentation	655
14.39.2.1 <code>CNodeCallback()</code>	655
14.39.2.2 <code>~CNodeCallback()</code>	655
14.39.3 Member Function Documentation	655
14.39.3.1 <code>Destroy()</code>	656
14.39.3.2 <code>GetCallbackType()</code>	656
14.39.3.3 <code>GetNode()</code>	656
14.39.3.4 <code>operator()()</code>	656
14.39.4 Member Data Documentation	656
14.39.4.1 <code>m_CallbackType</code>	656
14.39.4.2 <code>m_pNode</code>	657
14.40 <code>CNodeMapFactory</code> Class Reference	657
14.40.1 Detailed Description	658
14.40.2 Constructor & Destructor Documentation	659
14.40.2.1 <code>CNodeMapFactory()</code> [1/5]	659
14.40.2.2 <code>~CNodeMapFactory()</code>	659
14.40.2.3 <code>CNodeMapFactory()</code> [2/5]	659
14.40.2.4 <code>CNodeMapFactory()</code> [3/5]	659
14.40.2.5 <code>CNodeMapFactory()</code> [4/5]	660
14.40.2.6 <code>CNodeMapFactory()</code> [5/5]	661
14.40.3 Member Function Documentation	661
14.40.3.1 <code>AddInjectionData()</code>	661
14.40.3.2 <code>ApplyStyleSheet()</code>	661
14.40.3.3 <code>ClearCache()</code>	662
14.40.3.4 <code>CreateEmptyNodeMap()</code>	662
14.40.3.5 <code>CreateNodeDataFromNodeMap()</code>	662
14.40.3.6 <code>CreateNodeMap()</code> [1/2]	662
14.40.3.7 <code>CreateNodeMap()</code> [2/2]	662

---

---

14.40.3.8 ExtractSubtree()	663
14.40.3.9 GetNodeStatistics()	663
14.40.3.10 GetSupportedSchemaVersions()	663
14.40.3.11 IsCameraDescriptionFileDataReleased()	663
14.40.3.12 IsEmpty()	663
14.40.3.13 IsLoaded()	664
14.40.3.14 IsPreprocessed()	664
14.40.3.15 LoadAndInject()	664
14.40.3.16 operator=()	664
14.40.3.17 Preprocess()	664
14.40.3.18 ReleaseCameraDescriptionFileData()	665
14.40.3.19 ToString()	665
14.40.3.20 ToXml()	665
14.41 CNodeMapRef Class Reference	665
14.41.1 Detailed Description	666
14.41.2 Constructor & Destructor Documentation	666
14.41.2.1 CNodeMapRef() [1/3]	667
14.41.2.2 CNodeMapRef() [2/3]	667
14.41.2.3 CNodeMapRef() [3/3]	667
14.41.3 Member Function Documentation	667
14.41.3.1 operator=() [1/2]	667
14.41.3.2 operator=() [2/2]	667
14.42 CNodeMapRefT< TCameraParams > Class Template Reference	668
14.42.1 Detailed Description	669
14.42.2 Constructor & Destructor Documentation	670
14.42.2.1 CNodeMapRefT() [1/3]	670
14.42.2.2 CNodeMapRefT() [2/3]	670
14.42.2.3 CNodeMapRefT() [3/3]	670
14.42.2.4 ~CNodeMapRefT()	670
14.42.3 Member Function Documentation	670
14.42.3.1 _ClearXMLCache()	671
14.42.3.2 _Connect() [1/2]	671
14.42.3.3 _Connect() [2/2]	671
14.42.3.4 _Destroy()	671
14.42.3.5 _GetDeviceName()	671
14.42.3.6 _GetNode()	671
14.42.3.7 _GetNodes()	672
14.42.3.8 _GetSupportedSchemaVersions()	672
14.42.3.9 _InvalidateNodes()	672
14.42.3.10 _LoadXMLFromFile()	672
14.42.3.11 _LoadXMLFromFileInject()	672
14.42.3.12 _LoadXMLFromString()	673

14.42.3.13 _LoadXMLFromStringInject()	673
14.42.3.14 _LoadXMLFromZIPData()	673
14.42.3.15 _LoadXMLFromZIPFile()	673
14.42.3.16 _Poll()	673
14.42.3.17 operator=() [1/2]	674
14.42.3.18 operator=() [2/2]	674
14.42.4 Member Data Documentation	674
14.42.4.1 _Ptr	674
14.43 CommandNode Class Reference	674
14.43.1 Detailed Description	675
14.43.2 Constructor & Destructor Documentation	675
14.43.2.1 CommandNode() [1/2]	676
14.43.2.2 CommandNode() [2/2]	676
14.43.2.3 ~CommandNode()	676
14.43.3 Member Function Documentation	676
14.43.3.1 Execute()	676
14.43.3.2 IsDone()	676
14.43.3.3 operator()()	677
14.43.3.4 SetReference()	677
14.44 Counter Class Reference	677
14.44.1 Detailed Description	677
14.44.2 Constructor & Destructor Documentation	678
14.44.2.1 Counter()	678
14.44.3 Member Function Documentation	678
14.44.3.1 GetValue()	678
14.44.3.2 IsZero()	678
14.44.3.3 operator unsigned int()	678
14.44.3.4 operator++() [1/2]	678
14.44.3.5 operator++() [2/2]	678
14.44.3.6 operator--() [1/2]	679
14.44.3.7 operator--() [2/2]	679
14.45 CPointer< T, B > Class Template Reference	679
14.45.1 Detailed Description	680
14.45.2 Constructor & Destructor Documentation	680
14.45.2.1 CPointer() [1/2]	680
14.45.2.2 CPointer() [2/2]	680
14.45.2.3 ~CPointer()	680
14.45.3 Member Function Documentation	681
14.45.3.1 IsValid()	681
14.45.3.2 operator bool()	681
14.45.3.3 operator T*()	681
14.45.3.4 operator"!=() [1/5]	681

14.45.3.5 operator"!=() [2/5] . . . . .	681
14.45.3.6 operator"!=() [3/5] . . . . .	682
14.45.3.7 operator"!=() [4/5] . . . . .	682
14.45.3.8 operator"!=() [5/5] . . . . .	682
14.45.3.9 operator()() . . . . .	682
14.45.3.10 operator*() . . . . .	682
14.45.3.11 operator->() . . . . .	682
14.45.3.12 operator=() . . . . .	683
14.45.3.13 operator==() [1/3] . . . . .	683
14.45.3.14 operator==() [2/3] . . . . .	683
14.45.3.15 operator==() [3/3] . . . . .	683
14.45.4 Member Data Documentation . . . . .	683
14.45.4.1 m_pT . . . . .	683
14.46 CPortImpl Class Reference . . . . .	684
14.46.1 Detailed Description . . . . .	685
14.46.2 Constructor & Destructor Documentation . . . . .	685
14.46.2.1 CPortImpl() . . . . .	685
14.46.2.2 ~CPortImpl() . . . . .	685
14.46.3 Member Function Documentation . . . . .	685
14.46.3.1 GetAccessMode() . . . . .	685
14.46.3.2 GetSwapEndianess() . . . . .	686
14.46.3.3 InvalidateNode() . . . . .	686
14.46.3.4 Read() . . . . .	686
14.46.3.5 Replay() . . . . .	686
14.46.3.6 SetPortImpl() . . . . .	686
14.46.3.7 Write() . . . . .	687
14.46.4 Member Data Documentation . . . . .	687
14.46.4.1 m_ptrPort . . . . .	687
14.47 CPortWriteList Class Reference . . . . .	687
14.47.1 Detailed Description . . . . .	688
14.47.2 Constructor & Destructor Documentation . . . . .	688
14.47.2.1 CPortWriteList() . . . . .	688
14.47.2.2 ~CPortWriteList() . . . . .	689
14.47.3 Member Function Documentation . . . . .	689
14.47.3.1 GetCookie() . . . . .	689
14.47.3.2 GetPortWriteListHandle() . . . . .	689
14.47.3.3 Replay() . . . . .	689
14.47.3.4 SetCookie() . . . . .	689
14.47.3.5 Write() . . . . .	689
14.47.4 Member Data Documentation . . . . .	690
14.47.4.1 m_pWriteList . . . . .	690
14.48 CpuUsageInfo Struct Reference . . . . .	690

---

14.48.1 Member Data Documentation . . . . .	690
14.48.1.1 dummy . . . . .	690
14.49 CRegisterPortImpl Class Reference . . . . .	691
14.49.1 Detailed Description . . . . .	692
14.49.2 Constructor & Destructor Documentation . . . . .	692
14.49.2.1 CRegisterPortImpl() . . . . .	692
14.49.2.2 ~CRegisterPortImpl() . . . . .	692
14.49.3 Member Function Documentation . . . . .	692
14.49.3.1 GetAccessMode() . . . . .	693
14.49.3.2 Read() . . . . .	693
14.49.3.3 ReadRegister() . . . . .	693
14.49.3.4 SetPortImpl() . . . . .	693
14.49.3.5 Write() . . . . .	694
14.49.3.6 WriteRegister() . . . . .	694
14.50 CSelectorSet Class Reference . . . . .	694
14.50.1 Detailed Description . . . . .	695
14.50.2 Constructor & Destructor Documentation . . . . .	695
14.50.2.1 CSelectorSet() . . . . .	695
14.50.2.2 ~CSelectorSet() . . . . .	696
14.50.3 Member Function Documentation . . . . .	696
14.50.3.1 GetSelectorList() . . . . .	696
14.50.3.2 IsEmpty() . . . . .	696
14.50.3.3 Restore() . . . . .	696
14.50.3.4 SetFirst() . . . . .	696
14.50.3.5 SetNext() . . . . .	697
14.50.3.6 ToString() . . . . .	697
14.51 CTestPortStruct< CDataStruct > Class Template Reference . . . . .	697
14.51.1 Detailed Description . . . . .	698
14.51.2 Constructor & Destructor Documentation . . . . .	699
14.51.2.1 CTestPortStruct() . . . . .	699
14.51.3 Member Function Documentation . . . . .	699
14.51.3.1 GetAccessMode() . . . . .	699
14.51.3.2 GetNumReads() . . . . .	699
14.51.3.3 GetNumWrites() . . . . .	699
14.51.3.4 GetPrincipalInterfaceType() . . . . .	699
14.51.3.5 MemSet() . . . . .	700
14.51.3.6 Read() . . . . .	700
14.51.3.7 ResetStatistics() . . . . .	700
14.51.3.8 Write() . . . . .	700
14.51.4 Member Data Documentation . . . . .	700
14.51.4.1 m_BaseAddress . . . . .	700
14.51.4.2 m_NumReads . . . . .	701

---

---

14.51.4.3 m_NumWrites . . . . .	701
14.52 DCAM_CHECKSUM Struct Reference . . . . .	701
14.52.1 Member Data Documentation . . . . .	701
14.52.1.1 CRCChecksum . . . . .	701
14.53 DCAM_CHUNK_TRAILER Struct Reference . . . . .	701
14.53.1 Member Data Documentation . . . . .	702
14.53.1.1 ChunkID . . . . .	702
14.53.1.2 ChunkLength . . . . .	702
14.53.1.3 InverseChunkLength . . . . .	702
14.54 DeviceArrivalEventHandler Class Reference . . . . .	702
14.54.1 Detailed Description . . . . .	703
14.54.2 Constructor & Destructor Documentation . . . . .	703
14.54.2.1 DeviceArrivalEventHandler() . . . . .	704
14.54.2.2 ~DeviceArrivalEventHandler() . . . . .	704
14.54.3 Member Function Documentation . . . . .	704
14.54.3.1 OnDeviceArrival() . . . . .	704
14.54.3.2 operator=( ) . . . . .	704
14.55 DeviceEventHandler Class Reference . . . . .	705
14.55.1 Detailed Description . . . . .	706
14.55.2 Constructor & Destructor Documentation . . . . .	706
14.55.2.1 DeviceEventHandler() . . . . .	706
14.55.2.2 ~DeviceEventHandler() . . . . .	706
14.55.3 Member Function Documentation . . . . .	706
14.55.3.1 GetDeviceEventId() . . . . .	707
14.55.3.2 GetDeviceEventName() . . . . .	707
14.55.3.3 OnDeviceEvent() . . . . .	707
14.55.3.4 operator=( ) . . . . .	707
14.56 DeviceEventHandlerImpl Class Reference . . . . .	708
14.56.1 Constructor & Destructor Documentation . . . . .	709
14.56.1.1 DeviceEventHandlerImpl() . . . . .	709
14.56.1.2 ~DeviceEventHandlerImpl() . . . . .	709
14.56.2 Member Function Documentation . . . . .	710
14.56.2.1 OnDeviceEvent() . . . . .	710
14.57 DeviceRemovalEventHandler Class Reference . . . . .	710
14.57.1 Detailed Description . . . . .	711
14.57.2 Constructor & Destructor Documentation . . . . .	711
14.57.2.1 DeviceRemovalEventHandler() . . . . .	712
14.57.2.2 ~DeviceRemovalEventHandler() . . . . .	712
14.57.3 Member Function Documentation . . . . .	712
14.57.3.1 OnDeviceRemoval() . . . . .	712
14.57.3.2 operator=( ) . . . . .	712
14.58 double_autovector_t Class Reference . . . . .	712

---

---

14.58.1 Detailed Description . . . . .	713
14.58.2 Constructor & Destructor Documentation . . . . .	713
14.58.2.1 double <sub>_</sub> autovector <sub>_</sub> t() [1/3] . . . . .	713
14.58.2.2 double <sub>_</sub> autovector <sub>_</sub> t() [2/3] . . . . .	713
14.58.2.3 double <sub>_</sub> autovector <sub>_</sub> t() [3/3] . . . . .	713
14.58.2.4 ~double <sub>_</sub> autovector <sub>_</sub> t() . . . . .	714
14.58.3 Member Function Documentation . . . . .	714
14.58.3.1 operator delete() . . . . .	714
14.58.3.2 operator new() . . . . .	714
14.58.3.3 operator=() . . . . .	714
14.58.3.4 operator[]() [1/2] . . . . .	714
14.58.3.5 operator[]() [2/2] . . . . .	714
14.58.3.6 size() . . . . .	715
14.58.4 Member Data Documentation . . . . .	715
14.58.4.1 _pCount . . . . .	715
14.58.4.2 _pv . . . . .	715
14.59 EAccessModeClass Class Reference . . . . .	715
14.59.1 Detailed Description . . . . .	715
14.59.2 Member Function Documentation . . . . .	715
14.59.2.1 FromString() . . . . .	716
14.59.2.2 ToString() [1/2] . . . . .	716
14.59.2.3 ToString() [2/2] . . . . .	716
14.60 ECachingModeClass Class Reference . . . . .	716
14.60.1 Detailed Description . . . . .	716
14.60.2 Member Function Documentation . . . . .	717
14.60.2.1 FromString() . . . . .	717
14.60.2.2 ToString() [1/2] . . . . .	717
14.60.2.3 ToString() [2/2] . . . . .	717
14.61 EDisplayNotationClass Class Reference . . . . .	717
14.61.1 Detailed Description . . . . .	718
14.61.2 Member Function Documentation . . . . .	718
14.61.2.1 FromString() . . . . .	718
14.61.2.2 ToString() [1/2] . . . . .	718
14.61.2.3 ToString() [2/2] . . . . .	718
14.62 EEndianessClass Class Reference . . . . .	718
14.62.1 Detailed Description . . . . .	719
14.62.2 Member Function Documentation . . . . .	719
14.62.2.1 FromString() . . . . .	719
14.62.2.2 ToString() [1/2] . . . . .	719
14.62.2.3 ToString() [2/2] . . . . .	719
14.63 EGenApiSchemaVersionClass Class Reference . . . . .	720
14.63.1 Detailed Description . . . . .	720

---

---

14.63.2 Member Function Documentation . . . . .	720
14.63.2.1 FromString() . . . . .	720
14.63.2.2 ToString() [1/2] . . . . .	720
14.63.2.3 ToString() [2/2] . . . . .	720
14.64 EInputDirectionClass Class Reference . . . . .	721
14.64.1 Detailed Description . . . . .	721
14.64.2 Member Function Documentation . . . . .	721
14.64.2.1 FromString() . . . . .	721
14.64.2.2 ToString() [1/2] . . . . .	721
14.64.2.3 ToString() [2/2] . . . . .	721
14.65 ENameSpaceClass Class Reference . . . . .	722
14.65.1 Detailed Description . . . . .	722
14.65.2 Member Function Documentation . . . . .	722
14.65.2.1 FromString() . . . . .	722
14.65.2.2 ToString() [1/2] . . . . .	722
14.65.2.3 ToString() [2/2] . . . . .	722
14.66 EnumEntryNode Class Reference . . . . .	723
14.66.1 Detailed Description . . . . .	724
14.66.2 Constructor & Destructor Documentation . . . . .	724
14.66.2.1 EnumEntryNode() [1/2] . . . . .	724
14.66.2.2 EnumEntryNode() [2/2] . . . . .	724
14.66.2.3 ~EnumEntryNode() . . . . .	724
14.66.3 Member Function Documentation . . . . .	724
14.66.3.1 GetNumericValue() . . . . .	725
14.66.3.2 GetSymbolic() . . . . .	725
14.66.3.3 GetValue() . . . . .	725
14.66.3.4 IsSelfClearing() . . . . .	725
14.66.3.5 SetReference() . . . . .	725
14.67 EnumNode Class Reference . . . . .	726
14.67.1 Detailed Description . . . . .	728
14.67.2 Constructor & Destructor Documentation . . . . .	728
14.67.2.1 EnumNode() [1/2] . . . . .	728
14.67.2.2 EnumNode() [2/2] . . . . .	728
14.67.2.3 ~EnumNode() . . . . .	728
14.67.3 Member Function Documentation . . . . .	728
14.67.3.1 GetCurrentEntry() . . . . .	728
14.67.3.2 GetEntries() . . . . .	729
14.67.3.3 GetEntry() . . . . .	729
14.67.3.4 GetEntryByName() . . . . .	729
14.67.3.5 GetIntValue() . . . . .	729
14.67.3.6 GetSymbolics() . . . . .	730
14.67.3.7 operator*() . . . . .	730

---

---

14.67.3.8 operator=()	730
14.67.3.9 SetIntValue()	730
14.67.3.10 SetReference()	730
14.67.4 Member Data Documentation	731
14.67.4.1 m_pEnumeration	731
14.68 ERepresentationClass Class Reference	731
14.68.1 Detailed Description	731
14.68.2 Member Function Documentation	731
14.68.2.1 FromString()	732
14.68.2.2 ToString() [1/2]	732
14.68.2.3 ToString() [2/2]	732
14.69 ESignClass Class Reference	732
14.69.1 Detailed Description	732
14.69.2 Member Function Documentation	733
14.69.2.1 FromString()	733
14.69.2.2 ToString() [1/2]	733
14.69.2.3 ToString() [2/2]	733
14.70 ESlopeClass Class Reference	733
14.70.1 Detailed Description	734
14.70.2 Member Function Documentation	734
14.70.2.1 FromString()	734
14.70.2.2 ToString() [1/2]	734
14.70.2.3 ToString() [2/2]	734
14.71 EStandardNameSpaceClass Class Reference	734
14.71.1 Detailed Description	735
14.71.2 Member Function Documentation	735
14.71.2.1 FromString()	735
14.71.2.2 ToString() [1/2]	735
14.71.2.3 ToString() [2/2]	735
14.72 EventHandler Class Reference	736
14.72.1 Detailed Description	737
14.72.2 Constructor & Destructor Documentation	737
14.72.2.1 ~EventHandler()	737
14.72.2.2 EventHandler()	737
14.72.3 Member Function Documentation	737
14.72.3.1 GetEventPayloadData()	737
14.72.3.2 GetEventPayloadDataSize()	737
14.72.3.3 GetEventType()	738
14.72.3.4 operator=()	738
14.72.3.5 SetEventPayload()	738
14.72.3.6 SetEventType()	738
14.72.4 Friends And Related Function Documentation	738

---

14.72.4.1 EventProcessor . . . . .	738
14.72.4.2 IDataStream . . . . .	739
14.72.4.3 Stream . . . . .	739
14.72.5 Member Data Documentation . . . . .	739
14.72.5.1 m_pEventData . . . . .	739
14.73 EVisibilityClass Class Reference . . . . .	739
14.73.1 Detailed Description . . . . .	739
14.73.2 Member Function Documentation . . . . .	739
14.73.2.1 FromString() . . . . .	740
14.73.2.2 ToString() [1/2] . . . . .	740
14.73.2.3 ToString() [2/2] . . . . .	740
14.74 Exception Class Reference . . . . .	740
14.74.1 Detailed Description . . . . .	741
14.74.2 Constructor & Destructor Documentation . . . . .	742
14.74.2.1 Exception() [1/4] . . . . .	742
14.74.2.2 Exception() [2/4] . . . . .	742
14.74.2.3 Exception() [3/4] . . . . .	742
14.74.2.4 Exception() [4/4] . . . . .	743
14.74.2.5 ~Exception() . . . . .	743
14.74.3 Member Function Documentation . . . . .	743
14.74.3.1 GetBuildDate() . . . . .	743
14.74.3.2 GetBuildTime() . . . . .	743
14.74.3.3 GetError() . . . . .	743
14.74.3.4 GetErrorMessage() . . . . .	744
14.74.3.5 GetFileName() . . . . .	744
14.74.3.6 GetFullErrorMessage() . . . . .	744
14.74.3.7 GetFunctionName() . . . . .	744
14.74.3.8 GetLineNumber() . . . . .	744
14.74.3.9 operator"!="() . . . . .	744
14.74.3.10 operator=() . . . . .	745
14.74.3.11 operator==() . . . . .	745
14.74.3.12 what() . . . . .	745
14.75 EYesNoClass Class Reference . . . . .	745
14.75.1 Detailed Description . . . . .	745
14.75.2 Member Function Documentation . . . . .	746
14.75.2.1 FromString() . . . . .	746
14.75.2.2 ToString() [1/2] . . . . .	746
14.75.2.3 ToString() [2/2] . . . . .	746
14.76 FileProtocolAdapter Class Reference . . . . .	746
14.76.1 Detailed Description . . . . .	747
14.76.2 Constructor & Destructor Documentation . . . . .	747
14.76.2.1 FileProtocolAdapter() . . . . .	747

---

---

14.76.2.2 ~FileProtocolAdapter()	747
14.76.3 Member Function Documentation	747
14.76.3.1 attach()	747
14.76.3.2 closeFile()	748
14.76.3.3 deleteFile()	748
14.76.3.4 getBufSize()	748
14.76.3.5 openFile()	749
14.76.3.6 read()	749
14.76.3.7 write()	750
14.77 FloatNode Class Reference	750
14.77.1 Detailed Description	752
14.77.2 Constructor & Destructor Documentation	753
14.77.2.1 FloatNode() [1/2]	753
14.77.2.2 FloatNode() [2/2]	753
14.77.2.3 ~FloatNode()	753
14.77.3 Member Function Documentation	753
14.77.3.1 GetDisplayNotation()	753
14.77.3.2 GetDisplayPrecision()	753
14.77.3.3 GetEnumAlias()	753
14.77.3.4 GetInc()	754
14.77.3.5 GetIncMode()	754
14.77.3.6 GetIntAlias()	754
14.77.3.7 GetListOfValidValues()	754
14.77.3.8 GetMax()	754
14.77.3.9 GetMin()	754
14.77.3.10 GetRepresentation()	755
14.77.3.11 GetUnit()	755
14.77.3.12 GetValue()	755
14.77.3.13 HasInc()	755
14.77.3.14 ImposeMax()	755
14.77.3.15 ImposeMin()	756
14.77.3.16 operator()()	756
14.77.3.17 operator*()	756
14.77.3.18 operator=()	756
14.77.3.19 SetReference()	756
14.77.3.20 SetValue()	756
14.78 FloatRegNode Class Reference	757
14.78.1 Detailed Description	758
14.78.2 Constructor & Destructor Documentation	758
14.78.2.1 FloatRegNode() [1/2]	759
14.78.2.2 FloatRegNode() [2/2]	759
14.78.2.3 ~FloatRegNode()	759

---

---

14.78.3 Member Function Documentation . . . . .	759
14.78.3.1 SetReference() . . . . .	759
14.79 Function_NodeCallback< Function > Class Template Reference . . . . .	760
14.79.1 Detailed Description . . . . .	760
14.79.2 Constructor & Destructor Documentation . . . . .	761
14.79.2.1 Function_NodeCallback() . . . . .	761
14.79.3 Member Function Documentation . . . . .	761
14.79.3.1 Destroy() . . . . .	761
14.79.3.2 operator()() . . . . .	761
14.80 gcstring Class Reference . . . . .	762
14.80.1 Constructor & Destructor Documentation . . . . .	763
14.80.1.1 gcstring() [1/5] . . . . .	763
14.80.1.2 gcstring() [2/5] . . . . .	763
14.80.1.3 gcstring() [3/5] . . . . .	763
14.80.1.4 gcstring() [4/5] . . . . .	763
14.80.1.5 gcstring() [5/5] . . . . .	764
14.80.1.6 ~gcstring() . . . . .	764
14.80.2 Member Function Documentation . . . . .	764
14.80.2.1 _npos() . . . . .	764
14.80.2.2 append() [1/2] . . . . .	764
14.80.2.3 append() [2/2] . . . . .	764
14.80.2.4 assign() [1/4] . . . . .	764
14.80.2.5 assign() [2/4] . . . . .	765
14.80.2.6 assign() [3/4] . . . . .	765
14.80.2.7 assign() [4/4] . . . . .	765
14.80.2.8 c_str() . . . . .	765
14.80.2.9 compare() . . . . .	765
14.80.2.10 empty() . . . . .	765
14.80.2.11 find() [1/5] . . . . .	766
14.80.2.12 find() [2/5] . . . . .	766
14.80.2.13 find() [3/5] . . . . .	766
14.80.2.14 find() [4/5] . . . . .	766
14.80.2.15 find() [5/5] . . . . .	766
14.80.2.16 find_first_not_of() . . . . .	766
14.80.2.17 find_first_of() . . . . .	767
14.80.2.18 length() . . . . .	767
14.80.2.19 max_size() . . . . .	767
14.80.2.20 operator const char *() . . . . .	767
14.80.2.21 operator delete() [1/2] . . . . .	767
14.80.2.22 operator delete() [2/2] . . . . .	767
14.80.2.23 operator new() [1/2] . . . . .	767
14.80.2.24 operator new() [2/2] . . . . .	768

---

14.80.2.25 operator"!=() [1/2]	768
14.80.2.26 operator"!=() [2/2]	768
14.80.2.27 operator+=() [1/5]	768
14.80.2.28 operator+=() [2/5]	768
14.80.2.29 operator+=() [3/5]	768
14.80.2.30 operator+=() [4/5]	768
14.80.2.31 operator+=() [5/5]	769
14.80.2.32 operator<()	769
14.80.2.33 operator=()	769
14.80.2.34 operator==() [1/2]	769
14.80.2.35 operator==() [2/2]	769
14.80.2.36 operator>()	769
14.80.2.37 resize()	769
14.80.2.38 size()	770
14.80.2.39 substr()	770
14.80.2.40 swap()	770
14.80.3 Friends And Related Function Documentation	770
14.80.3.1 operator+ [1/3]	770
14.80.3.2 operator+ [2/3]	770
14.80.3.3 operator+ [3/3]	770
14.80.4 Member Data Documentation	771
14.80.4.1 npos	771
14.81 GrabInfo Struct Reference	771
14.81.1 Constructor & Destructor Documentation	771
14.81.1.1 GrabInfo()	771
14.81.2 Member Data Documentation	771
14.81.2.1 imageEventHandler	771
14.81.2.2 numImagesGrabbed	772
14.81.2.3 numIncompleteImages	772
14.81.2.4 numRemovals	772
14.82 GVCP_CHUNK_TRAILER Struct Reference	772
14.82.1 Detailed Description	772
14.82.2 Member Data Documentation	772
14.82.2.1 ChunkID	772
14.82.2.2 ChunkLength	773
14.83 GVCP_EVENT_ITEM Struct Reference	773
14.83.1 Detailed Description	773
14.83.2 Member Data Documentation	773
14.83.2.1 BlockId	773
14.83.2.2 EventId	773
14.83.2.3 ReservedOrEventSize	774
14.83.2.4 StreamChannelId	774

---

---

14.83.2.5 TimestampHigh . . . . .	774
14.83.2.6 TimestampLow . . . . .	774
14.84 GVCP_EVENT_ITEM_BASIC Struct Reference . . . . .	774
14.84.1 Detailed Description . . . . .	774
14.84.2 Member Data Documentation . . . . .	774
14.84.2.1 EventId . . . . .	775
14.84.2.2 ReservedOrEventSize . . . . .	775
14.85 GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference . . . . .	775
14.85.1 Detailed Description . . . . .	775
14.85.2 Member Data Documentation . . . . .	775
14.85.2.1 BlockId . . . . .	775
14.85.2.2 BlockId64High . . . . .	776
14.85.2.3 BlockId64Low . . . . .	776
14.85.2.4 EventId . . . . .	776
14.85.2.5 ReservedOrEventSize . . . . .	776
14.85.2.6 StreamChannelId . . . . .	776
14.85.2.7 TimestampHigh . . . . .	776
14.85.2.8 TimestampLow . . . . .	776
14.86 GVCP_EVENT_REQUEST Struct Reference . . . . .	777
14.86.1 Detailed Description . . . . .	777
14.86.2 Member Data Documentation . . . . .	777
14.86.2.1 Header . . . . .	777
14.86.2.2 Items . . . . .	777
14.87 GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference . . . . .	778
14.87.1 Detailed Description . . . . .	778
14.87.2 Member Data Documentation . . . . .	778
14.87.2.1 Header . . . . .	778
14.87.2.2 Items . . . . .	778
14.88 GVCP_EVENTDATA_REQUEST Struct Reference . . . . .	779
14.88.1 Detailed Description . . . . .	779
14.88.2 Member Data Documentation . . . . .	779
14.88.2.1 Data . . . . .	779
14.88.2.2 Event . . . . .	779
14.88.2.3 Header . . . . .	780
14.89 GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference . . . . .	780
14.89.1 Detailed Description . . . . .	780
14.89.2 Member Data Documentation . . . . .	780
14.89.2.1 Data . . . . .	780
14.89.2.2 Event . . . . .	781
14.89.2.3 Header . . . . .	781
14.90 GVCP_REQUEST_HEADER Struct Reference . . . . .	781
14.90.1 Detailed Description . . . . .	781

---

---

14.90.2 Member Data Documentation . . . . .	781
14.90.2.1 Command . . . . .	781
14.90.2.2 Flags . . . . .	781
14.90.2.3 Length . . . . .	782
14.90.2.4 Magic . . . . .	782
14.90.2.5 ReqId . . . . .	782
14.91 H264Option Struct Reference . . . . .	782
14.91.1 Detailed Description . . . . .	782
14.91.2 Constructor & Destructor Documentation . . . . .	783
14.91.2.1 H264Option() . . . . .	783
14.91.3 Member Data Documentation . . . . .	783
14.91.3.1 bitrate . . . . .	783
14.91.3.2 frameRate . . . . .	783
14.91.3.3 height . . . . .	783
14.91.3.4 reserved . . . . .	783
14.91.3.5 width . . . . .	784
14.92 ICameraBase Class Reference . . . . .	784
14.92.1 Detailed Description . . . . .	786
14.92.2 Constructor & Destructor Documentation . . . . .	786
14.92.2.1 ~ICameraBase() . . . . .	786
14.92.2.2 ICameraBase() [1/2] . . . . .	786
14.92.2.3 ICameraBase() [2/2] . . . . .	786
14.92.3 Member Function Documentation . . . . .	786
14.92.3.1 BeginAcquisition() . . . . .	786
14.92.3.2 DelInit() . . . . .	787
14.92.3.3 DiscoverMaxPacketSize() . . . . .	787
14.92.3.4 EndAcquisition() . . . . .	787
14.92.3.5 ForceIP() . . . . .	787
14.92.3.6 GetAccessMode() . . . . .	787
14.92.3.7 GetBufferOwnership() . . . . .	787
14.92.3.8 GetGuiXml() . . . . .	788
14.92.3.9 GetNextImage() . . . . .	788
14.92.3.10 GetNodeMap() . . . . .	788
14.92.3.11 GetNumDataStreams() . . . . .	788
14.92.3.12 GetNumImagesInUse() . . . . .	788
14.92.3.13 GetTLDeviceNodeMap() . . . . .	788
14.92.3.14 GetTlStreamNodeMap() . . . . .	789
14.92.3.15 GetUniqueId() . . . . .	789
14.92.3.16 GetUserBufferCount() . . . . .	789
14.92.3.17 GetUserBufferSize() . . . . .	789
14.92.3.18 GetUserBufferTotalSize() . . . . .	789
14.92.3.19 Init() . . . . .	789

---

---

14.92.3.20 IsInitialized()	790
14.92.3.21 IsStreaming()	790
14.92.3.22 IsValid()	790
14.92.3.23 operator=()	790
14.92.3.24 ReadPort()	790
14.92.3.25 RegisterEventHandler() [1/2]	790
14.92.3.26 RegisterEventHandler() [2/2]	791
14.92.3.27 SetBufferOwnership()	791
14.92.3.28 SetUserBuffers() [1/2]	791
14.92.3.29 SetUserBuffers() [2/2]	791
14.92.3.30 UnregisterEventHandler()	791
14.92.3.31 WritePort()	792
14.92.4 Friends And Related Function Documentation	792
14.92.4.1 CameraInternal	792
14.92.4.2 Interfacelmpl	792
14.92.5 Member Data Documentation	792
14.92.5.1 m_pCameraBaseData	792
14.92.5.2 TLDevice	792
14.92.5.3 TLStream	793
14.93 ICameraList Class Reference	793
14.93.1 Detailed Description	794
14.93.2 Constructor & Destructor Documentation	794
14.93.2.1 ~ICameraList()	794
14.93.2.2 ICameraList() [1/2]	794
14.93.2.3 ICameraList() [2/2]	794
14.93.3 Member Function Documentation	794
14.93.3.1 Append()	794
14.93.3.2 Clear()	795
14.93.3.3 GetByDeviceID()	795
14.93.3.4 GetByIndex()	795
14.93.3.5 GetBySerial()	795
14.93.3.6 GetSize()	795
14.93.3.7 operator=()	795
14.93.3.8 operator[]()	796
14.93.3.9 RemoveByDeviceID()	796
14.93.3.10 RemoveByIndex()	796
14.93.3.11 RemoveBySerial()	796
14.93.4 Friends And Related Function Documentation	796
14.93.4.1 CameraListImpl	796
14.93.4.2 Interfacelmpl	796
14.93.5 Member Data Documentation	797
14.93.5.1 m_pCameraListData	797

---

14.94 IChunkData Class Reference . . . . .	797
14.94.1 Detailed Description . . . . .	798
14.94.2 Constructor & Destructor Documentation . . . . .	798
14.94.2.1 ~IChunkData() . . . . .	798
14.94.2.2 IChunkData() . . . . .	798
14.94.3 Member Function Documentation . . . . .	799
14.94.3.1 GetBlackLevel() . . . . .	799
14.94.3.2 GetCounterValue() . . . . .	799
14.94.3.3 GetCRC() . . . . .	799
14.94.3.4 GetEncoderValue() . . . . .	799
14.94.3.5 GetExposureEndLineStatusAll() . . . . .	799
14.94.3.6 GetExposureTime() . . . . .	800
14.94.3.7 GetFrameID() . . . . .	800
14.94.3.8 GetGain() . . . . .	800
14.94.3.9 GetHeight() . . . . .	800
14.94.3.10 GetImage() . . . . .	800
14.94.3.11 GetInferenceBoundingBoxResult() . . . . .	800
14.94.3.12 GetInferenceConfidence() . . . . .	801
14.94.3.13 GetInferenceFrameId() . . . . .	801
14.94.3.14 GetInferenceResult() . . . . .	801
14.94.3.15 GetLinePitch() . . . . .	801
14.94.3.16 GetLineStatusAll() . . . . .	801
14.94.3.17 GetOffsetX() . . . . .	801
14.94.3.18 GetOffsetY() . . . . .	802
14.94.3.19 GetPartSelector() . . . . .	802
14.94.3.20 GetPixelDynamicRangeMax() . . . . .	802
14.94.3.21 GetPixelDynamicRangeMin() . . . . .	802
14.94.3.22 GetScan3dAxisMax() . . . . .	802
14.94.3.23 GetScan3dAxisMin() . . . . .	802
14.94.3.24 GetScan3dCoordinateOffset() . . . . .	803
14.94.3.25 GetScan3dCoordinateReferenceValue() . . . . .	803
14.94.3.26 GetScan3dCoordinateScale() . . . . .	803
14.94.3.27 GetScan3dInvalidDataValue() . . . . .	803
14.94.3.28 GetScan3dTransformValue() . . . . .	803
14.94.3.29 GetScanLineSelector() . . . . .	803
14.94.3.30 GetSequencerSetActive() . . . . .	804
14.94.3.31 GetSerialDataLength() . . . . .	804
14.94.3.32 GetStreamChannelID() . . . . .	804
14.94.3.33 GetTimerValue() . . . . .	804
14.94.3.34 GetTimestamp() . . . . .	804
14.94.3.35 GetTimestampLatchValue() . . . . .	804
14.94.3.36 GetTransferBlockID() . . . . .	805

14.94.3.37 GetTransferQueueCurrentBlockCount()	805
14.94.3.38 GetWidth()	805
14.94.3.39 SetChunks()	805
14.95 IDataStream Class Reference	806
14.95.1 Constructor & Destructor Documentation	806
14.95.1.1 ~IDataStream()	807
14.95.1.2 IDataStream()	807
14.95.2 Member Function Documentation	807
14.95.2.1 AnnounceImage() [1/3]	807
14.95.2.2 AnnounceImage() [2/3]	807
14.95.2.3 AnnounceImage() [3/3]	807
14.95.2.4 AttachBuffer()	807
14.95.2.5 CleanupChunkAdapter()	808
14.95.2.6 FlushQueueAllDiscard()	808
14.95.2.7 GetBufferChunkData()	808
14.95.2.8 GetBufferInfoBool8Type()	808
14.95.2.9 GetBufferInfoPtrType()	808
14.95.2.10 GetBufferInfoSizeType()	808
14.95.2.11 GetBufferInfoUInt64Type()	809
14.95.2.12 GetDeviceNodeMap()	809
14.95.2.13 GetNextImage()	809
14.95.2.14 GetNextImageInternal()	809
14.95.2.15 GetNodeMap()	809
14.95.2.16 GetNumImagesInUse()	809
14.95.2.17 GetPort()	809
14.95.2.18 GetStreamInfoBool8Type()	810
14.95.2.19 GetStreamInfoSizeType()	810
14.95.2.20 GetStreamType()	810
14.95.2.21 InitChunkAdapter()	810
14.95.2.22 IsCRCCheckEnabled()	810
14.95.2.23 IsImageInUse()	810
14.95.2.24 IsStreaming()	810
14.95.2.25 KillBufferEvent()	811
14.95.2.26 RegisterImageEventHandler()	811
14.95.2.27 ReleaseImage()	811
14.95.2.28 RevokeImages()	811
14.95.2.29 StartStream()	811
14.95.2.30 StopStream()	811
14.95.2.31 TransportLayerStreamInfo()	811
14.95.2.32 UnregisterImageEventHandler()	812
14.95.2.33 WaitOnImageEvent()	812
14.96 IDevFileStreamBase< CharType, Traits > Class Template Reference	812

---

14.96.1 Member Typedef Documentation . . . . .	813
14.96.1.1 filebuf_type . . . . .	813
14.96.1.2 ios_type . . . . .	813
14.96.1.3 istream_type . . . . .	813
14.96.2 Member Function Documentation . . . . .	813
14.96.2.1 close() . . . . .	813
14.96.2.2 is_open() . . . . .	814
14.96.2.3 open() . . . . .	814
14.96.2.4 rdbuf() . . . . .	814
14.97 IDevFileStreamBuf< CharType, Traits > Class Template Reference . . . . .	814
14.97.1 Constructor & Destructor Documentation . . . . .	815
14.97.1.1 IDevFileStreamBuf() . . . . .	815
14.97.1.2 ~IDevFileStreamBuf() . . . . .	815
14.97.2 Member Function Documentation . . . . .	816
14.97.2.1 close() . . . . .	816
14.97.2.2 is_open() . . . . .	816
14.97.2.3 open() . . . . .	816
14.97.2.4 pbackfail() . . . . .	816
14.97.2.5 underflow() . . . . .	816
14.98 IDeviceArrivalEventHandler Class Reference . . . . .	817
14.98.1 Constructor & Destructor Documentation . . . . .	818
14.98.1.1 ~IDeviceArrivalEventHandler() . . . . .	818
14.98.1.2 IDeviceArrivalEventHandler() [1/2] . . . . .	818
14.98.1.3 IDeviceArrivalEventHandler() [2/2] . . . . .	818
14.98.2 Member Function Documentation . . . . .	818
14.98.2.1 OnDeviceArrival() . . . . .	818
14.98.2.2 operator=(()) . . . . .	819
14.99 IDeviceEventHandler Class Reference . . . . .	819
14.99.1 Constructor & Destructor Documentation . . . . .	820
14.99.1.1 ~IDeviceEventHandler() . . . . .	820
14.99.1.2 IDeviceEventHandler() [1/2] . . . . .	820
14.99.1.3 IDeviceEventHandler() [2/2] . . . . .	820
14.99.2 Member Function Documentation . . . . .	820
14.99.2.1 GetDeviceEventId() . . . . .	820
14.99.2.2 GetDeviceEventName() . . . . .	821
14.99.2.3 OnDeviceEvent() . . . . .	821
14.99.2.4 operator=(()) . . . . .	821
14.100 IDeviceRemovalEventHandler Class Reference . . . . .	821
14.100.1 Constructor & Destructor Documentation . . . . .	822
14.100.1.1 ~IDeviceRemovalEventHandler() . . . . .	822
14.100.1.2 IDeviceRemovalEventHandler() [1/2] . . . . .	822
14.100.1.3 IDeviceRemovalEventHandler() [2/2] . . . . .	823

---

14.100.2 Member Function Documentation . . . . .	823
14.100.2.1 OnDeviceRemoval() . . . . .	823
14.100.2.2 operator=( ) . . . . .	823
14.101 IImage Class Reference . . . . .	823
14.101.1 Detailed Description . . . . .	825
14.101.2 Constructor & Destructor Documentation . . . . .	825
14.101.2.1 ~IImage() . . . . .	825
14.101.2.2 IImage() . . . . .	825
14.101.3 Member Function Documentation . . . . .	825
14.101.3.1 CalculateStatistics() . . . . .	825
14.101.3.2 CheckCRC() . . . . .	825
14.101.3.3 Convert() [1/2] . . . . .	826
14.101.3.4 Convert() [2/2] . . . . .	826
14.101.3.5 DeepCopy() . . . . .	826
14.101.3.6 GetBitsPerPixel() . . . . .	826
14.101.3.7 GetBufferSize() . . . . .	826
14.101.3.8 GetChunkData() . . . . .	827
14.101.3.9 GetChunkLayoutId() . . . . .	827
14.101.3.10 GetColorProcessing() . . . . .	827
14.101.3.11 GetData() . . . . .	827
14.101.3.12 GetDataAbsoluteMax() . . . . .	827
14.101.3.13 GetDataAbsoluteMin() . . . . .	827
14.101.3.14 GetFrameID() . . . . .	828
14.101.3.15 GetHeight() . . . . .	828
14.101.3.16 GetID() . . . . .	828
14.101.3.17 GetImageData() . . . . .	828
14.101.3.18 GetImageSize() . . . . .	828
14.101.3.19 GetImageStatus() . . . . .	828
14.101.3.20 GetNumChannels() . . . . .	829
14.101.3.21 GetPayloadType() . . . . .	829
14.101.3.22 GetPixelFormat() . . . . .	829
14.101.3.23 GetPixelFormatIntType() . . . . .	829
14.101.3.24 GetPixelFormatName() . . . . .	829
14.101.3.25 GetPrivateData() . . . . .	829
14.101.3.26 GetStride() . . . . .	830
14.101.3.27 GetTimeStamp() . . . . .	830
14.101.3.28 GetTLPayloadType() . . . . .	830
14.101.3.29 GetTLPixelFormat() . . . . .	830
14.101.3.30 GetTLPixelFormatNamespace() . . . . .	830
14.101.3.31 GetValidPayloadSize() . . . . .	830
14.101.3.32 GetWidth() . . . . .	831
14.101.3.33 GetXOffset() . . . . .	831

---

14.101.3.34 GetXPadding()	831
14.101.3.35 GetYOffset()	831
14.101.3.36 GetYPadding()	831
14.101.3.37 HasCRC()	831
14.101.3.38 IsIncomplete()	832
14.101.3.39 IsInUse()	832
14.101.3.40 Release()	832
14.101.3.41 ResetImage() [1/2]	832
14.101.3.42 ResetImage() [2/2]	832
14.101.3.43 Save() [1/8]	833
14.101.3.44 Save() [2/8]	833
14.101.3.45 Save() [3/8]	833
14.101.3.46 Save() [4/8]	833
14.101.3.47 Save() [5/8]	833
14.101.3.48 Save() [6/8]	834
14.101.3.49 Save() [7/8]	834
14.101.3.50 Save() [8/8]	834
14.101.4 Friends And Related Function Documentation	834
14.101.4.1 Stream	834
14.102 IImageEventHandler Class Reference	835
14.102.1 Constructor & Destructor Documentation	836
14.102.1.1 ~IImageEventHandler()	836
14.102.1.2 IImageEventHandler() [1/2]	836
14.102.1.3 IImageEventHandler() [2/2]	836
14.102.2 Member Function Documentation	836
14.102.2.1 OnImageEvent()	836
14.102.2.2 operator=( )	836
14.103 IImageStatistics Class Reference	837
14.103.1 Detailed Description	837
14.103.2 Constructor & Destructor Documentation	838
14.103.2.1 ~IImageStatistics()	838
14.103.2.2 IImageStatistics() [1/2]	838
14.103.2.3 IImageStatistics() [2/2]	838
14.103.3 Member Function Documentation	838
14.103.3.1 DisableAll()	838
14.103.3.2 EnableAll()	838
14.103.3.3 EnableGreyOnly()	838
14.103.3.4 EnableHSLOnly()	839
14.103.3.5 EnableRGBOnly()	839
14.103.3.6 GetChannelStatus()	839
14.103.3.7 GetHistogram()	839
14.103.3.8 GetMean()	839

---

14.103.3.9 GetNumPixelValues()	840
14.103.3.10 GetPixelValueRange()	840
14.103.3.11 GetRange()	840
14.103.3.12 GetStatistics()	840
14.103.3.13 SetChannelStatus()	841
14.104 IInterface Class Reference	841
14.104.1 Detailed Description	842
14.104.2 Constructor & Destructor Documentation	842
14.104.2.1 ~IInterface()	842
14.104.2.2 IInterface() [1/2]	843
14.104.2.3 IInterface() [2/2]	843
14.104.3 Member Function Documentation	843
14.104.3.1 GetCameras()	843
14.104.3.2 GetTLNodeMap()	843
14.104.3.3 IsInUse()	843
14.104.3.4 IsValid()	843
14.104.3.5 operator=()	844
14.104.3.6 RegisterEventHandler()	844
14.104.3.7 SendActionCommand()	844
14.104.3.8 UnregisterEventHandler()	844
14.104.3.9 UpdateCameras()	844
14.104.4 Friends And Related Function Documentation	844
14.104.4.1 InterfaceInternal	845
14.104.4.2 SystemImpl	845
14.104.5 Member Data Documentation	845
14.104.5.1 m_pInterfaceData	845
14.104.5.2 TLIInterface	845
14.105 IInterfaceArrivalEventHandler Class Reference	846
14.105.1 Constructor & Destructor Documentation	847
14.105.1.1 ~IInterfaceArrivalEventHandler()	847
14.105.1.2 IInterfaceArrivalEventHandler() [1/2]	847
14.105.1.3 IInterfaceArrivalEventHandler() [2/2]	847
14.105.2 Member Function Documentation	847
14.105.2.1 OnInterfaceArrival()	847
14.105.2.2 operator=()	848
14.106 IInterfaceEventHandler Class Reference	848
14.106.1 Constructor & Destructor Documentation	849
14.106.1.1 ~IInterfaceEventHandler()	849
14.106.1.2 IInterfaceEventHandler() [1/2]	849
14.106.1.3 IInterfaceEventHandler() [2/2]	850
14.106.2 Member Function Documentation	850
14.106.2.1 OnDeviceArrival()	850

---

---

14.106.2.2 OnDeviceRemoval()	850
14.106.2.3 operator=()	850
14.107 IInterfaceList Class Reference	851
14.107.1 Detailed Description	851
14.107.2 Constructor & Destructor Documentation	851
14.107.2.1 ~IInterfaceList()	852
14.107.2.2 IInterfaceList() [1/2]	852
14.107.2.3 IInterfaceList() [2/2]	852
14.107.3 Member Function Documentation	852
14.107.3.1 Clear()	852
14.107.3.2 GetByIndex()	852
14.107.3.3 GetSize()	852
14.107.3.4 operator=()	853
14.107.3.5 operator[]()	853
14.107.4 Member Data Documentation	853
14.107.4.1 m_pInterfaceListData	853
14.108 IInterfaceRemovalEventHandler Class Reference	853
14.108.1 Constructor & Destructor Documentation	854
14.108.1.1 ~IInterfaceRemovalEventHandler()	854
14.108.1.2 IInterfaceRemovalEventHandler() [1/2]	854
14.108.1.3 IInterfaceRemovalEventHandler() [2/2]	855
14.108.2 Member Function Documentation	855
14.108.2.1 OnInterfaceRemoval()	855
14.108.2.2 operator=()	855
14.109 ILoggingEventHandler Class Reference	855
14.109.1 Constructor & Destructor Documentation	856
14.109.1.1 ~ILoggingEventHandler()	856
14.109.1.2 ILoggingEventHandler() [1/2]	856
14.109.1.3 ILoggingEventHandler() [2/2]	857
14.109.2 Member Function Documentation	857
14.109.2.1 OnLogEvent()	857
14.109.2.2 operator=()	857
14.110 Image Class Reference	857
14.110.1 Detailed Description	861
14.110.2 Constructor & Destructor Documentation	861
14.110.2.1 ~Image()	861
14.110.2.2 Image() [1/3]	861
14.110.2.3 Image() [2/3]	861
14.110.2.4 Image() [3/3]	861
14.110.3 Member Function Documentation	862
14.110.3.1 CalculateStatistics()	862
14.110.3.2 CheckCRC()	862

---

---

14.110.3.3 Convert() [1/3] . . . . .	862
14.110.3.4 Convert() [2/3] . . . . .	863
14.110.3.5 Convert() [3/3] . . . . .	863
14.110.3.6 Create() [1/3] . . . . .	863
14.110.3.7 Create() [2/3] . . . . .	864
14.110.3.8 Create() [3/3] . . . . .	864
14.110.3.9 CreateShared() . . . . .	864
14.110.3.10 DeepCopy() [1/2] . . . . .	864
14.110.3.11 DeepCopy() [2/2] . . . . .	865
14.110.3.12 GetBitsPerPixel() . . . . .	865
14.110.3.13 GetBufferSize() . . . . .	865
14.110.3.14 GetChunkData() . . . . .	866
14.110.3.15 GetChunkLayoutId() . . . . .	866
14.110.3.16 GetColorProcessing() . . . . .	866
14.110.3.17 GetData() . . . . .	867
14.110.3.18 GetDataAbsoluteMax() . . . . .	867
14.110.3.19 GetDataAbsoluteMin() . . . . .	867
14.110.3.20 GetDefaultColorProcessing() . . . . .	868
14.110.3.21 GetFrameID() . . . . .	868
14.110.3.22 GetHeight() . . . . .	868
14.110.3.23 GetID() . . . . .	869
14.110.3.24 GetImageData() . . . . .	869
14.110.3.25 GetImageSize() . . . . .	869
14.110.3.26 GetImageStatus() . . . . .	869
14.110.3.27 GetImageStatusDescription() . . . . .	870
14.110.3.28 GetNumChannels() . . . . .	870
14.110.3.29 GetPayloadType() . . . . .	870
14.110.3.30 GetPixelFormat() . . . . .	871
14.110.3.31 GetPixelFormatIntType() . . . . .	871
14.110.3.32 GetPixelFormatName() . . . . .	871
14.110.3.33 GetPrivateData() . . . . .	872
14.110.3.34 GetStride() . . . . .	872
14.110.3.35 GetTimeStamp() . . . . .	872
14.110.3.36 GetTLPayloadType() . . . . .	873
14.110.3.37 GetTLPixelFormat() . . . . .	873
14.110.3.38 GetTLPixelFormatNamespace() . . . . .	873
14.110.3.39 GetValidPayloadSize() . . . . .	874
14.110.3.40 GetWidth() . . . . .	874
14.110.3.41 GetXOffset() . . . . .	874
14.110.3.42 GetXPadding() . . . . .	875
14.110.3.43 GetYOffset() . . . . .	875
14.110.3.44 GetYPadding() . . . . .	875

---

---

14.110.3.45 HasCRC()	876
14.110.3.46 IsCompressed()	876
14.110.3.47 IsIncomplete()	876
14.110.3.48 IsInUse()	876
14.110.3.49 Release()	877
14.110.3.50 ResetImage() [1/2]	877
14.110.3.51 ResetImage() [2/2]	877
14.110.3.52 Save() [1/8]	878
14.110.3.53 Save() [2/8]	878
14.110.3.54 Save() [3/8]	878
14.110.3.55 Save() [4/8]	879
14.110.3.56 Save() [5/8]	879
14.110.3.57 Save() [6/8]	879
14.110.3.58 Save() [7/8]	880
14.110.3.59 Save() [8/8]	880
14.110.3.60 SetDefaultColorProcessing()	880
14.110.4 Friends And Related Function Documentation	881
14.110.4.1 IDataStream	881
14.110.4.2 ImageConverter	881
14.110.4.3 ImageFiler	881
14.110.4.4 ImageStatsCalculator	881
14.110.4.5 ImageUtilityImpl	881
14.110.4.6 ImageUtilityPolarizationImpl	882
14.110.4.7 Stream	882
14.111 ImageEventHandler Class Reference	882
14.111.1 Detailed Description	883
14.111.2 Constructor & Destructor Documentation	883
14.111.2.1 ImageEventHandler()	884
14.111.2.2 ~ImageEventHandler()	884
14.111.3 Member Function Documentation	884
14.111.3.1 OnImageEvent()	884
14.111.3.2 operator=( )	884
14.112 ImageEventHandlerImpl Class Reference	885
14.112.1 Constructor & Destructor Documentation	886
14.112.1.1 ImageEventHandlerImpl() [1/2]	886
14.112.1.2 ~ImageEventHandlerImpl() [1/2]	886
14.112.1.3 ImageEventHandlerImpl() [2/2]	886
14.112.1.4 ~ImageEventHandlerImpl() [2/2]	886
14.112.2 Member Function Documentation	886
14.112.2.1 getImageCount()	887
14.112.2.2 getMaxImages()	887
14.112.2.3 OnImageEvent() [1/2]	887

---

---

14.112.2.4 OnImageEvent() [2/2] . . . . .	887
14.113 ImagePtr Class Reference . . . . .	888
14.113.1 Detailed Description . . . . .	889
14.113.2 Constructor & Destructor Documentation . . . . .	889
14.113.2.1 ImagePtr() [1/4] . . . . .	889
14.113.2.2 ImagePtr() [2/4] . . . . .	889
14.113.2.3 ImagePtr() [3/4] . . . . .	889
14.113.2.4 ImagePtr() [4/4] . . . . .	889
14.113.2.5 ~ImagePtr() . . . . .	890
14.113.3 Member Function Documentation . . . . .	890
14.113.3.1 operator=() . . . . .	890
14.114 ImageStatistics Class Reference . . . . .	890
14.114.1 Detailed Description . . . . .	891
14.114.2 Constructor & Destructor Documentation . . . . .	892
14.114.2.1 ImageStatistics() [1/2] . . . . .	892
14.114.2.2 ~ImageStatistics() . . . . .	892
14.114.2.3 ImageStatistics() [2/2] . . . . .	892
14.114.3 Member Function Documentation . . . . .	892
14.114.3.1 DisableAll() . . . . .	892
14.114.3.2 EnableAll() . . . . .	892
14.114.3.3 EnableGreyOnly() . . . . .	893
14.114.3.4 EnableHSLOnly() . . . . .	893
14.114.3.5 EnableRGBOnly() . . . . .	893
14.114.3.6 GetChannelStatus() . . . . .	893
14.114.3.7 GetHistogram() . . . . .	894
14.114.3.8 GetMean() . . . . .	894
14.114.3.9 GetNumPixelValues() . . . . .	894
14.114.3.10 GetPixelValueRange() . . . . .	895
14.114.3.11 GetRange() . . . . .	895
14.114.3.12 GetStatistics() . . . . .	895
14.114.3.13 operator=() . . . . .	896
14.114.3.14 SetChannelStatus() . . . . .	896
14.114.4 Friends And Related Function Documentation . . . . .	897
14.114.4.1 ImageStatsCalculator . . . . .	897
14.115 ImageUtility Class Reference . . . . .	897
14.115.1 Detailed Description . . . . .	898
14.115.2 Member Enumeration Documentation . . . . .	898
14.115.2.1 ImageScalingAlgorithm . . . . .	898
14.115.2.2 SourceDataRange . . . . .	898
14.115.3 Member Function Documentation . . . . .	899
14.115.3.1 CreateNormalized() [1/5] . . . . .	899
14.115.3.2 CreateNormalized() [2/5] . . . . .	899

---

---

14.115.3.3 CreateNormalized() [3/5] . . . . .	900
14.115.3.4 CreateNormalized() [4/5] . . . . .	900
14.115.3.5 CreateNormalized() [5/5] . . . . .	901
14.115.3.6 CreateScaled() [1/2] . . . . .	901
14.115.3.7 CreateScaled() [2/2] . . . . .	901
<b>14.116 ImageUtilityHeatmap Class Reference . . . . .</b>	<b>902</b>
<b>14.116.1 Detailed Description . . . . .</b>	<b>903</b>
<b>14.116.2 Member Enumeration Documentation . . . . .</b>	<b>903</b>
<b>14.116.2.1 HeatmapColor . . . . .</b>	<b>903</b>
<b>14.116.3 Member Function Documentation . . . . .</b>	<b>903</b>
<b>14.116.3.1 CreateHeatmap() [1/2] . . . . .</b>	<b>903</b>
<b>14.116.3.2 CreateHeatmap() [2/2] . . . . .</b>	<b>904</b>
<b>14.116.3.3 GetHeatmapColorGradient() . . . . .</b>	<b>904</b>
<b>14.116.3.4 GetHeatmapRange() . . . . .</b>	<b>905</b>
<b>14.116.3.5 SetHeatmapColorGradient() . . . . .</b>	<b>905</b>
<b>14.116.3.6 SetHeatmapRange() . . . . .</b>	<b>905</b>
<b>14.117 ImageUtilityPolarization Class Reference . . . . .</b>	<b>906</b>
<b>14.117.1 Detailed Description . . . . .</b>	<b>907</b>
<b>14.117.2 Member Enumeration Documentation . . . . .</b>	<b>907</b>
<b>14.117.2.1 PolarizationQuadrant . . . . .</b>	<b>907</b>
<b>14.117.3 Member Function Documentation . . . . .</b>	<b>907</b>
<b>14.117.3.1 CreateAolp() [1/2] . . . . .</b>	<b>908</b>
<b>14.117.3.2 CreateAolp() [2/2] . . . . .</b>	<b>908</b>
<b>14.117.3.3 CreateDolp() [1/2] . . . . .</b>	<b>908</b>
<b>14.117.3.4 CreateDolp() [2/2] . . . . .</b>	<b>909</b>
<b>14.117.3.5 CreateGlareReduced() [1/2] . . . . .</b>	<b>909</b>
<b>14.117.3.6 CreateGlareReduced() [2/2] . . . . .</b>	<b>910</b>
<b>14.117.3.7 CreateStokesS0() [1/2] . . . . .</b>	<b>910</b>
<b>14.117.3.8 CreateStokesS0() [2/2] . . . . .</b>	<b>910</b>
<b>14.117.3.9 CreateStokesS1() [1/2] . . . . .</b>	<b>911</b>
<b>14.117.3.10 CreateStokesS1() [2/2] . . . . .</b>	<b>911</b>
<b>14.117.3.11 CreateStokesS2() [1/2] . . . . .</b>	<b>912</b>
<b>14.117.3.12 CreateStokesS2() [2/2] . . . . .</b>	<b>912</b>
<b>14.117.3.13 ExtractPolarQuadrant() [1/2] . . . . .</b>	<b>913</b>
<b>14.117.3.14 ExtractPolarQuadrant() [2/2] . . . . .</b>	<b>913</b>
<b>14.118 InferenceBoundingBox Struct Reference . . . . .</b>	<b>913</b>
<b>14.118.1 Detailed Description . . . . .</b>	<b>914</b>
<b>14.118.2 Member Data Documentation . . . . .</b>	<b>914</b>
<b>14.118.2.1 boxType . . . . .</b>	<b>914</b>
<b>14.118.2.2 circle . . . . .</b>	<b>914</b>
<b>14.118.2.3 classId . . . . .</b>	<b>914</b>
<b>14.118.2.4 confidence . . . . .</b>	<b>915</b>

---

---

14.118.2.5 rect . . . . .	915
14.118.2.6 rotatedRect . . . . .	915
14.119 InferenceBoundingBoxResult Class Reference . . . . .	915
14.119.1 Detailed Description . . . . .	916
14.119.2 Constructor & Destructor Documentation . . . . .	916
14.119.2.1 InferenceBoundingBoxResult() [1/3] . . . . .	916
14.119.2.2 ~InferenceBoundingBoxResult() . . . . .	916
14.119.2.3 InferenceBoundingBoxResult() [2/3] . . . . .	916
14.119.2.4 InferenceBoundingBoxResult() [3/3] . . . . .	916
14.119.3 Member Function Documentation . . . . .	917
14.119.3.1 GetBoxAt() . . . . .	917
14.119.3.2 GetBoxCount() . . . . .	917
14.119.3.3 GetBoxSize() . . . . .	917
14.119.3.4 GetVersion() . . . . .	917
14.119.3.5 operator=() . . . . .	917
14.120 InferenceBoxCircle Struct Reference . . . . .	918
14.120.1 Member Data Documentation . . . . .	918
14.120.1.1 centerXCoord . . . . .	918
14.120.1.2 centerYCoord . . . . .	918
14.120.1.3 radius . . . . .	918
14.121 InferenceBoxRect Struct Reference . . . . .	918
14.121.1 Detailed Description . . . . .	919
14.121.2 Member Data Documentation . . . . .	919
14.121.2.1 bottomRightXCoord . . . . .	919
14.121.2.2 bottomRightYCoord . . . . .	919
14.121.2.3 topLeftXCoord . . . . .	919
14.121.2.4 topLeftYCoord . . . . .	919
14.122 InferenceBoxRotatedRect Struct Reference . . . . .	919
14.122.1 Member Data Documentation . . . . .	920
14.122.1.1 bottomRightXCoord . . . . .	920
14.122.1.2 bottomRightYCoord . . . . .	920
14.122.1.3 rotationAngle . . . . .	920
14.122.1.4 topLeftXCoord . . . . .	920
14.122.1.5 topLeftYCoord . . . . .	920
14.123 int64_avector_t Class Reference . . . . .	920
14.123.1 Detailed Description . . . . .	921
14.123.2 Constructor & Destructor Documentation . . . . .	921
14.123.2.1 int64_avector_t() [1/3] . . . . .	921
14.123.2.2 int64_avector_t() [2/3] . . . . .	921
14.123.2.3 int64_avector_t() [3/3] . . . . .	921
14.123.2.4 ~int64_avector_t() . . . . .	922
14.123.3 Member Function Documentation . . . . .	922

---

---

14.123.3.1 operator delete() . . . . .	922
14.123.3.2 operator new() . . . . .	922
14.123.3.3 operator=() . . . . .	922
14.123.3.4 operator[]() [1/2] . . . . .	922
14.123.3.5 operator[]() [2/2] . . . . .	922
14.123.3.6 size() . . . . .	923
14.123.4 Member Data Documentation . . . . .	923
14.123.4.1 _pCount . . . . .	923
14.123.4.2 _pv . . . . .	923
14.124 IntegerNode Class Reference . . . . .	923
14.124.1 Detailed Description . . . . .	925
14.124.2 Constructor & Destructor Documentation . . . . .	925
14.124.2.1 IntegerNode() [1/2] . . . . .	925
14.124.2.2 IntegerNode() [2/2] . . . . .	925
14.124.2.3 ~IntegerNode() . . . . .	925
14.124.3 Member Function Documentation . . . . .	925
14.124.3.1 GetFloatAlias() . . . . .	926
14.124.3.2 GetInc() . . . . .	926
14.124.3.3 GetIncMode() . . . . .	926
14.124.3.4 GetListOfValidValues() . . . . .	926
14.124.3.5 GetMax() . . . . .	926
14.124.3.6 GetMin() . . . . .	926
14.124.3.7 GetRepresentation() . . . . .	927
14.124.3.8 GetUnit() . . . . .	927
14.124.3.9 GetValue() . . . . .	927
14.124.3.10 ImposeMax() . . . . .	927
14.124.3.11 ImposeMin() . . . . .	927
14.124.3.12 operator()() . . . . .	928
14.124.3.13 operator*() . . . . .	928
14.124.3.14 operator=() . . . . .	928
14.124.3.15 SetReference() . . . . .	928
14.124.3.16 SetValue() . . . . .	928
14.125 Interface Class Reference . . . . .	929
14.125.1 Detailed Description . . . . .	930
14.125.2 Constructor & Destructor Documentation . . . . .	930
14.125.2.1 ~Interface() . . . . .	930
14.125.3 Member Function Documentation . . . . .	930
14.125.3.1 GetCameras() . . . . .	931
14.125.3.2 GetTLNodeMap() . . . . .	931
14.125.3.3 IsInUse() . . . . .	931
14.125.3.4 IsValid() . . . . .	932
14.125.3.5 RegisterEventHandler() . . . . .	932

---

---

14.125.3.6 SendActionCommand() . . . . .	932
14.125.3.7 UnregisterEventHandler() . . . . .	933
14.125.3.8 UpdateCameras() . . . . .	933
14.125.4 Friends And Related Function Documentation . . . . .	934
14.125.4.1 InterfaceInternal . . . . .	934
14.126 InterfaceArrivalEventHandler Class Reference . . . . .	934
14.126.1 Detailed Description . . . . .	935
14.126.2 Constructor & Destructor Documentation . . . . .	935
14.126.2.1 InterfaceArrivalEventHandler() . . . . .	935
14.126.2.2 ~InterfaceArrivalEventHandler() . . . . .	935
14.126.3 Member Function Documentation . . . . .	935
14.126.3.1 OnInterfaceArrival() . . . . .	935
14.126.3.2 operator=() . . . . .	936
14.127 InterfaceEventHandler Class Reference . . . . .	936
14.127.1 Detailed Description . . . . .	937
14.127.2 Constructor & Destructor Documentation . . . . .	937
14.127.2.1 InterfaceEventHandler() . . . . .	938
14.127.2.2 ~InterfaceEventHandler() . . . . .	938
14.127.3 Member Function Documentation . . . . .	938
14.127.3.1 OnDeviceArrival() . . . . .	938
14.127.3.2 OnDeviceRemoval() . . . . .	938
14.127.3.3 operator=() . . . . .	939
14.128 InterfaceEventHandlerImpl Class Reference . . . . .	939
14.128.1 Constructor & Destructor Documentation . . . . .	940
14.128.1.1 InterfaceEventHandlerImpl() [1/3] . . . . .	941
14.128.1.2 ~InterfaceEventHandlerImpl() [1/2] . . . . .	941
14.128.1.3 InterfaceEventHandlerImpl() [2/3] . . . . .	941
14.128.1.4 InterfaceEventHandlerImpl() [3/3] . . . . .	941
14.128.1.5 ~InterfaceEventHandlerImpl() [2/2] . . . . .	941
14.128.2 Member Function Documentation . . . . .	941
14.128.2.1 GetInterfaceId() . . . . .	941
14.128.2.2 OnDeviceArrival() [1/2] . . . . .	942
14.128.2.3 OnDeviceArrival() [2/2] . . . . .	942
14.128.2.4 OnDeviceRemoval() [1/2] . . . . .	942
14.128.2.5 OnDeviceRemoval() [2/2] . . . . .	942
14.128.2.6 PrintGenericHandlerMessage() . . . . .	943
14.129 InterfaceList Class Reference . . . . .	943
14.129.1 Detailed Description . . . . .	944
14.129.2 Constructor & Destructor Documentation . . . . .	944
14.129.2.1 InterfaceList() [1/2] . . . . .	944
14.129.2.2 ~InterfaceList() . . . . .	944
14.129.2.3 InterfaceList() [2/2] . . . . .	945

---

14.129.3 Member Function Documentation . . . . .	945
14.129.3.1 Clear() . . . . .	945
14.129.3.2 GetByIndex() . . . . .	945
14.129.3.3 GetSize() . . . . .	945
14.129.3.4 operator=() . . . . .	946
14.129.3.5 operator[]() . . . . .	946
14.129.4 Friends And Related Function Documentation . . . . .	946
14.129.4.1 SystemImpl . . . . .	946
14.130 InterfacePtr Class Reference . . . . .	947
14.130.1 Detailed Description . . . . .	947
14.130.2 Constructor & Destructor Documentation . . . . .	948
14.130.2.1 InterfacePtr() [1/4] . . . . .	948
14.130.2.2 InterfacePtr() [2/4] . . . . .	948
14.130.2.3 InterfacePtr() [3/4] . . . . .	948
14.130.2.4 InterfacePtr() [4/4] . . . . .	948
14.131 InterfaceRemovalEventHandler Class Reference . . . . .	949
14.131.1 Detailed Description . . . . .	950
14.131.2 Constructor & Destructor Documentation . . . . .	950
14.131.2.1 InterfaceRemovalEventHandler() . . . . .	950
14.131.2.2 ~InterfaceRemovalEventHandler() . . . . .	950
14.131.3 Member Function Documentation . . . . .	950
14.131.3.1 OnInterfaceRemoval() . . . . .	950
14.131.3.2 operator=() . . . . .	951
14.132 IntRegNode Class Reference . . . . .	951
14.132.1 Detailed Description . . . . .	952
14.132.2 Constructor & Destructor Documentation . . . . .	952
14.132.2.1 IntRegNode() [1/2] . . . . .	953
14.132.2.2 IntRegNode() [2/2] . . . . .	953
14.132.2.3 ~IntRegNode() . . . . .	953
14.132.3 Member Function Documentation . . . . .	953
14.132.3.1 SetReference() . . . . .	953
14.133 IplInfo Struct Reference . . . . .	953
14.133.1 Constructor & Destructor Documentation . . . . .	954
14.133.1.1 IplInfo() . . . . .	954
14.133.2 Member Data Documentation . . . . .	954
14.133.2.1 gateway . . . . .	954
14.133.2.2 ipAddress . . . . .	954
14.133.2.3 subnetLength . . . . .	954
14.133.2.4 subnetMask . . . . .	954
14.134 ISystem Class Reference . . . . .	955
14.134.1 Detailed Description . . . . .	956
14.134.2 Constructor & Destructor Documentation . . . . .	956

---

14.134.2.1 ~ISystem()	956
14.134.2.2 ISystem() [1/2]	956
14.134.2.3 ISystem() [2/2]	957
14.134.3 Member Function Documentation	957
14.134.3.1 GetCameras()	957
14.134.3.2 GetInterfaces()	957
14.134.3.3 GetLibraryVersion()	957
14.134.3.4 GetLoggingEventPriorityLevel()	957
14.134.3.5 GetTLNodeMap()	958
14.134.3.6 IsInUse()	958
14.134.3.7 operator=()	958
14.134.3.8 RegisterEventHandler()	958
14.134.3.9 RegisterInterfaceEventHandler()	958
14.134.3.10 RegisterLoggingEventHandler()	958
14.134.3.11 ReleaseInstance()	959
14.134.3.12 SendActionCommand()	959
14.134.3.13 SetLoggingEventPriorityLevel()	959
14.134.3.14 UnregisterAllLoggingEventHandlers()	959
14.134.3.15 UnregisterEventHandler()	959
14.134.3.16 UnregisterInterfaceEventHandler()	960
14.134.3.17 UnregisterLoggingEventHandler()	960
14.134.3.18 UpdateCameras()	960
14.134.3.19 UpdateInterfaceList()	960
14.134.4 Friends And Related Function Documentation	960
14.134.4.1 SystemPtrInternal	960
14.134.5 Member Data Documentation	960
14.134.5.1 TLSYSTEM	961
14.135 ISYSTEMEventHandler Class Reference	961
14.135.1 Constructor & Destructor Documentation	962
14.135.1.1 ~ISYSTEMEventHandler()	962
14.135.1.2 ISYSTEMEventHandler() [1/2]	962
14.135.1.3 ISYSTEMEventHandler() [2/2]	962
14.135.2 Member Function Documentation	962
14.135.2.1 OnInterfaceArrival()	962
14.135.2.2 OnInterfaceRemoval()	963
14.135.2.3 operator=()	963
14.136 JPEGOption Struct Reference	963
14.136.1 Detailed Description	963
14.136.2 Constructor & Destructor Documentation	963
14.136.2.1 JPEGOption()	964
14.136.3 Member Data Documentation	964
14.136.3.1 progressive	964

---

---

14.136.3.2 quality . . . . .	964
14.136.3.3 reserved . . . . .	964
14.137 JPG2Option Struct Reference . . . . .	964
14.137.1 Detailed Description . . . . .	965
14.137.2 Constructor & Destructor Documentation . . . . .	965
14.137.2.1 JPG2Option() . . . . .	965
14.137.3 Member Data Documentation . . . . .	965
14.137.3.1 quality . . . . .	965
14.137.3.2 reserved . . . . .	965
14.138 LibraryVersion Struct Reference . . . . .	966
14.138.1 Detailed Description . . . . .	966
14.138.2 Member Data Documentation . . . . .	966
14.138.2.1 build . . . . .	966
14.138.2.2 major . . . . .	966
14.138.2.3 minor . . . . .	966
14.138.2.4 type . . . . .	967
14.139 LockableObject< Object >::Lock Class Reference . . . . .	967
14.139.1 Detailed Description . . . . .	967
14.139.2 Constructor & Destructor Documentation . . . . .	967
14.139.2.1 Lock() . . . . .	967
14.139.2.2 ~Lock() . . . . .	967
14.140 LockableObject< Object > Class Template Reference . . . . .	968
14.140.1 Detailed Description . . . . .	968
14.140.2 Member Function Documentation . . . . .	969
14.140.2.1 GetLock() . . . . .	969
14.140.3 Friends And Related Function Documentation . . . . .	969
14.140.3.1 Lock . . . . .	969
14.140.4 Member Data Documentation . . . . .	969
14.140.4.1 m_Lock . . . . .	969
14.141 LoggingEventData Class Reference . . . . .	969
14.141.1 Detailed Description . . . . .	970
14.141.2 Constructor & Destructor Documentation . . . . .	970
14.141.2.1 ~LoggingEventData() . . . . .	970
14.141.2.2 LoggingEventData() . . . . .	971
14.141.3 Member Function Documentation . . . . .	971
14.141.3.1 GetCategoryName() . . . . .	971
14.141.3.2 GetLogMessage() . . . . .	971
14.141.3.3 GetNDC() . . . . .	971
14.141.3.4 GetPriority() . . . . .	972
14.141.3.5 GetPriorityName() . . . . .	972
14.141.3.6 GetThreadName() . . . . .	972
14.141.3.7 GetTimestamp() . . . . .	972

---

---

14.141.4 Friends And Related Function Documentation . . . . .	972
14.141.4.1 SystemImpl . . . . .	973
14.142 LoggingEventDataPtr Class Reference . . . . .	973
14.142.1 Detailed Description . . . . .	974
14.142.2 Constructor & Destructor Documentation . . . . .	974
14.142.2.1 LoggingEventDataPtr() [1/4] . . . . .	974
14.142.2.2 LoggingEventDataPtr() [2/4] . . . . .	974
14.142.2.3 LoggingEventDataPtr() [3/4] . . . . .	974
14.142.2.4 LoggingEventDataPtr() [4/4] . . . . .	974
14.143 LoggingEventHandler Class Reference . . . . .	975
14.143.1 Detailed Description . . . . .	976
14.143.2 Constructor & Destructor Documentation . . . . .	976
14.143.2.1 LoggingEventHandler() . . . . .	976
14.143.2.2 ~LoggingEventHandler() . . . . .	976
14.143.3 Member Function Documentation . . . . .	976
14.143.3.1 OnLogEvent() . . . . .	976
14.143.3.2 operator=() . . . . .	977
14.144 LoggingEventHandlerImpl Class Reference . . . . .	977
14.145 Member_NodeCallback< Client, Member > Class Template Reference . . . . .	978
14.145.1 Detailed Description . . . . .	979
14.145.2 Member Typedef Documentation . . . . .	979
14.145.2.1 PMEMBERFUNC . . . . .	979
14.145.3 Constructor & Destructor Documentation . . . . .	980
14.145.3.1 Member_NodeCallback() . . . . .	980
14.145.4 Member Function Documentation . . . . .	980
14.145.4.1 Destroy() . . . . .	980
14.145.4.2 operator()() . . . . .	980
14.146 MJPGOption Struct Reference . . . . .	980
14.146.1 Detailed Description . . . . .	981
14.146.2 Constructor & Destructor Documentation . . . . .	981
14.146.2.1 MJPGOption() . . . . .	981
14.146.3 Member Data Documentation . . . . .	981
14.146.3.1 frameRate . . . . .	981
14.146.3.2 quality . . . . .	981
14.146.3.3 reserved . . . . .	982
14.147 Node Class Reference . . . . .	982
14.147.1 Detailed Description . . . . .	984
14.147.2 Constructor & Destructor Documentation . . . . .	985
14.147.2.1 Node() [1/2] . . . . .	985
14.147.2.2 Node() [2/2] . . . . .	985
14.147.2.3 ~Node() . . . . .	985
14.147.3 Member Function Documentation . . . . .	985

---

---

14.147.3.1 DereisterCallback()	985
14.147.3.2 GetAccessMode()	986
14.147.3.3 GetAlias()	986
14.147.3.4 GetCachingMode()	986
14.147.3.5 GetCastAlias()	986
14.147.3.6 GetChildren()	986
14.147.3.7 GetDescription()	987
14.147.3.8 GetDeviceName()	987
14.147.3.9 GetDisplayName()	987
14.147.3.10 GetDocuURL()	987
14.147.3.11 GetEventID()	987
14.147.3.12 GetName()	987
14.147.3.13 GetNameSpace()	988
14.147.3.14 GetNodeHandle()	988
14.147.3.15 GetNodeMap()	988
14.147.3.16 GetParents()	988
14.147.3.17 GetPollingTime()	988
14.147.3.18 GetPrincipallInterfaceType()	989
14.147.3.19 GetProperty()	989
14.147.3.20 GetPropertyNames()	989
14.147.3.21 GetSelectedFeatures()	989
14.147.3.22 GetSelectingFeatures()	989
14.147.3.23 GetToolTip()	990
14.147.3.24 GetVisibility()	990
14.147.3.25 ImposeAccessMode()	990
14.147.3.26 ImposeVisibility()	990
14.147.3.27 InvalidateNode()	990
14.147.3.28 IsAccessModeCacheable()	990
14.147.3.29 IsCachable()	991
14.147.3.30 IsDeprecated()	991
14.147.3.31 IsFeature()	991
14.147.3.32 IsSelector()	991
14.147.3.33 IsStreamable()	991
14.147.3.34 operator"!="()	991
14.147.3.35 operator==()	992
14.147.3.36 RegisterCallback()	992
14.147.3.37 SetNodeHandle()	992
14.147.3.38 SetNodeMap()	992
14.147.3.39 SetReference() [1/2]	992
14.147.3.40 SetReference() [2/2]	992
14.147.4 Member Data Documentation	993
14.147.4.1 m_Callbacks	993

---

---

14.147.4.2 m_pNodeData . . . . .	993
14.147.4.3 m_pNodeMap . . . . .	993
14.148 NodeMap Class Reference . . . . .	993
14.148.1 Detailed Description . . . . .	995
14.148.2 Constructor & Destructor Documentation . . . . .	995
14.148.2.1 NodeMap() . . . . .	996
14.148.2.2 ~NodeMap() . . . . .	996
14.148.3 Member Function Documentation . . . . .	996
14.148.3.1 ClearXMLCache() . . . . .	996
14.148.3.2 Connect() [1/2] . . . . .	996
14.148.3.3 Connect() [2/2] . . . . .	996
14.148.3.4 Destroy() . . . . .	997
14.148.3.5 GetDeviceName() . . . . .	997
14.148.3.6 GetDeviceVersion() . . . . .	997
14.148.3.7 GetGenApiVersion() . . . . .	997
14.148.3.8 GetLock() . . . . .	997
14.148.3.9 GetModelName() . . . . .	997
14.148.3.10 GetNode() . . . . .	998
14.148.3.11 GetNodeMapHandle() . . . . .	998
14.148.3.12 GetNodes() . . . . .	998
14.148.3.13 GetNumNodes() . . . . .	998
14.148.3.14 GetProductGuid() . . . . .	998
14.148.3.15 GetSchemaVersion() . . . . .	998
14.148.3.16 GetStandardNameSpace() . . . . .	999
14.148.3.17 GetSupportedSchemaVersions() . . . . .	999
14.148.3.18 GetToolTip() . . . . .	999
14.148.3.19 GetVendorName() . . . . .	999
14.148.3.20 GetVersionGuid() . . . . .	1000
14.148.3.21 InvalidateNodes() . . . . .	1000
14.148.3.22 LoadXMLFromFile() . . . . .	1000
14.148.3.23 LoadXMLFromFileInject() . . . . .	1000
14.148.3.24 LoadXMLFromString() . . . . .	1000
14.148.3.25 LoadXMLFromStringInject() . . . . .	1001
14.148.3.26 LoadXMLFromZIPData() . . . . .	1001
14.148.3.27 LoadXMLFromZIPFile() . . . . .	1001
14.148.3.28 Poll() . . . . .	1001
14.148.4 Member Data Documentation . . . . .	1001
14.148.4.1 _Ptr . . . . .	1001
14.149 CNodeMapFactory::NodeStatistics_t Struct Reference . . . . .	1002
14.149.1 Member Data Documentation . . . . .	1002
14.149.1.1 NumLinks . . . . .	1002
14.149.1.2 NumNodes . . . . .	1002

---

---

14.149.1.3 NumProperties . . . . .	1002
14.149.1.4 NumStrings . . . . .	1002
14.150 ODevFileStreamBase< CharType, Traits > Class Template Reference . . . . .	1003
14.150.1 Member Typedef Documentation . . . . .	1004
14.150.1.1 filebuf_type . . . . .	1004
14.150.1.2 ios_type . . . . .	1004
14.150.1.3 ostream_type . . . . .	1004
14.150.2 Member Function Documentation . . . . .	1004
14.150.2.1 close() . . . . .	1004
14.150.2.2 is_open() . . . . .	1004
14.150.2.3 open() . . . . .	1004
14.150.2.4 rdbuf() . . . . .	1005
14.151 ODevFileStreamBuf< CharType, Traits > Class Template Reference . . . . .	1005
14.151.1 Constructor & Destructor Documentation . . . . .	1006
14.151.1.1 ODevFileStreamBuf() . . . . .	1006
14.151.1.2 ~ODevFileStreamBuf() . . . . .	1006
14.151.2 Member Function Documentation . . . . .	1006
14.151.2.1 close() . . . . .	1006
14.151.2.2 is_open() . . . . .	1006
14.151.2.3 open() . . . . .	1007
14.151.2.4 overflow() . . . . .	1007
14.151.2.5 sync() . . . . .	1007
14.151.2.6 xsputn() . . . . .	1007
14.152 PGMOOption Struct Reference . . . . .	1007
14.152.1 Detailed Description . . . . .	1008
14.152.2 Constructor & Destructor Documentation . . . . .	1008
14.152.2.1 PGMOOption() . . . . .	1008
14.152.3 Member Data Documentation . . . . .	1008
14.152.3.1 binaryFile . . . . .	1008
14.152.3.2 reserved . . . . .	1008
14.153 PNGOption Struct Reference . . . . .	1008
14.153.1 Detailed Description . . . . .	1009
14.153.2 Constructor & Destructor Documentation . . . . .	1009
14.153.2.1 PNGOption() . . . . .	1009
14.153.3 Member Data Documentation . . . . .	1009
14.153.3.1 compressionLevel . . . . .	1009
14.153.3.2 interlaced . . . . .	1009
14.153.3.3 reserved . . . . .	1010
14.154 PortNode Class Reference . . . . .	1010
14.154.1 Detailed Description . . . . .	1012
14.154.2 Constructor & Destructor Documentation . . . . .	1012
14.154.2.1 PortNode() [1/2] . . . . .	1012

---

14.154.2.2 PortNode() [2/2] . . . . .	1012
14.154.2.3 ~PortNode() . . . . .	1012
14.154.3 Member Function Documentation . . . . .	1012
14.154.3.1 CacheChunkData() . . . . .	1012
14.154.3.2 GetChunkID() . . . . .	1013
14.154.3.3 GetPortHandle() . . . . .	1013
14.154.3.4 GetSwapEndianess() . . . . .	1013
14.154.3.5 Read() . . . . .	1013
14.154.3.6 Replay() . . . . .	1013
14.154.3.7 SetPortImpl() . . . . .	1014
14.154.3.8 SetReference() [1/5] . . . . .	1014
14.154.3.9 SetReference() [2/5] . . . . .	1014
14.154.3.10 SetReference() [3/5] . . . . .	1014
14.154.3.11 SetReference() [4/5] . . . . .	1014
14.154.3.12 SetReference() [5/5] . . . . .	1015
14.154.3.13 StartRecording() . . . . .	1015
14.154.3.14 StopRecording() . . . . .	1015
14.154.3.15 Write() . . . . .	1015
14.155 PortRecorder Class Reference . . . . .	1016
14.155.1 Detailed Description . . . . .	1017
14.155.2 Constructor & Destructor Documentation . . . . .	1017
14.155.2.1 PortRecorder() . . . . .	1017
14.155.2.2 ~PortRecorder() . . . . .	1017
14.155.3 Member Function Documentation . . . . .	1017
14.155.3.1 GetAccessMode() . . . . .	1018
14.155.3.2 Read() . . . . .	1018
14.155.3.3 Replay() . . . . .	1018
14.155.3.4 SetReference() . . . . .	1018
14.155.3.5 StartRecording() . . . . .	1019
14.155.3.6 StopRecording() . . . . .	1019
14.155.3.7 Write() . . . . .	1019
14.156 PortReplay Class Reference . . . . .	1020
14.156.1 Detailed Description . . . . .	1021
14.156.2 Constructor & Destructor Documentation . . . . .	1021
14.156.2.1 PortReplay() . . . . .	1021
14.156.2.2 ~PortReplay() . . . . .	1021
14.156.3 Member Function Documentation . . . . .	1021
14.156.3.1 GetAccessMode() . . . . .	1022
14.156.3.2 GetPortReplayHandle() . . . . .	1022
14.156.3.3 Read() . . . . .	1022
14.156.3.4 Replay() . . . . .	1022
14.156.3.5 SetReference() . . . . .	1023

---

14.156.3.6 Write() . . . . .	1023
14.157 PPMOption Struct Reference . . . . .	1023
14.157.1 Detailed Description . . . . .	1023
14.157.2 Constructor & Destructor Documentation . . . . .	1024
14.157.2.1 PPMOption() . . . . .	1024
14.157.3 Member Data Documentation . . . . .	1024
14.157.3.1 binaryFile . . . . .	1024
14.157.3.2 reserved . . . . .	1024
14.158 RegisterNode Class Reference . . . . .	1024
14.158.1 Detailed Description . . . . .	1026
14.158.2 Constructor & Destructor Documentation . . . . .	1026
14.158.2.1 RegisterNode() [1/2] . . . . .	1026
14.158.2.2 RegisterNode() [2/2] . . . . .	1026
14.158.2.3 ~RegisterNode() . . . . .	1026
14.158.3 Member Function Documentation . . . . .	1026
14.158.3.1 Get() . . . . .	1027
14.158.3.2 GetAddress() . . . . .	1028
14.158.3.3 GetLength() . . . . .	1028
14.158.3.4 Set() . . . . .	1028
14.158.3.5 SetReference() . . . . .	1029
14.159 SingleChunkData_t Struct Reference . . . . .	1029
14.159.1 Member Data Documentation . . . . .	1029
14.159.1.1 ChunkID . . . . .	1029
14.159.1.2 ChunkLength . . . . .	1029
14.159.1.3 ChunkOffset . . . . .	1029
14.160 SingleChunkDataStr_t Struct Reference . . . . .	1030
14.160.1 Member Data Documentation . . . . .	1030
14.160.1.1 ChunkID . . . . .	1030
14.160.1.2 ChunkLength . . . . .	1030
14.160.1.3 ChunkOffset . . . . .	1030
14.161 SpinTestCamera Class Reference . . . . .	1031
14.162 SpinVideo Class Reference . . . . .	1031
14.162.1 Detailed Description . . . . .	1032
14.162.2 Constructor & Destructor Documentation . . . . .	1032
14.162.2.1 SpinVideo() . . . . .	1032
14.162.2.2 ~SpinVideo() . . . . .	1032
14.162.3 Member Function Documentation . . . . .	1032
14.162.3.1 Append() . . . . .	1032
14.162.3.2 Close() . . . . .	1033
14.162.3.3 Open() [1/3] . . . . .	1033
14.162.3.4 Open() [2/3] . . . . .	1033
14.162.3.5 Open() [3/3] . . . . .	1035

---

---

14.162.3.6 SetMaximumFileSize() . . . . .	1035
14.163 StringNode Class Reference . . . . .	1036
14.163.1 Detailed Description . . . . .	1037
14.163.2 Constructor & Destructor Documentation . . . . .	1038
14.163.2.1 StringNode() [1/2] . . . . .	1038
14.163.2.2 StringNode() [2/2] . . . . .	1038
14.163.2.3 ~StringNode() . . . . .	1038
14.163.3 Member Function Documentation . . . . .	1038
14.163.3.1 GetMaxLength() . . . . .	1038
14.163.3.2 GetValue() . . . . .	1038
14.163.3.3 operator()() . . . . .	1039
14.163.3.4 operator*() . . . . .	1039
14.163.3.5 operator=() . . . . .	1039
14.163.3.6 SetReference() . . . . .	1039
14.163.3.7 SetValue() . . . . .	1039
14.164 StringRegNode Class Reference . . . . .	1040
14.164.1 Detailed Description . . . . .	1041
14.164.2 Constructor & Destructor Documentation . . . . .	1041
14.164.2.1 StringRegNode() [1/2] . . . . .	1042
14.164.2.2 StringRegNode() [2/2] . . . . .	1042
14.164.2.3 ~StringRegNode() . . . . .	1042
14.164.3 Member Function Documentation . . . . .	1042
14.164.3.1 SetReference() . . . . .	1042
14.165 System Class Reference . . . . .	1043
14.165.1 Detailed Description . . . . .	1044
14.165.2 Constructor & Destructor Documentation . . . . .	1045
14.165.2.1 ~System() . . . . .	1045
14.165.2.2 System() . . . . .	1045
14.165.3 Member Function Documentation . . . . .	1045
14.165.3.1 GetCameras() . . . . .	1045
14.165.3.2 GetInstance() . . . . .	1046
14.165.3.3 GetInterfaces() . . . . .	1046
14.165.3.4 GetLibraryVersion() . . . . .	1047
14.165.3.5 GetLoggingEventPriorityLevel() . . . . .	1047
14.165.3.6 GetTLNodeMap() . . . . .	1047
14.165.3.7 IsInUse() . . . . .	1048
14.165.3.8 RegisterEventHandler() . . . . .	1048
14.165.3.9 RegisterInterfaceEventHandler() . . . . .	1048
14.165.3.10 RegisterLoggingEventHandler() . . . . .	1049
14.165.3.11 ReleaseInstance() . . . . .	1049
14.165.3.12 SendActionCommand() . . . . .	1049
14.165.3.13 SetLoggingEventPriorityLevel() . . . . .	1050

---

14.165.3.14 UnregisterAllLoggingEventHandlers()	1051
14.165.3.15 UnregisterEventHandler()	1051
14.165.3.16 UnregisterInterfaceEventHandler()	1051
14.165.3.17 UnregisterLoggingEventHandler()	1051
14.165.3.18 UpdateCameras()	1052
14.165.3.19 UpdateInterfaceList()	1052
14.166 SystemEventHandler Class Reference	1053
14.166.1 Detailed Description	1054
14.166.2 Constructor & Destructor Documentation	1054
14.166.2.1 SystemEventHandler()	1054
14.166.2.2 ~SystemEventHandler()	1054
14.166.3 Member Function Documentation	1054
14.166.3.1 OnInterfaceArrival()	1054
14.166.3.2 OnInterfaceRemoval()	1055
14.166.3.3 operator=()	1055
14.167 SystemEventHandlerImpl Class Reference	1056
14.167.1 Constructor & Destructor Documentation	1057
14.167.1.1 SystemEventHandlerImpl()	1057
14.167.1.2 ~SystemEventHandlerImpl()	1057
14.167.2 Member Function Documentation	1057
14.167.2.1 LockEventHandlerMutex()	1057
14.167.2.2 OnInterfaceArrival()	1057
14.167.2.3 OnInterfaceRemoval()	1058
14.167.2.4 RegisterAllInterfaceEvents()	1058
14.167.2.5 RegisterInterfaceEventToSystem()	1058
14.167.2.6 UnlockEventHandlerMutex()	1058
14.167.2.7 UnregisterAllInterfaceEvents()	1058
14.167.2.8 UnregisterInterfaceEventFromSystem()	1059
14.168 SystemPtr Class Reference	1059
14.168.1 Detailed Description	1060
14.168.2 Constructor & Destructor Documentation	1060
14.168.2.1 SystemPtr() [1/4]	1060
14.168.2.2 SystemPtr() [2/4]	1060
14.168.2.3 SystemPtr() [3/4]	1060
14.168.2.4 SystemPtr() [4/4]	1060
14.168.2.5 ~SystemPtr()	1061
14.169 TIFFOption Struct Reference	1061
14.169.1 Detailed Description	1061
14.169.2 Member Enumeration Documentation	1061
14.169.2.1 CompressionMethod	1061
14.169.3 Constructor & Destructor Documentation	1062
14.169.3.1 TIFFOption()	1062

---

---

14.169.4 Member Data Documentation . . . . .	1062
14.169.4.1 compression . . . . .	1062
14.169.4.2 reserved . . . . .	1062
14.170 TransportLayerDevice Class Reference . . . . .	1063
14.170.1 Detailed Description . . . . .	1065
14.170.2 Constructor & Destructor Documentation . . . . .	1065
14.170.2.1 TransportLayerDevice() [1/2] . . . . .	1065
14.170.2.2 ~TransportLayerDevice() . . . . .	1065
14.170.2.3 TransportLayerDevice() [2/2] . . . . .	1065
14.170.3 Friends And Related Function Documentation . . . . .	1065
14.170.3.1 CameraBase . . . . .	1065
14.170.3.2 CameralInternal . . . . .	1066
14.170.3.3 ICameraBase . . . . .	1066
14.170.4 Member Data Documentation . . . . .	1066
14.170.4.1 DeviceAccessStatus . . . . .	1066
14.170.4.2 DeviceCurrentSpeed . . . . .	1066
14.170.4.3 DeviceDisplayName . . . . .	1066
14.170.4.4 DeviceDriverVersion . . . . .	1066
14.170.4.5 DeviceEndianessMechanism . . . . .	1067
14.170.4.6 DeviceID . . . . .	1067
14.170.4.7 DeviceInstanceld . . . . .	1067
14.170.4.8 DevicelsUpdater . . . . .	1067
14.170.4.9 DeviceLinkSpeed . . . . .	1067
14.170.4.10 DeviceLocation . . . . .	1067
14.170.4.11 DeviceModelName . . . . .	1068
14.170.4.12 DeviceMulticastMonitorMode . . . . .	1068
14.170.4.13 DeviceSerialNumber . . . . .	1068
14.170.4.14 DeviceType . . . . .	1068
14.170.4.15 DeviceU3VProtocol . . . . .	1068
14.170.4.16 DeviceUserID . . . . .	1068
14.170.4.17 DeviceVendorName . . . . .	1069
14.170.4.18 DeviceVersion . . . . .	1069
14.170.4.19 GenICamXMLlocation . . . . .	1069
14.170.4.20 GenICamXMLPath . . . . .	1069
14.170.4.21 GevCCP . . . . .	1069
14.170.4.22 GevDeviceAutoForceIP . . . . .	1069
14.170.4.23 GevDeviceDiscoverMaximumPacketSize . . . . .	1070
14.170.4.24 GevDeviceForceGateway . . . . .	1070
14.170.4.25 GevDeviceForceIP . . . . .	1070
14.170.4.26 GevDeviceForceIPAddress . . . . .	1070
14.170.4.27 GevDeviceForceSubnetMask . . . . .	1070
14.170.4.28 GevDeviceGateway . . . . .	1070

---

---

14.170.4.29 GevDeviceIPAddress . . . . .	1071
14.170.4.30 GevDeviceIsWrongSubnet . . . . .	1071
14.170.4.31 GevDeviceMACAddress . . . . .	1071
14.170.4.32 GevDeviceMaximumPacketSize . . . . .	1071
14.170.4.33 GevDeviceMaximumRetryCount . . . . .	1071
14.170.4.34 GevDeviceModelsBigEndian . . . . .	1071
14.170.4.35 GevDevicePort . . . . .	1072
14.170.4.36 GevDeviceReadAndWriteTimeout . . . . .	1072
14.170.4.37 GevDeviceSubnetMask . . . . .	1072
14.170.4.38 GevVersionMajor . . . . .	1072
14.170.4.39 GevVersionMinor . . . . .	1072
14.170.4.40 GUIXMLLocation . . . . .	1072
14.170.4.41 GUIXMLPath . . . . .	1073
<b>14.171 TransportLayerInterface Class Reference . . . . .</b>	<b>1073</b>
<b>14.171.1 Detailed Description . . . . .</b>	<b>1075</b>
<b>14.171.2 Constructor &amp; Destructor Documentation . . . . .</b>	<b>1075</b>
14.171.2.1 TransportLayerInterface() [1/2] . . . . .	1075
14.171.2.2 ~TransportLayerInterface() . . . . .	1076
14.171.2.3 TransportLayerInterface() [2/2] . . . . .	1076
<b>14.171.3 Friends And Related Function Documentation . . . . .</b>	<b>1076</b>
14.171.3.1 IInterface . . . . .	1076
14.171.3.2 Interface . . . . .	1076
14.171.3.3 InterfaceInternal . . . . .	1076
<b>14.171.4 Member Data Documentation . . . . .</b>	<b>1076</b>
14.171.4.1 ActionCommand . . . . .	1076
14.171.4.2 DeviceAccessStatus . . . . .	1077
14.171.4.3 DeviceCount . . . . .	1077
14.171.4.4 DeviceID . . . . .	1077
14.171.4.5 DeviceModelName . . . . .	1077
14.171.4.6 DeviceSelector . . . . .	1077
14.171.4.7 DeviceSerialNumber . . . . .	1077
14.171.4.8 DeviceUnlock . . . . .	1078
14.171.4.9 DeviceUpdateList . . . . .	1078
14.171.4.10 DeviceVendorName . . . . .	1078
14.171.4.11 FilterDriverStatus . . . . .	1078
14.171.4.12 GevActionDeviceKey . . . . .	1078
14.171.4.13 GevActionGroupKey . . . . .	1078
14.171.4.14 GevActionGroupMask . . . . .	1079
14.171.4.15 GevActionTime . . . . .	1079
14.171.4.16 GevDeviceAutoForceIP . . . . .	1079
14.171.4.17 GevDeviceForceGateway . . . . .	1079
14.171.4.18 GevDeviceForceIP . . . . .	1079

---

14.171.4.19 GevDeviceForceIPAddress . . . . .	1079
14.171.4.20 GevDeviceForceSubnetMask . . . . .	1080
14.171.4.21 GevDeviceGateway . . . . .	1080
14.171.4.22 GevDeviceIPAddress . . . . .	1080
14.171.4.23 GevDeviceMACAddress . . . . .	1080
14.171.4.24 GevDeviceSubnetMask . . . . .	1080
14.171.4.25 GevInterfaceGateway . . . . .	1080
14.171.4.26 GevInterfaceGatewaySelector . . . . .	1081
14.171.4.27 GevInterfaceMACAddress . . . . .	1081
14.171.4.28 GevInterfaceMTU . . . . .	1081
14.171.4.29 GevInterfaceReceiveLinkSpeed . . . . .	1081
14.171.4.30 GevInterfaceSubnetIPAddress . . . . .	1081
14.171.4.31 GevInterfaceSubnetMask . . . . .	1081
14.171.4.32 GevInterfaceSubnetSelector . . . . .	1082
14.171.4.33 GevInterfaceTransmitLinkSpeed . . . . .	1082
14.171.4.34 HostAdapterDriverVersion . . . . .	1082
14.171.4.35 HostAdapterName . . . . .	1082
14.171.4.36 HostAdapterVendor . . . . .	1082
14.171.4.37 IncompatibleDeviceCount . . . . .	1082
14.171.4.38 IncompatibleDeviceID . . . . .	1083
14.171.4.39 IncompatibleDeviceModelName . . . . .	1083
14.171.4.40 IncompatibleDeviceSelector . . . . .	1083
14.171.4.41 IncompatibleDeviceVendorName . . . . .	1083
14.171.4.42 IncompatibleGevDeviceIPAddress . . . . .	1083
14.171.4.43 IncompatibleGevDeviceMACAddress . . . . .	1083
14.171.4.44 IncompatibleGevDeviceSubnetMask . . . . .	1084
14.171.4.45 InterfaceDisplayName . . . . .	1084
14.171.4.46 InterfaceID . . . . .	1084
14.171.4.47 InterfaceType . . . . .	1084
14.171.4.48 POEStatus . . . . .	1084
<b>14.172 TransportLayerStream Class Reference . . . . .</b>	<b>1084</b>
<b>14.172.1 Detailed Description . . . . .</b>	<b>1086</b>
<b>14.172.2 Constructor &amp; Destructor Documentation . . . . .</b>	<b>1086</b>
<b>14.172.2.1 TransportLayerStream() [1/2] . . . . .</b>	<b>1086</b>
<b>14.172.2.2 ~TransportLayerStream() . . . . .</b>	<b>1086</b>
<b>14.172.2.3 TransportLayerStream() [2/2] . . . . .</b>	<b>1086</b>
<b>14.172.3 Friends And Related Function Documentation . . . . .</b>	<b>1087</b>
<b>14.172.3.1 CameraBase . . . . .</b>	<b>1087</b>
<b>14.172.3.2 CameralInternal . . . . .</b>	<b>1087</b>
<b>14.172.3.3 ICameraBase . . . . .</b>	<b>1087</b>
<b>14.172.4 Member Data Documentation . . . . .</b>	<b>1087</b>
<b>14.172.4.1 GevFailedPacketCount . . . . .</b>	<b>1087</b>

---

---

14.172.4.2 GevMaximumNumberResendRequests . . . . .	1087
14.172.4.3 GevPacketResendMode . . . . .	1087
14.172.4.4 GevPacketResendTimeout . . . . .	1088
14.172.4.5 GevResendPacketCount . . . . .	1088
14.172.4.6 GevResendRequestCount . . . . .	1088
14.172.4.7 GevTotalPacketCount . . . . .	1088
14.172.4.8 StreamAnnounceBufferMinimum . . . . .	1088
14.172.4.9 StreamAnnouncedBufferCount . . . . .	1088
14.172.4.10 StreamBlockTransferSize . . . . .	1089
14.172.4.11 StreamBufferAlignment . . . . .	1089
14.172.4.12 StreamBufferCountManual . . . . .	1089
14.172.4.13 StreamBufferCountMax . . . . .	1089
14.172.4.14 StreamBufferCountMode . . . . .	1089
14.172.4.15 StreamBufferCountResult . . . . .	1089
14.172.4.16 StreamBufferHandlingMode . . . . .	1090
14.172.4.17 StreamChunkCountMaximum . . . . .	1090
14.172.4.18 StreamCRCCheckEnable . . . . .	1090
14.172.4.19 StreamDeliveredFrameCount . . . . .	1090
14.172.4.20 StreamFailedBufferCount . . . . .	1090
14.172.4.21 StreamID . . . . .	1090
14.172.4.22 StreamInputBufferCount . . . . .	1091
14.172.4.23 StreamIsGrabbing . . . . .	1091
14.172.4.24 StreamLostFrameCount . . . . .	1091
14.172.4.25 StreamOutputBufferCount . . . . .	1091
14.172.4.26 StreamStartedFrameCount . . . . .	1091
14.172.4.27 StreamType . . . . .	1092
14.173 TransportLayerSystem Class Reference . . . . .	1092
14.173.1 Detailed Description . . . . .	1093
14.173.2 Constructor & Destructor Documentation . . . . .	1093
14.173.2.1 TransportLayerSystem() [1/2] . . . . .	1094
14.173.2.2 ~TransportLayerSystem() . . . . .	1094
14.173.2.3 TransportLayerSystem() [2/2] . . . . .	1094
14.173.3 Friends And Related Function Documentation . . . . .	1094
14.173.3.1 ISystem . . . . .	1094
14.173.3.2 System . . . . .	1094
14.173.3.3 SystemPtrInternal . . . . .	1094
14.173.4 Member Data Documentation . . . . .	1094
14.173.4.1 EnumerateGEVInterfaces . . . . .	1095
14.173.4.2 GenTLSFNCVersionMajor . . . . .	1095
14.173.4.3 GenTLSFNCVersionMinor . . . . .	1095
14.173.4.4 GenTLSFNCVersionSubMinor . . . . .	1095
14.173.4.5 GenTLVersionMajor . . . . .	1095

---

---

14.173.4.6 GenTLVersionMinor . . . . .	1095
14.173.4.7 GevInterfaceDefaultGateway . . . . .	1096
14.173.4.8 GevInterfaceDefaultIPAddress . . . . .	1096
14.173.4.9 GevInterfaceDefaultSubnetMask . . . . .	1096
14.173.4.10 GevInterfaceMACAddress . . . . .	1096
14.173.4.11 GevVersionMajor . . . . .	1096
14.173.4.12 GevVersionMinor . . . . .	1096
14.173.4.13 InterfaceDisplayName . . . . .	1097
14.173.4.14 InterfaceID . . . . .	1097
14.173.4.15 InterfaceSelector . . . . .	1097
14.173.4.16 InterfaceUpdateList . . . . .	1097
14.173.4.17 TLDisplayName . . . . .	1097
14.173.4.18 TLFileName . . . . .	1097
14.173.4.19 TLID . . . . .	1098
14.173.4.20 TLModelName . . . . .	1098
14.173.4.21 TLPath . . . . .	1098
14.173.4.22 TLType . . . . .	1098
14.173.4.23 TLVendorName . . . . .	1098
14.173.4.24 TLVersion . . . . .	1098
14.174 U3V_CHUNK_TRAILER Struct Reference . . . . .	1099
14.174.1 Detailed Description . . . . .	1099
14.174.2 Member Data Documentation . . . . .	1099
14.174.2.1 ChunkID . . . . .	1099
14.174.2.2 ChunkLength . . . . .	1099
14.175 U3V_COMMAND_HEADER Struct Reference . . . . .	1099
14.175.1 Detailed Description . . . . .	1100
14.175.2 Member Data Documentation . . . . .	1100
14.175.2.1 CommandId . . . . .	1100
14.175.2.2 Flags . . . . .	1100
14.175.2.3 Length . . . . .	1100
14.175.2.4 Prefix . . . . .	1100
14.175.2.5 ReqId . . . . .	1100
14.176 U3V_EVENT_DATA Struct Reference . . . . .	1100
14.176.1 Detailed Description . . . . .	1101
14.176.2 Member Data Documentation . . . . .	1101
14.176.2.1 EventId . . . . .	1101
14.176.2.2 Reserved . . . . .	1101
14.176.2.3 Timestamp . . . . .	1101
14.177 U3V_EVENT_MESSAGE Struct Reference . . . . .	1101
14.177.1 Detailed Description . . . . .	1102
14.177.2 Member Data Documentation . . . . .	1102
14.177.2.1 CommandHeader . . . . .	1102

---

14.177.2.2 EventData . . . . .	1102
14.178 ValueNode Class Reference . . . . .	1102
14.178.1 Detailed Description . . . . .	1103
14.178.2 Constructor & Destructor Documentation . . . . .	1103
14.178.2.1 ValueNode() [1/2] . . . . .	1104
14.178.2.2 ValueNode() [2/2] . . . . .	1104
14.178.2.3 ~ValueNode() . . . . .	1104
14.178.3 Member Function Documentation . . . . .	1104
14.178.3.1 FromString() . . . . .	1104
14.178.3.2 GetNode() . . . . .	1104
14.178.3.3 IsValueCacheValid() . . . . .	1105
14.178.3.4 SetReference() . . . . .	1105
14.178.3.5 ToString() . . . . .	1105
14.179 Version_t Struct Reference . . . . .	1105
14.179.1 Detailed Description . . . . .	1106
14.179.2 Member Data Documentation . . . . .	1106
14.179.2.1 Major . . . . .	1106
14.179.2.2 Minor . . . . .	1106
14.179.2.3 SubMinor . . . . .	1106
<b>15 File Documentation</b>	<b>1107</b>
15.1 doc/spindocs/C++/GettingStarted.dox File Reference . . . . .	1107
15.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference . . . . .	1107
15.3 doc/spindocs/shared/Benefits.dox File Reference . . . . .	1107
15.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference . . . . .	1107
15.5 doc/spindocs/shared/GenICamGenTL.dox File Reference . . . . .	1107
15.6 doc/spindocs/shared/Licensing.dox File Reference . . . . .	1107
15.7 include/AdapterConfig.h File Reference . . . . .	1107
15.7.1 Macro Definition Documentation . . . . .	1109
15.7.1.1 ADAPTERCONFIG_API . . . . .	1109
15.8 include/AVIRecorder.h File Reference . . . . .	1109
15.9 include/BasePtr.h File Reference . . . . .	1110
15.10 include/Camera.h File Reference . . . . .	1111
15.11 include/CameraBase.h File Reference . . . . .	1111
15.12 include/CameraDefs.h File Reference . . . . .	1112
15.13 include/CameraList.h File Reference . . . . .	1144
15.14 include/CameraPtr.h File Reference . . . . .	1145
15.15 include/ChunkData.h File Reference . . . . .	1145
15.16 include/ChunkDataInference.h File Reference . . . . .	1146
15.17 include/DeviceArrivalEventHandler.h File Reference . . . . .	1147
15.18 include/DeviceEventHandler.h File Reference . . . . .	1148
15.19 include/DeviceRemovalEventHandler.h File Reference . . . . .	1149

---

15.20 include/EventHandler.h File Reference . . . . .	1150
15.21 include/Exception.h File Reference . . . . .	1150
15.22 include/Image.h File Reference . . . . .	1151
15.23 include/ImageEventHandler.h File Reference . . . . .	1152
15.24 include/ImagePtr.h File Reference . . . . .	1152
15.25 include/ImageStatistics.h File Reference . . . . .	1153
15.26 include/ImageUtility.h File Reference . . . . .	1154
15.27 include/ImageUtilityHeatmap.h File Reference . . . . .	1155
15.28 include/ImageUtilityPolarization.h File Reference . . . . .	1156
15.29 include/Interface.h File Reference . . . . .	1157
15.30 include/Interface/ICameraBase.h File Reference . . . . .	1157
15.31 include/Interface/ICameraList.h File Reference . . . . .	1158
15.32 include/Interface/IChunkData.h File Reference . . . . .	1159
15.33 include/Interface/IDeviceArrivalEventHandler.h File Reference . . . . .	1160
15.34 include/Interface/IDeviceEventHandler.h File Reference . . . . .	1161
15.35 include/Interface/IDeviceRemovalEventHandler.h File Reference . . . . .	1162
15.36 include/Interface/IImage.h File Reference . . . . .	1163
15.37 include/Interface/IImageEventHandler.h File Reference . . . . .	1163
15.38 include/Interface/IImageStatistics.h File Reference . . . . .	1164
15.39 include/Interface/IInterface.h File Reference . . . . .	1165
15.40 include/Interface/IInterfaceArrivalEventHandler.h File Reference . . . . .	1166
15.41 include/Interface/IInterfaceEventHandler.h File Reference . . . . .	1167
15.42 include/Interface/IInterfaceList.h File Reference . . . . .	1167
15.43 include/Interface/IInterfaceRemovalEventHandler.h File Reference . . . . .	1168
15.44 include/Interface/ILoggingEventHandler.h File Reference . . . . .	1169
15.45 include/Interface/IStream.h File Reference . . . . .	1169
15.46 include/Interface/ISystem.h File Reference . . . . .	1170
15.47 include/Interface/ISystemEventHandler.h File Reference . . . . .	1171
15.48 include/InterfaceArrivalEventHandler.h File Reference . . . . .	1172
15.49 include/InterfaceEventHandler.h File Reference . . . . .	1173
15.50 include/InterfaceList.h File Reference . . . . .	1174
15.51 include/InterfacePtr.h File Reference . . . . .	1174
15.52 include/InterfaceRemovalEventHandler.h File Reference . . . . .	1175
15.53 include/LoggingEventData.h File Reference . . . . .	1176
15.54 include/LoggingEventDataPtr.h File Reference . . . . .	1176
15.55 include/LoggingEventHandler.h File Reference . . . . .	1177
15.56 include/SpinGenApi/Autovector.h File Reference . . . . .	1178
15.57 include/SpinGenApi/Base.h File Reference . . . . .	1179
15.58 include/SpinGenApi/BooleanNode.h File Reference . . . . .	1180
15.59 include/SpinGenApi/CategoryNode.h File Reference . . . . .	1181
15.60 include/SpinGenApi/ChunkAdapter.h File Reference . . . . .	1182
15.61 include/SpinGenApi/ChunkAdapterDcam.h File Reference . . . . .	1183

---

15.62 include/SpinGenApi/ChunkAdapterGeneric.h File Reference . . . . .	1184
15.63 include/SpinGenApi/ChunkAdapterGEV.h File Reference . . . . .	1185
15.64 include/SpinGenApi/ChunkAdapterU3V.h File Reference . . . . .	1186
15.65 include/SpinGenApi/ChunkPort.h File Reference . . . . .	1187
15.66 include/SpinGenApi/CommandNode.h File Reference . . . . .	1187
15.67 include/SpinGenApi/Compatibility.h File Reference . . . . .	1188
15.67.1 Macro Definition Documentation . . . . .	1188
15.67.1.1 FMT_I64 . . . . .	1188
15.68 include/SpinGenApi/Container.h File Reference . . . . .	1189
15.69 include/SpinGenApi/Counter.h File Reference . . . . .	1189
15.70 include/SpinGenApi/EnumClasses.h File Reference . . . . .	1190
15.71 include/SpinGenApi/EnumEntryNode.h File Reference . . . . .	1191
15.72 include/SpinGenApi/EnumNode.h File Reference . . . . .	1192
15.73 include/SpinGenApi/EnumNodeT.h File Reference . . . . .	1193
15.74 include/SpinGenApi/EventAdapter.h File Reference . . . . .	1193
15.75 include/SpinGenApi/EventAdapter1394.h File Reference . . . . .	1194
15.76 include/SpinGenApi/EventAdapterGeneric.h File Reference . . . . .	1195
15.77 include/SpinGenApi/EventAdapterGEV.h File Reference . . . . .	1195
15.78 include/SpinGenApi/EventAdapterU3V.h File Reference . . . . .	1196
15.79 include/SpinGenApi/EventPort.h File Reference . . . . .	1197
15.80 include/SpinGenApi/Filestream.h File Reference . . . . .	1198
15.81 include/SpinGenApi/FloatNode.h File Reference . . . . .	1199
15.82 include/SpinGenApi/FloatRegNode.h File Reference . . . . .	1200
15.83 include/SpinGenApi/GCBase.h File Reference . . . . .	1201
15.84 include/SpinGenApi/GCString.h File Reference . . . . .	1201
15.84.1 Macro Definition Documentation . . . . .	1202
15.84.1.1 GCSTRING_NPOS . . . . .	1202
15.84.2 Function Documentation . . . . .	1202
15.84.2.1 operator<<() . . . . .	1203
15.84.2.2 operator>>() . . . . .	1203
15.85 include/SpinGenApi/GCStringVector.h File Reference . . . . .	1203
15.86 include/SpinGenApi/GCSynch.h File Reference . . . . .	1204
15.87 include/SpinGenApi/GCTypes.h File Reference . . . . .	1205
15.87.1 Macro Definition Documentation . . . . .	1206
15.87.1.1 __STDC_CONSTANT_MACROS . . . . .	1206
15.87.1.2 __STDC_LIMIT_MACROS . . . . .	1206
15.87.1.3 GC_INT32_MAX . . . . .	1206
15.87.1.4 GC_INT32_MIN . . . . .	1206
15.87.1.5 GC_INT64_MAX . . . . .	1206
15.87.1.6 GC_INT64_MIN . . . . .	1207
15.87.1.7 GC_INT8_MAX . . . . .	1207
15.87.1.8 GC_INT8_MIN . . . . .	1207

---

15.87.1.9 GC_UINT32_MAX . . . . .	1207
15.87.1.10 GC_UINT64_MAX . . . . .	1207
15.87.1.11 GC_UINT8_MAX . . . . .	1207
15.88 include/SpinGenApi/GCUtilities.h File Reference . . . . .	1208
15.88.1 Macro Definition Documentation . . . . .	1209
15.88.1.1 __ERR__ . . . . .	1210
15.88.1.2 __LINE_STR__ . . . . .	1210
15.88.1.3 __LOCATION__ . . . . .	1210
15.88.1.4 __OUTPUT_FORMATER__ . . . . .	1210
15.88.1.5 __TODO__ . . . . .	1210
15.88.1.6 __WARN__ . . . . .	1210
15.88.1.7 __TO_STRING__ . . . . .	1210
15.88.1.8 EXPAND_TO_STRINGISE . . . . .	1211
15.88.1.9 GC_COUNTOF . . . . .	1211
15.88.1.10 GENICAM_DEPRECATED . . . . .	1211
15.88.1.11 GENICAM_UNUSED . . . . .	1211
15.88.1.12 USE_TEMP_CACHE_FILE [1/2] . . . . .	1211
15.88.1.13 USE_TEMP_CACHE_FILE [2/2] . . . . .	1211
15.89 include/SpinGenApi/IBoolean.h File Reference . . . . .	1212
15.90 include/SpinGenApi/ICategory.h File Reference . . . . .	1213
15.91 include/SpinGenApi/IChunkPort.h File Reference . . . . .	1214
15.91.1 Macro Definition Documentation . . . . .	1215
15.91.1.1 CHUNK_BASE_ADDRESS_REGISTER . . . . .	1215
15.91.1.2 CHUNK_BASE_ADDRESS_REGISTER_LEN . . . . .	1215
15.91.1.3 CHUNK_LENGTH_REGISTER . . . . .	1215
15.91.1.4 CHUNK_LENGTH_REGISTER_LEN . . . . .	1215
15.92 include/SpinGenApi/ICommand.h File Reference . . . . .	1216
15.93 include/SpinGenApi/IDestroy.h File Reference . . . . .	1217
15.94 include/SpinGenApi/IDeviceInfo.h File Reference . . . . .	1218
15.95 include/SpinGenApi/IEnumEntry.h File Reference . . . . .	1219
15.96 include/SpinGenApi/IEnumeration.h File Reference . . . . .	1220
15.97 include/SpinGenApi/IEnumerationT.h File Reference . . . . .	1221
15.98 include/SpinGenApi/IFloat.h File Reference . . . . .	1222
15.99 include/SpinGenApi/IInteger.h File Reference . . . . .	1224
15.100 include/SpinGenApi/INode.h File Reference . . . . .	1225
15.101 include/SpinGenApi/INodeMap.h File Reference . . . . .	1228
15.102 include/SpinGenApi/INodeMapDyn.h File Reference . . . . .	1230
15.103 include/SpinGenApi/IntegerNode.h File Reference . . . . .	1231
15.104 include/SpinGenApi/IntRegNode.h File Reference . . . . .	1232
15.105 include/SpinGenApi/IPort.h File Reference . . . . .	1233
15.106 include/SpinGenApi/IPortConstruct.h File Reference . . . . .	1234
15.107 include/SpinGenApi/IPortRecorder.h File Reference . . . . .	1235

15.108 include/SpinGenApi/IRegister.h File Reference . . . . .	1236
15.109 include/SpinGenApi/ISelector.h File Reference . . . . .	1237
15.110 include/SpinGenApi/ISelectorDigit.h File Reference . . . . .	1238
15.111 include/SpinGenApi/IString.h File Reference . . . . .	1239
15.112 include/SpinGenApi/IValue.h File Reference . . . . .	1240
15.113 include/SpinGenApi/Node.h File Reference . . . . .	1241
15.114 include/SpinGenApi/NodeCallback.h File Reference . . . . .	1242
15.115 include/SpinGenApi/NodeCallbackImpl.h File Reference . . . . .	1244
15.116 include/SpinGenApi/NodeMap.h File Reference . . . . .	1244
15.117 include/SpinGenApi/NodeMapFactory.h File Reference . . . . .	1245
15.118 include/SpinGenApi/NodeMapRef.h File Reference . . . . .	1246
15.119 include/SpinGenApi/Persistence.h File Reference . . . . .	1247
15.120 include/SpinGenApi/Pointer.h File Reference . . . . .	1248
15.121 include/SpinGenApi/PortImpl.h File Reference . . . . .	1250
15.122 include/SpinGenApi/PortNode.h File Reference . . . . .	1251
15.123 include/SpinGenApi/PortRecorder.h File Reference . . . . .	1252
15.124 include/SpinGenApi/PortReplay.h File Reference . . . . .	1252
15.125 include/SpinGenApi/PortWriteList.h File Reference . . . . .	1253
15.126 include/SpinGenApi/Reference.h File Reference . . . . .	1254
15.127 include/SpinGenApi/RegisterNode.h File Reference . . . . .	1255
15.128 include/SpinGenApi/RegisterPortImpl.h File Reference . . . . .	1256
15.129 include/SpinGenApi/SelectorSet.h File Reference . . . . .	1256
15.130 include/SpinGenApi/SpinnakerGenApi.h File Reference . . . . .	1257
15.131 include/SpinGenApi/SpinTestCamera.h File Reference . . . . .	1257
15.132 include/SpinGenApi/StringNode.h File Reference . . . . .	1258
15.133 include/SpinGenApi/StringRegNode.h File Reference . . . . .	1259
15.134 include/SpinGenApi/StructPort.h File Reference . . . . .	1259
15.135 include/SpinGenApi/Synch.h File Reference . . . . .	1260
15.136 include/SpinGenApi/Types.h File Reference . . . . .	1261
15.136.1 Macro Definition Documentation . . . . .	1264
15.136.1.1 _UndefinedRepresentation . . . . .	1264
15.136.1.2 interface . . . . .	1264
15.137 include/SpinGenApi/ValueNode.h File Reference . . . . .	1264
15.138 include/Spinnaker.h File Reference . . . . .	1265
15.139 include/SpinnakerDefs.h File Reference . . . . .	1265
15.140 include/SpinnakerPlatform.h File Reference . . . . .	1269
15.140.1 Macro Definition Documentation . . . . .	1269
15.140.1.1 SPINNAKER_API . . . . .	1270
15.140.1.2 SPINNAKER_API_ABSTRACT . . . . .	1270
15.140.1.3 SPINNAKER_LOCAL . . . . .	1270
15.141 include/SpinUpdate.h File Reference . . . . .	1270
15.141.1 Macro Definition Documentation . . . . .	1271

---

15.141.1.1 SPINUPDATE_API . . . . .	1271
15.141.2 Function Documentation . . . . .	1271
15.141.2.1 GetErrorMessage() . . . . .	1271
15.141.2.2 SetMessageCallback() . . . . .	1271
15.141.2.3 SetProgressCallback() . . . . .	1271
15.141.2.4 UpdateFirmware() . . . . .	1271
15.141.2.5 UpdateFirmwareConsole() . . . . .	1271
15.141.2.6 UpdateFirmwareGUI() . . . . .	1272
15.141.3 Variable Documentation . . . . .	1272
15.141.3.1 UpdatorMessageCallback . . . . .	1272
15.141.3.2 UpdatorProgressCallback . . . . .	1272
15.142 include/SpinVideo.h File Reference . . . . .	1273
15.143 include/SpinVideoDefs.h File Reference . . . . .	1274
15.144 include/System.h File Reference . . . . .	1275
15.144.1 Macro Definition Documentation . . . . .	1275
15.144.1.1 FLIR_SPINNAKER_VERSION_BUILD . . . . .	1275
15.144.1.2 FLIR_SPINNAKER_VERSION_MAJOR . . . . .	1276
15.144.1.3 FLIR_SPINNAKER_VERSION_MINOR . . . . .	1276
15.144.1.4 FLIR_SPINNAKER_VERSION_TYPE . . . . .	1276
15.145 include/SystemEventHandler.h File Reference . . . . .	1276
15.146 include/SystemPtr.h File Reference . . . . .	1277
15.147 include/TransportLayerDefs.h File Reference . . . . .	1277
15.148 include/TransportLayerDevice.h File Reference . . . . .	1279
15.149 include/TransportLayerInterface.h File Reference . . . . .	1280
15.150 include/TransportLayerStream.h File Reference . . . . .	1280
15.151 include/TransportLayerSystem.h File Reference . . . . .	1281
15.152 src/Acquisition/Acquisition.cpp File Reference . . . . .	1281
15.152.1 Function Documentation . . . . .	1282
15.152.1.1 AcquireImages() . . . . .	1282
15.152.1.2 main() . . . . .	1282
15.152.1.3 PrintDeviceInfo() . . . . .	1282
15.152.1.4 RunSingleCamera() . . . . .	1282
15.153 src/Acquisition/resource.h File Reference . . . . .	1284
15.154 src/AcquisitionMultipleCameraRecovery/resource.h File Reference . . . . .	1284
15.155 src/AcquisitionMultipleThread/resource.h File Reference . . . . .	1284
15.156 src/ActionCommand/resource.h File Reference . . . . .	1284
15.157 src/BufferHandling/resource.h File Reference . . . . .	1284
15.158 src/ChunkData/resource.h File Reference . . . . .	1284
15.159 src/CounterAndTimer/resource.h File Reference . . . . .	1284
15.160 src/DeviceEvents/resource.h File Reference . . . . .	1284
15.161 src/Enumeration/resource.h File Reference . . . . .	1284
15.162 src/Enumeration_QuickSpin/resource.h File Reference . . . . .	1284

---

---

15.163 src/EnumerationEvents/resource.h File Reference . . . . .	1284
15.164 src/ExceptionHandling/resource.h File Reference . . . . .	1284
15.165 src/Exposure/resource.h File Reference . . . . .	1284
15.166 src/Exposure_QuickSpin/resource.h File Reference . . . . .	1284
15.167 src/FileAccess_QuickSpin/resource.h File Reference . . . . .	1284
15.168 src/GigEVisionPerformance/resource.h File Reference . . . . .	1284
15.169 src/HighDynamicRange/resource.h File Reference . . . . .	1284
15.170 src/GenTLInfo_QuickSpin/resource.h File Reference . . . . .	1284
15.171 src/ImageEvents/resource.h File Reference . . . . .	1284
15.172 src/ImageFormatControl/resource.h File Reference . . . . .	1284
15.173 src/ImageFormatControl_QuickSpin/resource.h File Reference . . . . .	1284
15.174 src/Inference/resource.h File Reference . . . . .	1284
15.175 src/Logging/resource.h File Reference . . . . .	1284
15.176 src/LogicBlock/resource.h File Reference . . . . .	1284
15.177 src/LookupTable/resource.h File Reference . . . . .	1284
15.178 src/NodeMapCallback/resource.h File Reference . . . . .	1284
15.179 src/NodeMapInfo/resource.h File Reference . . . . .	1284
15.180 src/Polarization/resource.h File Reference . . . . .	1284
15.181 src/SaveToAvi/resource.h File Reference . . . . .	1284
15.182 src/Sequencer/resource.h File Reference . . . . .	1284
15.183 src/SerialRxTx/resource.h File Reference . . . . .	1284
15.184 src/Trigger/resource.h File Reference . . . . .	1284
15.185 src/Trigger_QuickSpin/resource.h File Reference . . . . .	1284
15.186 src/Acquisition/stdafx.cpp File Reference . . . . .	1284
15.187 src/ActionCommand/stdafx.cpp File Reference . . . . .	1285
15.188 src/BufferHandling/stdafx.cpp File Reference . . . . .	1285
15.189 src/CounterAndTimer/stdafx.cpp File Reference . . . . .	1286
15.190 src/DeviceEvents/stdafx.cpp File Reference . . . . .	1286
15.191 src/Enumeration/stdafx.cpp File Reference . . . . .	1287
15.192 src/Enumeration_QuickSpin/stdafx.cpp File Reference . . . . .	1287
15.193 src/ExceptionHandling/stdafx.cpp File Reference . . . . .	1288
15.194 src/Exposure/stdafx.cpp File Reference . . . . .	1288
15.195 src/Exposure_QuickSpin/stdafx.cpp File Reference . . . . .	1289
15.196 src/FileAccess_QuickSpin/stdafx.cpp File Reference . . . . .	1290
15.197 src/GigEVisionPerformance/stdafx.cpp File Reference . . . . .	1291
15.198 src/NodeMapInfo/stdafx.cpp File Reference . . . . .	1291
15.199 src/Sequencer/stdafx.cpp File Reference . . . . .	1292
15.200 src/SerialRxTx/stdafx.cpp File Reference . . . . .	1292
15.201 src/Acquisition/stdafx.h File Reference . . . . .	1293
15.202 src/ActionCommand/stdafx.h File Reference . . . . .	1294
15.203 src/BufferHandling/stdafx.h File Reference . . . . .	1295
15.204 src/CounterAndTimer/stdafx.h File Reference . . . . .	1296

---

---

15.205 src/DeviceEvents/stdafx.h File Reference . . . . .	1297
15.206 src/Enumeration/stdafx.h File Reference . . . . .	1298
15.207 src/Enumeration_QuickSpin/stdafx.h File Reference . . . . .	1299
15.208 src/ExceptionHandling/stdafx.h File Reference . . . . .	1300
15.209 src/Exposure/stdafx.h File Reference . . . . .	1301
15.210 src/Exposure_QuickSpin/stdafx.h File Reference . . . . .	1302
15.211 src/FileAccess_QuickSpin/stdafx.h File Reference . . . . .	1303
15.212 src/GigEVisionPerformance/stdafx.h File Reference . . . . .	1304
15.213 src/ImageFormatControl/stdafx.h File Reference . . . . .	1305
15.214 src/ImageFormatControl_QuickSpin/stdafx.h File Reference . . . . .	1305
15.215 src/NodeMapInfo/stdafx.h File Reference . . . . .	1306
15.216 src/Polarization/stdafx.h File Reference . . . . .	1307
15.217 src/Sequencer/stdafx.h File Reference . . . . .	1307
15.218 src/SerialRxTx/stdafx.h File Reference . . . . .	1308
15.219 src/Acquisition/targetver.h File Reference . . . . .	1309
15.220 src/ActionCommand/targetver.h File Reference . . . . .	1310
15.221 src/BufferHandling/targetver.h File Reference . . . . .	1311
15.222 src/CounterAndTimer/targetver.h File Reference . . . . .	1312
15.223 src/DeviceEvents/targetver.h File Reference . . . . .	1313
15.224 src/Enumeration/targetver.h File Reference . . . . .	1314
15.225 src/Enumeration_QuickSpin/targetver.h File Reference . . . . .	1315
15.226 src/ExceptionHandling/targetver.h File Reference . . . . .	1316
15.227 src/Exposure/targetver.h File Reference . . . . .	1317
15.228 src/Exposure_QuickSpin/targetver.h File Reference . . . . .	1318
15.229 src/FileAccess_QuickSpin/targetver.h File Reference . . . . .	1319
15.230 src/GigEVisionPerformance/targetver.h File Reference . . . . .	1320
15.231 src/GenTLInfo_QuickSpin/targetver.h File Reference . . . . .	1321
15.232 src/NodeMapInfo/targetver.h File Reference . . . . .	1321
15.233 src/Sequencer/targetver.h File Reference . . . . .	1322
15.234 src/SerialRxTx/targetver.h File Reference . . . . .	1323
15.235 src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery.cpp File Reference .	1324
15.235.1 Function Documentation . . . . .	1324
15.235.1.1 ConfigureCamera() . . . . .	1324
15.235.1.2 ConfigureUserSet1() . . . . .	1325
15.235.1.3 GetDeviceSerial() . . . . .	1325
15.235.1.4 main() . . . . .	1325
15.235.1.5 PrintExampleStatistics() . . . . .	1325
15.235.1.6 RefreshCameraList() . . . . .	1325
15.235.1.7 ResetCameraUserSetToDefault() . . . . .	1325
15.235.1.8 SleepyWrapper() . . . . .	1325
15.235.2 Variable Documentation . . . . .	1326
15.235.2.1 cameraGrabInfoMap . . . . .	1326

---

---

15.235.2.2 globalCamList . . . . .	1326
15.236 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference . . . . .	1326
15.236.1 Function Documentation . . . . .	1326
15.236.1.1 AcquireImages() . . . . .	1326
15.236.1.2 main() . . . . .	1327
15.236.1.3 PrintDeviceInfo() . . . . .	1327
15.236.1.4 RunMultipleCameras() . . . . .	1327
15.237 src/ActionCommand/ActionCommand.cpp File Reference . . . . .	1327
15.237.1 Function Documentation . . . . .	1328
15.237.1.1 AcquireImages() . . . . .	1328
15.237.1.2 ConfigureActionControl() . . . . .	1328
15.237.1.3 ConfigureChunkData() . . . . .	1328
15.237.1.4 ConfigureIEEE1588() . . . . .	1328
15.237.1.5 ConfigureInterface() . . . . .	1328
15.237.1.6 ConfigureOtherNodes() . . . . .	1328
15.237.1.7 ConfigureTrigger() . . . . .	1329
15.237.1.8 main() . . . . .	1329
15.237.1.9 PrintDeviceInfo() . . . . .	1329
15.237.1.10 RunMultipleCameras() . . . . .	1329
15.237.1.11 SleepyWrapper() . . . . .	1329
15.238 src/BufferHandling/BufferHandling.cpp File Reference . . . . .	1329
15.238.1 Macro Definition Documentation . . . . .	1330
15.238.1.1 k_numLoops . . . . .	1330
15.238.1.2 numBuffers . . . . .	1330
15.238.1.3 z_numTriggers . . . . .	1330
15.238.2 Function Documentation . . . . .	1330
15.238.2.1 AcquireImages() . . . . .	1330
15.238.2.2 ConfigureTrigger() . . . . .	1331
15.238.2.3 GrabNextImageByTrigger() . . . . .	1331
15.238.2.4 main() . . . . .	1331
15.238.2.5 PrintDeviceInfo() . . . . .	1331
15.238.2.6 ResetTrigger() . . . . .	1331
15.238.2.7 RunSingleCamera() . . . . .	1331
15.238.2.8 SleepyWrapper() . . . . .	1331
15.239 src/ChunkData/ChunkData.cpp File Reference . . . . .	1332
15.239.1 Enumeration Type Documentation . . . . .	1332
15.239.1.1 chunkDataType . . . . .	1332
15.239.2 Function Documentation . . . . .	1333
15.239.2.1 AcquireImages() . . . . .	1333
15.239.2.2 ConfigureChunkData() . . . . .	1333
15.239.2.3 DisableChunkData() . . . . .	1333
15.239.2.4 DisplayChunkData() [1 / 2] . . . . .	1333

---

---

15.239.2.5 DisplayChunkData() [2/2] . . . . .	1333
15.239.2.6 main() . . . . .	1333
15.239.2.7 PrintDeviceInfo() . . . . .	1334
15.239.2.8 RunSingleCamera() . . . . .	1334
15.239.3 Variable Documentation . . . . .	1334
15.239.3.1 chosenChunkData . . . . .	1334
15.240 src/CounterAndTimer/CounterAndTimer.cpp File Reference . . . . .	1334
15.240.1 Function Documentation . . . . .	1334
15.240.1.1 AcquireImages() . . . . .	1335
15.240.1.2 ConfigureDigitalIO() . . . . .	1335
15.240.1.3 ConfigureExposureandTrigger() . . . . .	1335
15.240.1.4 main() . . . . .	1335
15.240.1.5 PrintDeviceInfo() . . . . .	1335
15.240.1.6 ResetTrigger() . . . . .	1335
15.240.1.7 RunSingleCamera() . . . . .	1336
15.240.1.8 SetupCounterAndTimer() . . . . .	1336
15.241 src/DeviceEvents/DeviceEvents.cpp File Reference . . . . .	1336
15.241.1 Enumeration Type Documentation . . . . .	1336
15.241.1.1 eventType . . . . .	1336
15.241.2 Function Documentation . . . . .	1337
15.241.2.1 AcquireImages() . . . . .	1337
15.241.2.2 ConfigureDeviceEvents() . . . . .	1337
15.241.2.3 main() . . . . .	1337
15.241.2.4 PrintDeviceInfo() . . . . .	1337
15.241.2.5 ResetDeviceEvents() . . . . .	1337
15.241.2.6 RunSingleCamera() . . . . .	1338
15.241.3 Variable Documentation . . . . .	1338
15.241.3.1 chosenEvent . . . . .	1338
15.242 src/Enumeration/Enumeration.cpp File Reference . . . . .	1338
15.242.1 Function Documentation . . . . .	1338
15.242.1.1 main() . . . . .	1338
15.242.1.2 QueryInterface() . . . . .	1338
15.243 src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp File Reference . . . . .	1339
15.243.1 Function Documentation . . . . .	1339
15.243.1.1 main() . . . . .	1339
15.243.1.2 QueryInterface() . . . . .	1339
15.244 src/EnumerationEvents/EnumerationEvents.cpp File Reference . . . . .	1339
15.244.1 Function Documentation . . . . .	1340
15.244.1.1 CheckGevEnabled() . . . . .	1340
15.244.1.2 main() . . . . .	1340
15.245 src/ExceptionHandling/ExceptionHandling.cpp File Reference . . . . .	1340
15.245.1 Enumeration Type Documentation . . . . .	1341

---

15.245.1.1 exceptionType . . . . .	1341
15.245.2 Function Documentation . . . . .	1341
15.245.2.1 causeSpinnakerException() . . . . .	1341
15.245.2.2 causeStandardException() . . . . .	1341
15.245.2.3 main() . . . . .	1341
15.245.3 Variable Documentation . . . . .	1342
15.245.3.1 chosenException . . . . .	1342
15.246 src/Exposure/Exposure.cpp File Reference . . . . .	1342
15.246.1 Function Documentation . . . . .	1342
15.246.1.1 AcquireImages() . . . . .	1342
15.246.1.2 ConfigureExposure() . . . . .	1342
15.246.1.3 main() . . . . .	1343
15.246.1.4 PrintDeviceInfo() . . . . .	1343
15.246.1.5 ResetExposure() . . . . .	1343
15.246.1.6 RunSingleCamera() . . . . .	1343
15.247 src/Exposure_QuickSpin/Exposure_QuickSpin.cpp File Reference . . . . .	1343
15.247.1 Function Documentation . . . . .	1344
15.247.1.1 AcquireImages() . . . . .	1344
15.247.1.2 ConfigureExposure() . . . . .	1344
15.247.1.3 main() . . . . .	1344
15.247.1.4 PrintDeviceInfo() . . . . .	1344
15.247.1.5 ResetExposure() . . . . .	1344
15.247.1.6 RunSingleCamera() . . . . .	1344
15.248 src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp File Reference . . . . .	1345
15.248.1 Function Documentation . . . . .	1345
15.248.1.1 AcquireImages() . . . . .	1345
15.248.1.2 CloseFile() . . . . .	1346
15.248.1.3 DownloadImage() . . . . .	1346
15.248.1.4 ExecuteDeleteCommand() . . . . .	1346
15.248.1.5 ExecuteReadCommand() . . . . .	1346
15.248.1.6 ExecuteWriteCommand() . . . . .	1346
15.248.1.7 InitializeSystem() . . . . .	1346
15.248.1.8 main() . . . . .	1346
15.248.1.9 OpenFileToRead() . . . . .	1347
15.248.1.10 OpenFileToWrite() . . . . .	1347
15.248.1.11 PrintDebugMessage() . . . . .	1347
15.248.1.12 PrintDeviceInfo() . . . . .	1347
15.248.1.13 PrintResultMessage() . . . . .	1347
15.248.1.14 PrintUsage() . . . . .	1347
15.248.1.15 UploadImage() . . . . .	1347
15.248.2 Variable Documentation . . . . .	1348
15.248.2.1 _enableDebug . . . . .	1348

---

15.248.2.2 _fileSelector . . . . .	1348
15.249 src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp File Reference . . . . .	1348
15.249.1 Function Documentation . . . . .	1348
15.249.1.1 main() . . . . .	1348
15.249.1.2 PrintApplicationLayerDeviceInfo() . . . . .	1349
15.249.1.3 PrintTransportLayerDeviceInfo() . . . . .	1349
15.249.1.4 PrintTransportLayerInterfaceInfo() . . . . .	1349
15.249.1.5 PrintTransportLayerStreamInfo() . . . . .	1349
15.250 src/GigEVisionPerformance/CpuUtil.cpp File Reference . . . . .	1349
15.251 src/GigEVisionPerformance/CpuUtil.h File Reference . . . . .	1350
15.252 src/GigEVisionPerformance/GigEVisionPerformance.cpp File Reference . . . . .	1351
15.252.1 Function Documentation . . . . .	1352
15.252.1.1 AcquireImages() . . . . .	1352
15.252.1.2 EnableManualFramerate() . . . . .	1352
15.252.1.3 getCameraCategory() . . . . .	1352
15.252.1.4 main() . . . . .	1353
15.252.1.5 ParseArguments() . . . . .	1353
15.252.1.6 PrintAllNodes() . . . . .	1353
15.252.1.7 PrintCPUUsage() . . . . .	1353
15.252.1.8 PrintDataStreamInfo() . . . . .	1353
15.252.1.9 PrintDeviceInfo() . . . . .	1353
15.252.1.10 PrintUsage() . . . . .	1353
15.252.1.11 RunSingleCamera() . . . . .	1354
15.252.1.12 SetFrameRate() . . . . .	1354
15.252.2 Variable Documentation . . . . .	1354
15.252.2.1 argBayerRG . . . . .	1354
15.252.2.2 argDuration . . . . .	1354
15.252.2.3 argMaxFrames . . . . .	1354
15.252.2.4 argNumImages . . . . .	1354
15.252.2.5 argPacketDelay . . . . .	1354
15.252.2.6 argPacketSize . . . . .	1355
15.252.2.7 argPrintUsage . . . . .	1355
15.252.2.8 argRelease . . . . .	1355
15.252.2.9 argUserSetFrames . . . . .	1355
15.252.2.10 cpuUsageInfo . . . . .	1355
15.252.2.11 IsRelease . . . . .	1355
15.252.2.12 NumImagesToGrab . . . . .	1355
15.252.2.13 PacketDelayToSet . . . . .	1355
15.252.2.14 PacketSizeToSet . . . . .	1356
15.252.2.15 PixelFormatToSet . . . . .	1356
15.252.2.16 TestDuration . . . . .	1356
15.252.2.17 UseDuration . . . . .	1356

---

15.252.2.18 UseMaxFramerate . . . . .	1356
15.252.2.19 UserSetFramerate . . . . .	1356
15.253 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference . . . . .	1356
15.254 src/HighDynamicRange/HighDynamicRange.cpp File Reference . . . . .	1357
15.254.1 Function Documentation . . . . .	1357
15.254.1.1 CheckNodeAccessibility() . . . . .	1357
15.254.1.2 InitializeHDRImages() . . . . .	1357
15.254.1.3 main() . . . . .	1358
15.254.1.4 PrintBuildInfo() . . . . .	1358
15.254.1.5 PrintDeviceInfo() . . . . .	1358
15.254.1.6 RunSingleCamera() . . . . .	1358
15.254.1.7 ToggleHDRMode() . . . . .	1358
15.254.2 Variable Documentation . . . . .	1358
15.254.2.1 k_HDRGain1 . . . . .	1358
15.254.2.2 k_HDRGain2 . . . . .	1359
15.254.2.3 k_HDRGain3 . . . . .	1359
15.254.2.4 k_HDRGain4 . . . . .	1359
15.254.2.5 k_HDRShutter1 . . . . .	1359
15.254.2.6 k_HDRShutter2 . . . . .	1359
15.254.2.7 k_HDRShutter3 . . . . .	1359
15.254.2.8 k_HDRShutter4 . . . . .	1359
15.255 src/ImageEvents/ImageEvents.cpp File Reference . . . . .	1360
15.255.1 Function Documentation . . . . .	1360
15.255.1.1 AcquireImages() . . . . .	1360
15.255.1.2 ConfigureImageEvents() . . . . .	1360
15.255.1.3 main() . . . . .	1361
15.255.1.4 PrintDeviceInfo() . . . . .	1361
15.255.1.5 ResetImageEvents() . . . . .	1361
15.255.1.6 RunSingleCamera() . . . . .	1361
15.255.1.7 SleepyWrapper() . . . . .	1361
15.255.1.8 WaitForImages() . . . . .	1361
15.256 src/ImageFormatControl/ImageFormatControl.cpp File Reference . . . . .	1362
15.256.1 Function Documentation . . . . .	1362
15.256.1.1 AcquireImages() . . . . .	1362
15.256.1.2 ConfigureCustomImageSettings() . . . . .	1362
15.256.1.3 main() . . . . .	1362
15.256.1.4 PrintDeviceInfo() . . . . .	1363
15.256.1.5 RunSingleCamera() . . . . .	1363
15.257 src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp File Reference . . . . .	1363
15.257.1 Function Documentation . . . . .	1363
15.257.1.1 AcquireImages() . . . . .	1363
15.257.1.2 ConfigureCustomImageSettings() . . . . .	1364

---

---

15.257.1.3 main() . . . . .	1364
15.257.1.4 PrintDeviceInfo() . . . . .	1364
15.257.1.5 RunSingleCamera() . . . . .	1364
15.258 src/Inference/Inference.cpp File Reference . . . . .	1364
15.258.1 Enumeration Type Documentation . . . . .	1365
15.258.1.1 FileUploadPersistence . . . . .	1365
15.258.1.2 InferenceNetworkType . . . . .	1366
15.258.2 Function Documentation . . . . .	1366
15.258.2.1 AcquireImages() . . . . .	1366
15.258.2.2 CameraCloseFile() . . . . .	1366
15.258.2.3 CameraDeleteFile() . . . . .	1366
15.258.2.4 CameraOpenFile() . . . . .	1366
15.258.2.5 CameraWriteToFile() . . . . .	1367
15.258.2.6 ConfigureChunkData() . . . . .	1367
15.258.2.7 ConfigureInference() . . . . .	1367
15.258.2.8 ConfigureTestPattern() . . . . .	1367
15.258.2.9 ConfigureTrigger() . . . . .	1367
15.258.2.10 DeleteFileOnCamera() . . . . .	1367
15.258.2.11 DisableChunkData() . . . . .	1368
15.258.2.12 DisableTrigger() . . . . .	1368
15.258.2.13 DisplayChunkData() . . . . .	1368
15.258.2.14 labelClassification() . . . . .	1368
15.258.2.15 labelDetection() . . . . .	1368
15.258.2.16 LoadFileIntoMemory() . . . . .	1368
15.258.2.17 main() . . . . .	1369
15.258.2.18 PrintDeviceInfo() . . . . .	1369
15.258.2.19 RunSingleCamera() . . . . .	1369
15.258.2.20 SetChunkEnable() . . . . .	1369
15.258.2.21 UploadFileToCamera() . . . . .	1369
15.258.3 Variable Documentation . . . . .	1369
15.258.3.1 arrayLabelClassification . . . . .	1369
15.258.3.2 arrayLabelDetection . . . . .	1370
15.258.3.3 chosenFileUploadPersistence . . . . .	1370
15.258.3.4 chosenInferenceNetworkType . . . . .	1370
15.258.3.5 injectedImagePath . . . . .	1370
15.258.3.6 injectedImageHeight . . . . .	1370
15.258.3.7 injectedImageWidth . . . . .	1370
15.258.3.8 networkFilePath . . . . .	1371
15.259 src/Logging/Logging.cpp File Reference . . . . .	1371
15.259.1 Function Documentation . . . . .	1371
15.259.1.1 main() . . . . .	1371
15.259.2 Variable Documentation . . . . .	1371

---

15.259.2.1 k_LoggingLevel . . . . .	1372
15.260 src/LogicBlock/LogicBlock.cpp File Reference . . . . .	1372
15.260.1 Function Documentation . . . . .	1372
15.260.1.1 AcquireImages() . . . . .	1372
15.260.1.2 ConfigureLogicBlock() . . . . .	1372
15.260.1.3 ConfigureTrigger() . . . . .	1373
15.260.1.4 GrabTwoImages() . . . . .	1373
15.260.1.5 main() . . . . .	1373
15.260.1.6 PrintDeviceInfo() . . . . .	1373
15.260.1.7 ResetExposure() . . . . .	1373
15.260.1.8 ResetTrigger() . . . . .	1373
15.260.1.9 RunSingleCamera() . . . . .	1373
15.261 src/LookupTable/LookupTable.cpp File Reference . . . . .	1374
15.261.1 Function Documentation . . . . .	1374
15.261.1.1 AcquireImages() . . . . .	1374
15.261.1.2 ConfigureLookupTables() . . . . .	1374
15.261.1.3 main() . . . . .	1374
15.261.1.4 PrintDeviceInfo() . . . . .	1375
15.261.1.5 PrintRetrieveNodeFailure() . . . . .	1375
15.261.1.6 ResetLookupTables() . . . . .	1375
15.261.1.7 RunSingleCamera() . . . . .	1375
15.262 src/NodeMapCallback/NodeMapCallback.cpp File Reference . . . . .	1375
15.262.1 Function Documentation . . . . .	1376
15.262.1.1 ChangeHeightAndGain() . . . . .	1376
15.262.1.2 ConfigureCallbacks() . . . . .	1376
15.262.1.3 main() . . . . .	1376
15.262.1.4 OnGainNodeUpdate() . . . . .	1376
15.262.1.5 OnHeightNodeUpdate() . . . . .	1376
15.262.1.6 PrintDeviceInfo() . . . . .	1376
15.262.1.7 ResetCallbacks() . . . . .	1377
15.262.1.8 RunSingleCamera() . . . . .	1377
15.263 src/NodeMapInfo/NodeMapInfo.cpp File Reference . . . . .	1377
15.263.1 Enumeration Type Documentation . . . . .	1378
15.263.1.1 readType . . . . .	1378
15.263.2 Function Documentation . . . . .	1378
15.263.2.1 Indent() . . . . .	1378
15.263.2.2 main() . . . . .	1378
15.263.2.3 PrintBooleanNode() . . . . .	1378
15.263.2.4 PrintCategoryNodeAndAllFeatures() . . . . .	1379
15.263.2.5 PrintCommandNode() . . . . .	1379
15.263.2.6 PrintEnumerationNodeAndCurrentEntry() . . . . .	1379
15.263.2.7 PrintEnumerationSelector() . . . . .	1379

---

15.263.2.8 PrintFloatNode()	1379
15.263.2.9 PrintIntegerNode()	1379
15.263.2.10 PrintNode()	1380
15.263.2.11 PrintStringNode()	1380
15.263.2.12 PrintValueNode()	1380
15.263.2.13 RunSingleCamera()	1380
15.263.3 Variable Documentation	1380
15.263.3.1 chosenRead	1380
15.263.3.2 maxChars	1380
15.264 src/Polarization/Polarization.cpp File Reference	1381
15.264.1 Function Documentation	1381
15.264.1.1 AcquireImages()	1381
15.264.1.2 ConfigureStream()	1382
15.264.1.3 CreateAndSaveAolpDolplImages()	1382
15.264.1.4 CreateAndSaveGlareReducedImage()	1382
15.264.1.5 CreateAndSaveStokesImages()	1382
15.264.1.6 CreateHeatmaplImages()	1382
15.264.1.7 CreateNormalizedImage()	1382
15.264.1.8 ExtractAndSavePolarQuadlImages()	1383
15.264.1.9 GetQuadFileNameAppendage()	1383
15.264.1.10 main()	1383
15.264.1.11 PrintDeviceInfo()	1383
15.264.1.12 RunSingleCamera()	1383
15.264.1.13 SaveImage()	1383
15.264.2 Variable Documentation	1383
15.264.2.1 isPixelFormatColor	1384
15.265 src/SaveToAvi/SaveToAvi.cpp File Reference	1384
15.265.1 Enumeration Type Documentation	1384
15.265.1.1 videoType	1384
15.265.2 Function Documentation	1385
15.265.2.1 AcquireImages()	1385
15.265.2.2 main()	1385
15.265.2.3 PrintDeviceInfo()	1385
15.265.2.4 RunSingleCamera()	1385
15.265.2.5 SaveVectorToVideo()	1385
15.265.3 Variable Documentation	1385
15.265.3.1 chosenVideoType	1386
15.266 src/Sequencer/Sequencer.cpp File Reference	1386
15.266.1 Function Documentation	1386
15.266.1.1 AcquireImages()	1386
15.266.1.2 ConfigureSequencerPartOne()	1386
15.266.1.3 ConfigureSequencerPartTwo()	1387

---

---

15.266.1.4 main()	1387
15.266.1.5 PrintDeviceInfo()	1387
15.266.1.6 PrintRetrieveNodeFailure()	1387
15.266.1.7 ResetSequencer()	1387
15.266.1.8 RunSingleCamera()	1387
15.266.1.9 SetSingleState()	1388
15.267 src/SerialRxTx/SerialRxTx.cpp File Reference	1388
15.267.1 Macro Definition Documentation	1388
15.267.1.1 COM_PORT_COUNT_MAX	1389
15.267.1.2 DATA_BITS	1389
15.267.1.3 MILLISECOND	1389
15.267.1.4 SERIAL_PORT_BAUD_RATE	1389
15.267.1.5 SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND	1389
15.267.1.6 SERIAL_PORT_DELAY	1389
15.267.1.7 SERIAL_PORT_PARITY_BITS	1389
15.267.1.8 SERIAL_PORT_STOP_BITS	1389
15.267.1.9 TWO_SECOND_DELAY	1390
15.267.2 Function Documentation	1390
15.267.2.1 CleanUp()	1390
15.267.2.2 ConfigureDevice()	1390
15.267.2.3 main()	1390
15.267.2.4 PrintDeviceInfo()	1390
15.267.2.5 RunSingleCamera()	1390
15.267.2.6 SerialRx()	1391
15.267.2.7 SerialTx()	1391
15.268 src/Trigger/Trigger.cpp File Reference	1391
15.268.1 Enumeration Type Documentation	1392
15.268.1.1 triggerType	1392
15.268.2 Function Documentation	1393
15.268.2.1 AcquireImages()	1393
15.268.2.2 ConfigureTrigger()	1393
15.268.2.3 GrabNextImageByTrigger()	1393
15.268.2.4 main()	1393
15.268.2.5 PrintDeviceInfo()	1393
15.268.2.6 ResetTrigger()	1394
15.268.2.7 RunSingleCamera()	1394
15.268.3 Variable Documentation	1394
15.268.3.1 chosenTrigger	1394
15.269 src/Trigger_QuickSpin/Trigger_QuickSpin.cpp File Reference	1394
15.269.1 Enumeration Type Documentation	1395
15.269.1.1 triggerType	1395
15.269.2 Function Documentation	1395

---

15.269.2.1 AcquireImages() . . . . .	1395
15.269.2.2 ConfigureTrigger() . . . . .	1395
15.269.2.3 GrabNextImageByTrigger() . . . . .	1395
15.269.2.4 main() . . . . .	1396
15.269.2.5 PrintDeviceInfo() . . . . .	1396
15.269.2.6 ResetTrigger() . . . . .	1396
15.269.2.7 RunSingleCamera() . . . . .	1396
15.269.3 Variable Documentation . . . . .	1396
15.269.3.1 chosenTrigger . . . . .	1396
<b>16 Example Documentation</b> . . . . .	<b>1397</b>
16.1 Acquisition.cpp . . . . .	1397
16.2 AcquisitionMultipleCameraRecovery.cpp . . . . .	1397
16.3 AcquisitionMultipleThread.cpp . . . . .	1397
16.4 ActionCommand.cpp . . . . .	1398
16.5 BufferHandling.cpp . . . . .	1398
16.6 CounterAndTimer.cpp . . . . .	1398
16.7 DeviceEvents.cpp . . . . .	1398
16.8 EnumerationEvents.cpp . . . . .	1399
16.9 ExceptionHandling.cpp . . . . .	1399
16.10 Exposure.cpp . . . . .	1399
16.11 FileAccess_Quickspin.cpp . . . . .	1399
16.12 GenTLInfo_QuickSpin.cpp . . . . .	1400
16.13 GigEVisionPerformance.cpp . . . . .	1400
16.14 HighDynamicRange.cpp . . . . .	1400
16.15 ImageEvents.cpp . . . . .	1400
16.16 ImageFormatControl.cpp . . . . .	1400
16.17 ImageFormatControl_QuickSpin.cpp . . . . .	1401
16.18 Inference.cpp . . . . .	1401
16.19 Logging.cpp . . . . .	1401
16.20 LogicBlock.cpp . . . . .	1402
16.21 LookupTable.cpp . . . . .	1402
16.22 NodeMapCallback.cpp . . . . .	1402
16.23 NodeMapInfo.cpp . . . . .	1402
16.24 Polarization.cpp . . . . .	1403
16.25 SaveToAvi.cpp . . . . .	1403
16.26 Sequencer.cpp . . . . .	1403
16.27 SerialRxTx.cpp . . . . .	1403
16.28 Trigger.cpp . . . . .	1404
16.29 Trigger_QuickSpin.cpp . . . . .	1404
<b>Index</b> . . . . .	<b>1405</b>

## 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](#)
- [FlyCapture2 Feature Comparison with Spinnaker](#)
- [Programmer's Guide](#)
- [SpinViewGuide](#)
- [Working with GenICam GenTL Devices](#)



## **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.GentLCatgeory=ERROR, GentLCatgeory

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

log4cpp.appender.GentLCatgeory=RollingFileAppender
log4cpp.appender.GentLCatgeory.fileName=$(ALLUSERSPROFILE)\Spinnaker\Logs\GenTL.log
log4cpp.appender.GentLCatgeory.append=true
log4cpp.appender.GentLCatgeory.maxFileSize=1000000
log4cpp.appender.GentLCatgeory.maxBackupIndex=5
log4cpp.appender.GentLCatgeory.layout=PatternLayout
log4cpp.appender.GentLCatgeory.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  $(bufferSize + 1024 - 1) / 1024 * 1024$  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
AdapterList	The Code Project Open License (CPOL)
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 (CPOL) 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

## Module Index

### 7.1 Modules

Here is a list of all modules:

Spinnaker Classes . . . . .	39
AVI Recorder Class . . . . .	40
BasePtr Class . . . . .	41
Camera Class . . . . .	42
Camera Base Class . . . . .	43
CameraDefs Class . . . . .	44
Camera List Class . . . . .	45
CameraPtr Class . . . . .	46
ChunkData Class . . . . .	47
Chunk Data Inference Class . . . . .	48
Spinnaker EventHandler Classes . . . . .	49
DeviceArrivalEventHandler Class . . . . .	50
DeviceEventHandler Class . . . . .	51
DeviceRemovalEventHandler Class . . . . .	52
EventHandler Class . . . . .	53
Exception Class . . . . .	54
Image Class . . . . .	55
ImageEventHandler Class . . . . .	56
ImagePtr Class . . . . .	57
ImageStatistics Class . . . . .	58
Image Utility Class . . . . .	59
Image Utility Heatmap Class . . . . .	60
Image Utility Polarization Class . . . . .	61
Interface Class . . . . .	62
InterfaceArrivalEventHandler Class . . . . .	63
InterfaceEventHandler Class . . . . .	64
InterfaceList Class . . . . .	65
InterfacePtr Class . . . . .	66
InterfaceRemovalEventHandler Class . . . . .	67
Logging EventHandler Class . . . . .	68
LoggingEventDataPtr Class . . . . .	69
LoggingEventHandler Class . . . . .	70
Spinnaker Headers . . . . .	71
Spinnaker.h . . . . .	72
Spinnaker Definitions . . . . .	73

Spinnaker Platform . . . . .	74
Spinnaker Video Class . . . . .	75
Spinnaker Video Definitions . . . . .	76
System Class . . . . .	77
SystemEventHandler Class . . . . .	78
SystemPtr Class . . . . .	79
Spinnaker QuickSpin Classes . . . . .	80
TransportLayerDefs Class . . . . .	81
TransportLayerDevice Class . . . . .	82
TransportLayerInterface Class . . . . .	83
TransportLayerStream Class . . . . .	84
TransportLayerSystem Class . . . . .	85
Camera Base Interface Class . . . . .	86
IChunkData Class . . . . .	87
IIImage Class . . . . .	88
IIImageStatistics Class . . . . .	89
IIInterface Class . . . . .	90
IIInterfaceList Class . . . . .	91
ISystem Class . . . . .	92
AutoVector Class . . . . .	94
Spinnaker GenApi Interfaces . . . . .	95
IBase Interface . . . . .	96
BooleanNode Class . . . . .	97
CategoryNode Class . . . . .	98
ChunkAdapter Class . . . . .	99
ChunkAdapterDcam Class . . . . .	100
ChunkAdapterGeneric Class . . . . .	101
ChunkAdapterGEV Class . . . . .	102
ChunkPort Class . . . . .	103
CommandNode Class . . . . .	104
Container Class . . . . .	105
Counter Class . . . . .	106
EnumClasses Class . . . . .	107
EnumEntryNode Class . . . . .	108
EnumNode Class . . . . .	109
EnumNodeT Class . . . . .	110
EventAdapter Class . . . . .	111
EventAdapter1394 Class . . . . .	112
EventAdapterGeneric Class . . . . .	113
EventAdapterGEV Class . . . . .	114
EventAdapterU3V Class . . . . .	115
EventPort Class . . . . .	116
Filestream Class . . . . .	117
FloatNode Class . . . . .	118
FloatRegNode Class . . . . .	119
GCString Class . . . . .	120
GCSynch Class . . . . .	121
GCTypes Class . . . . .	122
Spinnaker GenApi Utilities . . . . .	123
GCUtilities Utility . . . . .	124
IBoolean Interface . . . . .	125
ICategory Interfaces . . . . .	126
IChunkPort Interface . . . . .	127
ICommand Interface . . . . .	128
IDestroy Interface . . . . .	129
IDeviceInfo Interface . . . . .	130
IEnumEntry Interface . . . . .	131
IEnumeration Interface . . . . .	132

IEnumerationT Interface . . . . .	133
IFloat Interface . . . . .	134
IInteger Interface . . . . .	135
INode Interface . . . . .	136
INodeMap Interface . . . . .	140
INodeMapDyn Interface . . . . .	141
IntegerNode Class . . . . .	142
IntRegNode Class . . . . .	143
IPort Interface . . . . .	144
IPortConstruct Interface . . . . .	145
IPortRecorder Interface . . . . .	146
IRegister Interfaces . . . . .	147
ISelector Interface . . . . .	148
ISelectorDigit Interface . . . . .	149
IString Class . . . . .	150
IValue Class . . . . .	151
Node Class . . . . .	152
NodeCallback Class . . . . .	153
NodeMap Class . . . . .	154
NodeMapFactory Class . . . . .	155
NodeMapRef Class . . . . .	156
Persistence Class . . . . .	157
Pointer Class . . . . .	158
Spinnaker GenApi Classes . . . . .	93
PortImpl Class . . . . .	164
PortNode Class . . . . .	165
PortRecorder Class . . . . .	166
PortReplay Class . . . . .	167
PortWriteList Class . . . . .	168
Reference Interfaces . . . . .	169
RegisterNode Class . . . . .	170
RegisterPortImpl Class . . . . .	171
SelectorSet Class . . . . .	172
SpinTestCamera Class . . . . .	173
StringNode Class . . . . .	174
StringRegNode Class . . . . .	175
StructPort Class . . . . .	176
Synch Class . . . . .	177
Spinnaker GenApi Enums . . . . .	178
Types Enums . . . . .	179
ValueNode Class . . . . .	180
ChunkAdapterU3V Class . . . . .	181



# Chapter 8

## Namespace Index

### 8.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AdapterConfig . . . . .	183
Conversion . . . . .	187
CpuUtil . . . . .	188
PerformanceCounter . . . . .	188
SecondsCounter . . . . .	189
Spinnaker . . . . .	191
Spinnaker::GenApi . . . . .	329
Spinnaker::GenICam . . . . .	386
Spinnaker::Video . . . . .	391



# Chapter 9

## Hierarchical Index

### 9.1 Class Hierarchy

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

ActionCommandResult . . . . .	393
AdapterInfo . . . . .	395
AttachStatistics_t . . . . .	399
AutoLock . . . . .	400
AutoLock . . . . .	400
AVIOption . . . . .	401
BasePtr< T, B > . . . . .	402
BasePtr< Camera, ICameraBase > . . . . .	402
CameraPtr . . . . .	581
BasePtr< IImage > . . . . .	402
ImagePtr . . . . .	888
BasePtr< IInterface > . . . . .	402
InterfacePtr . . . . .	947
BasePtr< ISystem > . . . . .	402
SystemPtr . . . . .	1059
BasePtr< LoggingEventData > . . . . .	402
LoggingEventDataPtr . . . . .	973
basic_istream . . . . .	
IDevFileStreamBase< CharType, Traits > . . . . .	812
basic_ostream . . . . .	
ODevFileStreamBase< CharType, Traits > . . . . .	1003
basic_streampbuf . . . . .	
IDevFileStreamBuf< CharType, Traits > . . . . .	814
ODevFileStreamBuf< CharType, Traits > . . . . .	1005
BMPOption . . . . .	406
CChunkAdapter . . . . .	586
CChunkAdapterDcam . . . . .	589
CChunkAdapterGeneric . . . . .	592
CChunkAdapterGEV . . . . .	594
CChunkAdapterU3V . . . . .	596
CDataStruct . . . . .	
CTestPortStruct< CDataStruct > . . . . .	697
CEventAdapter . . . . .	609

CEventAdapter1394 . . . . .	611
CEventAdapterGeneric . . . . .	613
CEventAdapterGEV . . . . .	616
CEventAdapterU3V . . . . .	618
CGeneric_XMLLoaderParams . . . . .	630
CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams > . . . . .	668
CNodeMapRef . . . . .	665
CGlobalLock . . . . .	631
CGlobalLockUnlocker . . . . .	633
CLock . . . . .	648
CLockEx . . . . .	652
CLock . . . . .	649
CLockEx . . . . .	653
CNodeCallback . . . . .	654
Function_NodeCallback< Function > . . . . .	760
Member_NodeCallback< Client, Member > . . . . .	978
CNodeMapFactory . . . . .	657
Counter . . . . .	677
CPointer< T, B > . . . . .	679
CPointer< IFloat, IBase > . . . . .	679
CFloatPtr . . . . .	628
CPointer< INode, IBase > . . . . .	679
CpuUsageInfo . . . . .	690
DCAM_CHECKSUM . . . . .	701
DCAM_CHUNK_TRAILER . . . . .	701
double_avector_t . . . . .	712
EAccessModeClass . . . . .	715
ECachingModeClass . . . . .	716
EDisplayNotationClass . . . . .	717
EEndianessClass . . . . .	718
EGenApiSchemaVersionClass . . . . .	720
EInputDirectionClass . . . . .	721
ENameSpaceClass . . . . .	722
ERepresentationClass . . . . .	731
ESignClass . . . . .	732
ESlopeClass . . . . .	733
EStandardNameSpaceClass . . . . .	734
EventHandler . . . . .	736
IDeviceArrivalEventHandler . . . . .	817
DeviceArrivalEventHandler . . . . .	702
IInterfaceEventHandler . . . . .	848
InterfaceEventHandler . . . . .	936
InterfaceEventHandlerImpl . . . . .	939
IDeviceEventHandler . . . . .	819
DeviceEventHandler . . . . .	705
DeviceEventHandlerImpl . . . . .	708
IDeviceRemovalEventHandler . . . . .	821
DeviceRemovalEventHandler . . . . .	710
IInterfaceEventHandler . . . . .	848
ImageEventHandler . . . . .	835
ImageEventHandlerImpl . . . . .	882
ImageEventHandlerImpl . . . . .	885
IInterfaceArrivalEventHandler . . . . .	846
InterfaceArrivalEventHandler . . . . .	934
ISystemEventHandler . . . . .	961
SystemEventHandler . . . . .	1053

SystemEventHandlerImpl . . . . .	1056
IInterfaceRemovalEventHandler . . . . .	853
InterfaceRemovalEventHandler . . . . .	949
ISystemEventHandler . . . . .	961
ILoggingEventHandler . . . . .	855
LoggingEventHandler . . . . .	975
LoggingEventHandlerImpl . . . . .	977
EVisibilityClass . . . . .	739
exception	
AdapterConfigException . . . . .	394
Exception . . . . .	740
EYesNoClass . . . . .	745
FileProtocolAdapter . . . . .	746
gcstring . . . . .	762
GrabInfo . . . . .	771
GVCP_CHUNK_TRAILER . . . . .	772
GVCP_EVENT_ITEM . . . . .	773
GVCP_EVENT_ITEM_BASIC . . . . .	774
GVCP_EVENT_ITEM_EXTENDED_ID . . . . .	775
GVCP_EVENT_REQUEST . . . . .	777
GVCP_EVENT_REQUEST_EXTENDED_ID . . . . .	778
GVCP_EVENTDATA_REQUEST . . . . .	779
GVCP_EVENTDATA_REQUEST_EXTENDED_ID . . . . .	780
GVCP_REQUEST_HEADER . . . . .	781
H264Option . . . . .	782
IBoolean	
BooleanNode . . . . .	408
ICameraBase . . . . .	784
CameraBase . . . . .	561
Camera . . . . .	411
ICameraList . . . . .	793
CameraList . . . . .	576
ICategory	
CategoryNode . . . . .	584
IChunkData . . . . .	797
ChunkData . . . . .	635
IChunkPort	
PortNode . . . . .	1010
PortReplay . . . . .	1020
PortRecorder . . . . .	1016
ICommand	
CommandNode . . . . .	674
IDataStream . . . . .	806
IDeviceInfo	
NodeMap . . . . .	993
SpinTestCamera . . . . .	1031
IEnumEntry	
EnumEntryNode . . . . .	723
IEnumeration	
EnumNode . . . . .	726
CEnumerationTRef< EnumT > . . . . .	604
IEnumerationT	
CEnumerationTRef< EnumT > . . . . .	604
IFloat	
FloatNode . . . . .	750
FloatRegNode . . . . .	757

IImage . . . . .	823
Image . . . . .	857
IIImageStatistics . . . . .	837
ImageStatistics . . . . .	890
IIInteger . . . . .	841
IntegerNode . . . . .	923
IntRegNode . . . . .	951
IIInterface . . . . .	929
Interface . . . . .	851
InterfaceList . . . . .	943
ImageUtility . . . . .	897
ImageUtilityHeatmap . . . . .	902
ImageUtilityPolarization . . . . .	906
InferenceBoundingBox . . . . .	913
InferenceBoundingBoxResult . . . . .	915
InferenceBoxCircle . . . . .	918
InferenceBoxRect . . . . .	918
InferenceBoxRotatedRect . . . . .	919
INode . . . . .	982
Node . . . . .	694
CSelectorSet . . . . .	1010
PortNode . . . . .	1102
ValueNode . . . . .	408
BooleanNode . . . . .	584
CategoryNode . . . . .	674
CommandNode . . . . .	723
EnumEntryNode . . . . .	726
EnumNode . . . . .	750
FloatNode . . . . .	923
IntegerNode . . . . .	1024
RegisterNode . . . . .	757
FloatRegNode . . . . .	951
StringRegNode . . . . .	1040
StringNode . . . . .	1036
StringRegNode . . . . .	1040
INodeMap . . . . .	993
NodeMap . . . . .	920
int64_avector_t . . . . .	953
IPersistScript . . . . .	625
CFeatureBag . . . . .	684
IpInfo . . . . .	691
IPortConstruct . . . . .	697
CChunkPort . . . . .	1010
CEventPort . . . . .	1016
CPortImpl . . . . .	1020
CRegisterPortImpl . . . . .	1020
CTestPortStruct< CDataStruct > . . . . .	1020
PortNode . . . . .	1020
IPortRecorder . . . . .	1020
PortNode . . . . .	1020
PortRecorder . . . . .	1020
IPortReplay . . . . .	1020
CPortImpl . . . . .	1020
PortReplay . . . . .	1020
IPortWriteList . . . . .	1020

CPortWriteList . . . . .	687
IRegister	
RegisterNode . . . . .	1024
IString	
StringNode . . . . .	1036
ISystem	
System . . . . .	1043
IValue	
ValueNode . . . . .	1102
JPEGOption . . . . .	963
JPG2Option . . . . .	964
LibraryVersion . . . . .	966
LockableObject< Object >::Lock . . . . .	967
LockableObject< Object > . . . . .	968
LoggingEventData . . . . .	969
MJPGOption . . . . .	980
CNodeMapFactory::NodeStatistics_t . . . . .	1002
PGMOption . . . . .	1007
PNGOption . . . . .	1008
PPMOption . . . . .	1023
SingleChunkData_t . . . . .	1029
SingleChunkDataStr_t . . . . .	1030
SpinVideo . . . . .	1031
TCameraParams	
CNodeMapRefT< TCameraParams > . . . . .	668
TIFFOption . . . . .	1061
TransportLayerDevice . . . . .	1063
TransportLayerInterface . . . . .	1073
TransportLayerStream . . . . .	1084
TransportLayerSystem . . . . .	1092
U3V_CHUNK_TRAILER . . . . .	1099
U3V_COMMAND_HEADER . . . . .	1099
U3V_EVENT_DATA . . . . .	1100
U3V_EVENT_MESSAGE . . . . .	1101
Version_t . . . . .	1105



# Chapter 10

## Class Index

### 10.1 Class List

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

<a href="#">ActionCommandResult</a>	
Action Command Result . . . . .	393
<a href="#">AdapterConfigException</a>	
. . . . .	394
<a href="#">AdapterInfo</a>	
. . . . .	395
<a href="#">AttachStatistics_t</a>	
Delivers information about the attached chunks and nodes . . . . .	399
<a href="#">AutoLock</a>	
. . . . .	400
<a href="#">AutoLock</a>	
. . . . .	400
<a href="#">AVIOption</a>	
Options for saving AVI files . . . . .	401
<a href="#">BasePtr&lt; T, B &gt;</a>	
The base class of the <a href="#">SystemPtr</a> , <a href="#">CameraPtr</a> , <a href="#">InterfacePtr</a> , <a href="#">ImagePtr</a> and <a href="#">LoggingEventDataPtr</a> objects . . . . .	402
<a href="#">BMPOption</a>	
Options for saving Bitmap image . . . . .	406
<a href="#">BooleanNode</a>	
<a href="#">Interface</a> for string properties . . . . .	408
<a href="#">Camera</a>	
The camera object class . . . . .	411
<a href="#">CameraBase</a>	
The base class for the camera object . . . . .	561
<a href="#">CameraList</a>	
Used to hold a list of camera objects . . . . .	576
<a href="#">CameraPtr</a>	
A reference tracked pointer to a camera object . . . . .	581
<a href="#">CategoryNode</a>	
<a href="#">Interface</a> for string properties . . . . .	584
<a href="#">CChunkAdapter</a>	
Connects a chunked buffer to a node map . . . . .	586
<a href="#">CChunkAdapterDcam</a>	
Connects a chunked DCAM buffer to a node map . . . . .	589
<a href="#">CChunkAdapterGeneric</a>	
<a href="#">CChunkAdapterGEV</a>	
Connects a chunked DCAM buffer to a node map . . . . .	594
<a href="#">CChunkAdapterU3V</a>	
Connects a chunked U3V buffer to a node map . . . . .	596

CChunkPort	Port attachable to a chunk in a buffer . . . . .	599
CEnumerationTRef< EnumT >	Interface for string properties . . . . .	604
CEventAdapter	Delivers Events to ports . . . . .	609
CEventAdapter1394	Distribute the events to the node map . . . . .	611
CEventAdapterGeneric	Connects a generic event to a node map . . . . .	613
CEventAdapterGEV	Connects a GigE Event to a node map . . . . .	616
CEventAdapterU3V	Connects a U3V Event to a node map . . . . .	618
CEventPort	Port attachable to an event . . . . .	620
CFeatureBag	Bag holding streamable features of a nodetree . . . . .	625
CFloatPtr	SmartPointer for IFloat interface pointer . . . . .	628
CGeneric_XMLLoaderParams	Empty base class used by class CNodeMapRef as generic template argument . . . . .	630
CGlobalLock	Named global lock which can be used over process boundaries . . . . .	631
CGlobalLockUnlocker	Unlocks the global lock object on destruction . . . . .	633
ChunkData	The chunk data which contains additional information about an image . . . . .	635
CLock	A lock class . . . . .	648
CLock	A lock class . . . . .	649
CLockEx	This class is for testing purposes only . . . . .	652
CLockEx	This class is for testing purposes only . . . . .	653
CNodeCallback	Callback body instance for INode pointers . . . . .	654
CNodeMapFactory	The node map factory is used for creating node maps from camera description files . . . . .	657
CNodeMapRef	Smartpointer for NodeMaps with create function . . . . .	665
CNodeMapRefT< TCameraParams >	Smartpointer template for NodeMaps with create function . . . . .	668
CommandNode	Interface for string properties . . . . .	674
Counter	Definition of a simple Counter class . . . . .	677
CPointer< T, B >	Encapsulates a GenApi pointer dealing with the dynamic_cast automatically . . . . .	679
CPortImpl	Standard implementation for a port . . . . .	684
CPortWriteList	Container holding a list of port write commands . . . . .	687
CpuUsageInfo	. . . . .	690
CRegisterPortImpl	Standard implementation for a port using a register based transport layer . . . . .	691

<b>CSelectorSet</b>	
The set of selectors selecting a given node . . . . .	694
<b>CTestPortStruct&lt; CDataStruct &gt;</b>	
Implements a register spaces based on a C++ struct . . . . .	697
<b>DCAM_CHECKSUM</b>	
DCAM_CHUNK_TRAILER . . . . .	701
<b>DeviceArrivalEventHandler</b>	
An event handler for capturing the device arrival event . . . . .	702
<b>DeviceEventHandler</b>	
A handler to device events . . . . .	705
<b>DeviceEventHandlerImpl</b>	
DeviceEvent . . . . .	708
<b>DeviceRemovalEventHandler</b>	
An event handler for capturing the device removal event . . . . .	710
<b>double_automvector_t</b>	
Vector of doubles with reference counting . . . . .	712
<b>EAccessModeClass</b>	
Holds conversion methods for the access mode enumeration . . . . .	715
<b>ECachingModeClass</b>	
Holds conversion methods for the caching mode enumeration . . . . .	716
<b>EDisplayNotationClass</b>	
Holds conversion methods for the notation type of floats . . . . .	717
<b>EEndianessClass</b>	
Holds conversion methods for the endianess enumeration . . . . .	718
<b>EGenApiSchemaVersionClass</b>	
Helper class converting EGenApiSchemaVersion from and to string . . . . .	720
<b>EInputDirectionClass</b>	
Holds conversion methods for the notation type of floats . . . . .	721
<b>ENamespaceClass</b>	
Holds conversion methods for the namespace enumeration . . . . .	722
<b>EnumEntryNode</b>	
Interface for string properties . . . . .	723
<b>EnumNode</b>	
Interface for string properties . . . . .	726
<b>ERepresentationClass</b>	
Holds conversion methods for the representation enumeration . . . . .	731
<b>ESignClass</b>	
Holds conversion methods for the sign enumeration . . . . .	732
<b>ESlopeClass</b>	
Holds conversion methods for the converter formulas . . . . .	733
<b>EStandardNameSpaceClass</b>	
Holds conversion methods for the standard namespace enumeration . . . . .	734
<b>EventHandler</b>	
The base class for all event handler types . . . . .	736
<b>EVisibilityClass</b>	
Holds conversion methods for the visibility enumeration . . . . .	739
<b>Exception</b>	
The <b>Exception</b> object represents an error that is returned from the library . . . . .	740
<b>EYesNoClass</b>	
Holds conversion methods for the standard namespace enumeration . . . . .	745
<b>FileProtocolAdapter</b>	
Adapter between the std::iostreambuf and the SFNC Features representing the device file system . . . . .	746
<b>FloatNode</b>	
Interface for string properties . . . . .	750
<b>FloatRegNode</b>	
Interface for string properties . . . . .	757
<b>Function_NodeCallback&lt; Function &gt;</b>	
Container for a function pointer . . . . .	760

gcstring . . . . .	762
GrabInfo . . . . .	771
GVCP_CHUNK_TRAILER Header of a GVCP request packet . . . . .	772
GVCP_EVENT_ITEM Layout of a GVCP event item (Extended ID flag not set) . . . . .	773
GVCP_EVENT_ITEM_BASIC Layout of a GVCP event item (common to all types) . . . . .	774
GVCP_EVENT_ITEM_EXTENDED_ID Layout of a GVCP event item (Extended ID flag set) . . . . .	775
GVCP_EVENT_REQUEST Layout of a GVCP event request packet (Extended ID flag not set) . . . . .	777
GVCP_EVENT_REQUEST_EXTENDED_ID Layout of a GVCP event request packet (Extended ID flag set) . . . . .	778
GVCP_EVENTDATA_REQUEST Layout of a GVCP event data request packet (Extended ID flag not set) . . . . .	779
GVCP_EVENTDATA_REQUEST_EXTENDED_ID Layout of a GVCP event data request packet (Extended ID flag set) . . . . .	780
GVCP_REQUEST_HEADER Header of a GVCP request packet . . . . .	781
H264Option Options for saving H264 files . . . . .	782
ICameraBase The interface file for base class for the camera object . . . . .	784
ICameraList Used to hold a list of camera objects . . . . .	793
IChunkData The <a href="#">Interface</a> file for <a href="#">ChunkData</a> . . . . .	797
IDataStream . . . . .	806
IDevFileStreamBase< CharType, Traits > . . . . .	812
IDevFileStreamBuf< CharType, Traits > . . . . .	814
IDeviceArrivalEventHandler . . . . .	817
IDeviceEventHandler . . . . .	819
IDeviceRemovalEventHandler . . . . .	821
IIImage The interface file for <a href="#">Image</a> . . . . .	823
IIImageEventHandler . . . . .	835
IIImageStatistics The interface file for image statistics . . . . .	837
IIInterface The interface file for <a href="#">Interface</a> . . . . .	841
IIInterfaceArrivalEventHandler . . . . .	846
IIInterfaceEventHandler . . . . .	848
IIInterfaceList The interface file for <a href="#">InterfaceList</a> class . . . . .	851
IIInterfaceRemovalEventHandler . . . . .	853
ILoggingEventHandler . . . . .	855
Image The image object class . . . . .	857
ImageEventHandler A handler for capturing image arrival events . . . . .	882
ImageEventHandlerImpl . . . . .	885
ImagePtr A reference tracked pointer to an image object . . . . .	888
ImageStatistics Represents image statistics for an image . . . . .	890
ImageUtility Static helper functions for the image object class . . . . .	897

<a href="#">ImageUtilityHeatmap</a>	
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16	902
<a href="#">ImageUtilityPolarization</a>	
Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8	906
<a href="#">InferenceBoundingBox</a>	
Inference Bounding Boxes data structure	913
<a href="#">InferenceBoundingBoxResult</a>	
An inference bounding boxes object which holds information about the detected bounding boxes	915
<a href="#">InferenceBoxCircle</a>	
<a href="#">InferenceBoxRect</a>	
Inference Bounding Box Type Data Structures	918
<a href="#">InferenceBoxRotatedRect</a>	
<a href="#">int64_autovector_t</a>	
Vector of integers with reference counting	920
<a href="#">IntegerNode</a>	
<a href="#">Interface</a> for string properties	923
<a href="#">Interface</a>	
An interface object which holds a list of cameras	929
<a href="#">InterfaceArrivalEventHandler</a>	
An event handler for capturing the interface arrival event	934
<a href="#">InterfaceEventHandler</a>	
A handler to device arrival and removal events on all interfaces	936
<a href="#">InterfaceEventHandlerImpl</a>	
<a href="#">InterfaceList</a>	
A list of the available interfaces on the system	943
<a href="#">InterfacePtr</a>	
A reference tracked pointer to the interface object	947
<a href="#">InterfaceRemovalEventHandler</a>	
An event handler for capturing the interface removal event	949
<a href="#">IntRegNode</a>	
<a href="#">Interface</a> for string properties	951
<a href="#">IpInfo</a>	
<a href="#">ISystem</a>	
The interface file for <a href="#">System</a>	955
<a href="#">ISystemEventHandler</a>	
<a href="#">JPEGOption</a>	
Options for saving JPEG image	963
<a href="#">JPG2Option</a>	
Options for saving JPEG2000 image	964
<a href="#">LibraryVersion</a>	
Provides easier access to the current version of <a href="#">Spinnaker</a>	966
<a href="#">LockableObject&lt; Object &gt;::Lock</a>	
A scopelevel <a href="#">Lock</a> class	967
<a href="#">LockableObject&lt; Object &gt;</a>	
Instance-Lock for an object	968
<a href="#">LoggingEventData</a>	
The <a href="#">LoggingEventData</a> object	969
<a href="#">LoggingEventDataPtr</a>	
A reference tracked pointer to the LoggingEvent object	973
<a href="#">LoggingEventHandler</a>	
An event handler for capturing the device logging event	975
<a href="#">LoggingEventHandlerImpl</a>	
<a href="#">Member_NodeCallback&lt; Client, Member &gt;</a>	
Container for a member function pointer	978
<a href="#">MJPGOption</a>	
Options for saving MJPG files	980

<b>Node</b>	Class common to all nodes . . . . .	982
<b>NodeMap</b>	Smart pointer template for NodeMaps with create function . . . . .	993
CNodeMapFactory::NodeStatistics_t		1002
ODevFileStreamBase< CharType, Traits >		1003
ODevFileStreamBuf< CharType, Traits >		1005
<b>PGMOption</b>	Options for saving PGM images . . . . .	1007
<b>PNGOption</b>	Options for saving PNG images . . . . .	1008
<b>PortNode</b>	Interface for value properties . . . . .	1010
<b>PortRecorder</b>	Interface for recording write commands on a port . . . . .	1016
<b>PortReplay</b>	Interface for replaying write commands on a port . . . . .	1020
<b>PPMOption</b>	Options for saving PPM images . . . . .	1023
<b>RegisterNode</b>	Interface for string properties . . . . .	1024
SingleChunkData_t		1029
SingleChunkDataStr_t		1030
SpinTestCamera		1031
<b>SpinVideo</b>	Provides the functionality for the user to record images to an AVI/MP4 file . . . . .	1031
<b>StringNode</b>	Interface for string properties . . . . .	1036
<b>StringRegNode</b>	Interface for string properties . . . . .	1040
<b>System</b>	The system object is used to retrieve the list of interfaces and cameras available . . . . .	1043
<b>SystemEventHandler</b>	A handler to interface arrival and removal events on the system . . . . .	1053
<b>SystemEventHandlerImpl</b>		1056
<b>SystemPtr</b>	A reference tracked pointer to a system object . . . . .	1059
<b>TIFFOption</b>	Options for saving TIFF images . . . . .	1061
<b>TransportLayerDevice</b>	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . . .	1063
<b>TransportLayerInterface</b>	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . . .	1073
<b>TransportLayerStream</b>	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . . .	1084
<b>TransportLayerSystem</b>	Part of the QuickSpin API to provide access to camera information without having to first initialize the camera . . . . .	1092
<b>U3V_CHUNK_TRAILER</b>	Header of a GVCP request packet . . . . .	1099
<b>U3V_COMMAND_HEADER</b>	U3V/GenCP command header . . . . .	1099
<b>U3V_EVENT_DATA</b>	U3V/GenCP EVENT_CMD specific command data . . . . .	1100

<a href="#">U3V_EVENT_MESSAGE</a>	
Entire event data message (without the variable-sized data field)	1101
<a href="#">ValueNode</a>	
<a href="#">Interface</a> for value properties	1102
<a href="#">Version_t</a>	
Version	1105



# Chapter 11

## File Index

### 11.1 File List

Here is a list of all files with brief descriptions:

include/AdapterConfig.h	1107
include/AVIRecorder.h	1109
include/BasePtr.h	1110
include/Camera.h	1111
include/CameraBase.h	1111
include/CameraDefs.h	1112
include/CameraList.h	1144
include/CameraPtr.h	1145
include/ChunkData.h	1145
include/ChunkDataInference.h	1146
include/DeviceArrivalEventHandler.h	1147
include/DeviceEventHandler.h	1148
include/DeviceRemovalEventHandler.h	1149
include/EventHandler.h	1150
include/Exception.h	1150
include/IImage.h	1151
include/IImageEventHandler.h	1152
include/IImagePtr.h	1152
include/IImageStatistics.h	1153
include/IImageUtility.h	1154
include/IImageUtilityHeatmap.h	1155
include/IImageUtilityPolarization.h	1156
include/IInterface.h	1157
include/IInterfaceArrivalEventHandler.h	1172
include/IInterfaceEventHandler.h	1173
include/IInterfaceList.h	1174
include/IInterfacePtr.h	1174
include/IInterfaceRemovalEventHandler.h	1175
include/LoggingEventData.h	1176
include/LoggingEventDataPtr.h	1176
include/LoggingEventHandler.h	1177
include/Spinnaker.h	1265
include/SpinnakerDefs.h	1265
include/SpinnakerPlatform.h	1269
include/SpinUpdate.h	1270

include/SpinVideo.h . . . . .	1273
include/SpinVideoDefs.h . . . . .	1274
include/System.h . . . . .	1275
include/SystemEventHandler.h . . . . .	1276
include/SystemPtr.h . . . . .	1277
include/TransportLayerDefs.h . . . . .	1277
include/TransportLayerDevice.h . . . . .	1279
include/TransportLayerInterface.h . . . . .	1280
include/TransportLayerStream.h . . . . .	1280
include/TransportLayerSystem.h . . . . .	1281
include/Interface/ICameraBase.h . . . . .	1157
include/Interface/ICameraList.h . . . . .	1158
include/Interface/IChunkData.h . . . . .	1159
include/Interface/IDeviceArrivalEventHandler.h . . . . .	1160
include/Interface/IDeviceEventHandler.h . . . . .	1161
include/Interface/IDeviceRemovalEventHandler.h . . . . .	1162
include/Interface/IImage.h . . . . .	1163
include/Interface/IImageEventHandler.h . . . . .	1163
include/Interface/IImageStatistics.h . . . . .	1164
include/Interface/IInterface.h . . . . .	1165
include/Interface/IInterfaceArrivalEventHandler.h . . . . .	1166
include/Interface/IInterfaceEventHandler.h . . . . .	1167
include/Interface/IInterfaceList.h . . . . .	1167
include/Interface/IInterfaceRemovalEventHandler.h . . . . .	1168
include/Interface/ILoggingEventHandler.h . . . . .	1169
include/Interface/IStream.h . . . . .	1169
include/Interface/ISystem.h . . . . .	1170
include/Interface/ISystemEventHandler.h . . . . .	1171
include/SpinGenApi/Autovector.h . . . . .	1178
include/SpinGenApi/Base.h . . . . .	1179
include/SpinGenApi/BooleanNode.h . . . . .	1180
include/SpinGenApi/CategoryNode.h . . . . .	1181
include/SpinGenApi/ChunkAdapter.h . . . . .	1182
include/SpinGenApi/ChunkAdapterDcam.h . . . . .	1183
include/SpinGenApi/ChunkAdapterGeneric.h . . . . .	1184
include/SpinGenApi/ChunkAdapterGEV.h . . . . .	1185
include/SpinGenApi/ChunkAdapterU3V.h . . . . .	1186
include/SpinGenApi/ChunkPort.h . . . . .	1187
include/SpinGenApi/CommandNode.h . . . . .	1187
include/SpinGenApi/Compatibility.h . . . . .	1188
include/SpinGenApi/Container.h . . . . .	1189
include/SpinGenApi/Counter.h . . . . .	1189
include/SpinGenApi/EnumClasses.h . . . . .	1190
include/SpinGenApi/EnumEntryNode.h . . . . .	1191
include/SpinGenApi/EnumNode.h . . . . .	1192
include/SpinGenApi/EnumNodeT.h . . . . .	1193
include/SpinGenApi/EventAdapter.h . . . . .	1193
include/SpinGenApi/EventAdapter1394.h . . . . .	1194
include/SpinGenApi/EventAdapterGeneric.h . . . . .	1195
include/SpinGenApi/EventAdapterGEV.h . . . . .	1195
include/SpinGenApi/EventAdapterU3V.h . . . . .	1196
include/SpinGenApi/EventPort.h . . . . .	1197
include/SpinGenApi/Filestream.h . . . . .	1198
include/SpinGenApi/FloatNode.h . . . . .	1199
include/SpinGenApi/FloatRegNode.h . . . . .	1200
include/SpinGenApi/GCBase.h . . . . .	1201
include/SpinGenApi/GCString.h . . . . .	1201
include/SpinGenApi/GCStringVector.h . . . . .	1203

include/SpinGenApi/GCSynch.h	1204
include/SpinGenApi/GCTypes.h	1205
include/SpinGenApi/GCUtilities.h	1208
include/SpinGenApi/IBoolean.h	1212
include/SpinGenApi/ICategory.h	1213
include/SpinGenApi/IChunkPort.h	1214
include/SpinGenApi/ICommand.h	1216
include/SpinGenApi/IDestroy.h	1217
include/SpinGenApi/IDeviceInfo.h	1218
include/SpinGenApi/IEnumEntry.h	1219
include/SpinGenApi/IEnumeration.h	1220
include/SpinGenApi/IEnumerationT.h	1221
include/SpinGenApi/IFloat.h	1222
include/SpinGenApi/IInteger.h	1224
include/SpinGenApi/INode.h	1225
include/SpinGenApi/INodeMap.h	1228
include/SpinGenApi/INodeMapDyn.h	1230
include/SpinGenApi/IntegerNode.h	1231
include/SpinGenApi/IntRegNode.h	1232
include/SpinGenApi/IPort.h	1233
include/SpinGenApi/IPortConstruct.h	1234
include/SpinGenApi/IPortRecorder.h	1235
include/SpinGenApi/IRegister.h	1236
include/SpinGenApi/ISelector.h	1237
include/SpinGenApi/ISelectorDigit.h	1238
include/SpinGenApi/IString.h	1239
include/SpinGenApi/IValue.h	1240
include/SpinGenApi/Node.h	1241
include/SpinGenApi/NodeCallback.h	1242
include/SpinGenApi/NodeCallbackImpl.h	1244
include/SpinGenApi/NodeMap.h	1244
include/SpinGenApi/NodeMapFactory.h	1245
include/SpinGenApi/NodeMapRef.h	1246
include/SpinGenApi/Persistence.h	1247
include/SpinGenApi/Pointer.h	1248
include/SpinGenApi/PortImpl.h	1250
include/SpinGenApi/PortNode.h	1251
include/SpinGenApi/PortRecorder.h	1252
include/SpinGenApi/PortReplay.h	1252
include/SpinGenApi/PortWriteList.h	1253
include/SpinGenApi/Reference.h	1254
include/SpinGenApi/RegisterNode.h	1255
include/SpinGenApi/RegisterPortImpl.h	1256
include/SpinGenApi/SelectorSet.h	1256
include/SpinGenApi/SpinnakerGenApi.h	1257
include/SpinGenApi/SpinTestCamera.h	1257
include/SpinGenApi/StringNode.h	1258
include/SpinGenApi/StringRegNode.h	1259
include/SpinGenApi/StructPort.h	1259
include/SpinGenApi/Synch.h	1260
include/SpinGenApi/Types.h	1261
include/SpinGenApi/ValueNode.h	1264
src/Acquisition/ <a href="#">Acquisition.cpp</a>	1281
src/Acquisition/ <a href="#">resource.h</a>	1284
src/Acquisition/ <a href="#">stdafx.cpp</a>	1284
src/Acquisition/ <a href="#">stdafx.h</a>	1293
src/Acquisition/ <a href="#">targetver.h</a>	1309
src/AcquisitionMultipleCameraRecovery/ <a href="#">AcquisitionMultipleCameraRecovery.cpp</a>	1324

src/AcquisitionMultipleCameraRecovery/resource.h . . . . .	1284
src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp . . . . .	1326
src/AcquisitionMultipleThread/resource.h . . . . .	1284
src/ActionCommand/ActionCommand.cpp . . . . .	1327
src/ActionCommand/resource.h . . . . .	1284
src/ActionCommand/stdafx.cpp . . . . .	1285
src/ActionCommand/stdafx.h . . . . .	1294
src/ActionCommand/targetver.h . . . . .	1310
src/BufferHandling/BufferHandling.cpp . . . . .	1329
src/BufferHandling/resource.h . . . . .	1284
src/BufferHandling/stdafx.cpp . . . . .	1285
src/BufferHandling/stdafx.h . . . . .	1295
src/BufferHandling/targetver.h . . . . .	1311
src/ChunkData/ChunkData.cpp . . . . .	1332
src/ChunkData/resource.h . . . . .	1284
src/CounterAndTimer/CounterAndTimer.cpp . . . . .	1334
src/CounterAndTimer/resource.h . . . . .	1284
src/CounterAndTimer/stdafx.cpp . . . . .	1286
src/CounterAndTimer/stdafx.h . . . . .	1296
src/CounterAndTimer/targetver.h . . . . .	1312
src/DeviceEvents/DeviceEvents.cpp . . . . .	1336
src/DeviceEvents/resource.h . . . . .	1284
src/DeviceEvents/stdafx.cpp . . . . .	1286
src/DeviceEvents/stdafx.h . . . . .	1297
src/DeviceEvents/targetver.h . . . . .	1313
src/Enumeration/Enumeration.cpp . . . . .	1338
src/Enumeration/resource.h . . . . .	1284
src/Enumeration/stdafx.cpp . . . . .	1287
src/Enumeration/stdafx.h . . . . .	1298
src/Enumeration/targetver.h . . . . .	1314
src/Enumeration_QuickSpin/Enumeration_QuickSpin.cpp . . . . .	1339
src/Enumeration_QuickSpin/resource.h . . . . .	1284
src/Enumeration_QuickSpin/stdafx.cpp . . . . .	1287
src/Enumeration_QuickSpin/stdafx.h . . . . .	1299
src/Enumeration_QuickSpin/targetver.h . . . . .	1315
src/EnumerationEvents/EnumerationEvents.cpp . . . . .	1339
src/EnumerationEvents/resource.h . . . . .	1284
src/ExceptionHandling/ExceptionHandling.cpp . . . . .	1340
src/ExceptionHandling/resource.h . . . . .	1284
src/ExceptionHandling/stdafx.cpp . . . . .	1288
src/ExceptionHandling/stdafx.h . . . . .	1300
src/ExceptionHandling/targetver.h . . . . .	1316
src/Exposure/Exposure.cpp . . . . .	1342
src/Exposure/resource.h . . . . .	1284
src/Exposure/stdafx.cpp . . . . .	1288
src/Exposure/stdafx.h . . . . .	1301
src/Exposure/targetver.h . . . . .	1317
src/Exposure_QuickSpin/Exposure_QuickSpin.cpp . . . . .	1343
src/Exposure_QuickSpin/resource.h . . . . .	1284
src/Exposure_QuickSpin/stdafx.cpp . . . . .	1289
src/Exposure_QuickSpin/stdafx.h . . . . .	1302
src/Exposure_QuickSpin/targetver.h . . . . .	1318
src/FileAccess_QuickSpin/FileAccess_QuickSpin.cpp . . . . .	1345
src/FileAccess_QuickSpin/resource.h . . . . .	1284
src/FileAccess_QuickSpin/stdafx.cpp . . . . .	1290
src/FileAccess_QuickSpin/stdafx.h . . . . .	1303
src/FileAccess_QuickSpin/targetver.h . . . . .	1319
src/GenTLInfo_QuickSpin/GenTLInfo_QuickSpin.cpp . . . . .	1348

src/GenTLInfo_QuickSpin/resource.h . . . . .	1284
src/GenTLInfo_QuickSpin/targetver.h . . . . .	1321
src/GigEVisionPerformance/CpuUtil.cpp . . . . .	1349
src/GigEVisionPerformance/CpuUtil.h . . . . .	1350
src/GigEVisionPerformance/GigEVisionPerformance.cpp . . . . .	1351
src/GigEVisionPerformance/GigEVisionPerformance.h . . . . .	1356
src/GigEVisionPerformance/resource.h . . . . .	1284
src/GigEVisionPerformance/stdafx.cpp . . . . .	1291
src/GigEVisionPerformance/stdafx.h . . . . .	1304
src/GigEVisionPerformance/targetver.h . . . . .	1320
src/HighDynamicRange/HighDynamicRange.cpp . . . . .	1357
src/HighDynamicRange/resource.h . . . . .	1284
src/ImageEvents/ImageEvents.cpp . . . . .	1360
src/ImageEvents/resource.h . . . . .	1284
src/ImageFormatControl/ImageFormatControl.cpp . . . . .	1362
src/ImageFormatControl/resource.h . . . . .	1284
src/ImageFormatControl/stdafx.h . . . . .	1305
src/ImageFormatControl_QuickSpin/ImageFormatControl_QuickSpin.cpp . . . . .	1363
src/ImageFormatControl_QuickSpin/resource.h . . . . .	1284
src/ImageFormatControl_QuickSpin/stdafx.h . . . . .	1305
src/Inference/Inference.cpp . . . . .	1364
src/Inference/resource.h . . . . .	1284
src/Logging/Logging.cpp . . . . .	1371
src/Logging/resource.h . . . . .	1284
src/LogicBlock/LogicBlock.cpp . . . . .	1372
src/LogicBlock/resource.h . . . . .	1284
src/LookupTable/LookupTable.cpp . . . . .	1374
src/LookupTable/resource.h . . . . .	1284
src/NodeMapCallback/NodeMapCallback.cpp . . . . .	1375
src/NodeMapCallback/resource.h . . . . .	1284
src/NodeMapInfo/NodeMapInfo.cpp . . . . .	1377
src/NodeMapInfo/resource.h . . . . .	1284
src/NodeMapInfo/stdafx.cpp . . . . .	1291
src/NodeMapInfo/stdafx.h . . . . .	1306
src/NodeMapInfo/targetver.h . . . . .	1321
src/Polarization/Polarization.cpp . . . . .	1381
src/Polarization/resource.h . . . . .	1284
src/Polarization/stdafx.h . . . . .	1307
src/SaveToAvi/resource.h . . . . .	1284
src/SaveToAvi/SaveToAvi.cpp . . . . .	1384
src/Sequencer/resource.h . . . . .	1284
src/Sequencer/Sequencer.cpp . . . . .	1386
src/Sequencer/stdafx.cpp . . . . .	1292
src/Sequencer/stdafx.h . . . . .	1307
src/Sequencer/targetver.h . . . . .	1322
src/SerialRxTx/resource.h . . . . .	1284
src/SerialRxTx/SerialRxTx.cpp . . . . .	1388
src/SerialRxTx/stdafx.cpp . . . . .	1292
src/SerialRxTx/stdafx.h . . . . .	1308
src/SerialRxTx/targetver.h . . . . .	1323
src/Trigger/resource.h . . . . .	1284
src/Trigger/Trigger.cpp . . . . .	1391
src/Trigger_QuickSpin/resource.h . . . . .	1284
src/Trigger_QuickSpin/Trigger_QuickSpin.cpp . . . . .	1394



## **Chapter 12**

# **Module Documentation**

### **12.1 Spinnaker Classes**

## 12.2 AVI Recorder Class

## 12.3 BasePtr Class

## 12.4 Camera Class

## 12.5 Camera Base Class

## 12.6 CameraDefs Class

## 12.7 Camera List Class

## 12.8 CameraPtr Class

## 12.9 ChunkData Class

## 12.10 Chunk Data Inference Class

## 12.11 Spinnaker EventHandler Classes

## 12.12 DeviceArrivalEventHandler Class

## 12.13 DeviceEventHandler Class

## 12.14 DeviceRemovalEventHandler Class

## 12.15 EventHandler Class

## 12.16 Exception Class

## 12.17 Image Class

## 12.18 ImageEventHandler Class

## 12.19 ImagePtr Class

## 12.20 ImageStatistics Class

## 12.21 Image Utility Class

## 12.22 Image Utility Heatmap Class

## 12.23 Image Utility Polarization Class

## 12.24 Interface Class

## 12.25 InterfaceArrivalEventHandler Class

## 12.26 InterfaceEventHandler Class

## 12.27 InterfaceList Class

## 12.28 InterfacePtr Class

## 12.29 InterfaceRemovalEventHandler Class

## 12.30 Logging EventHandler Class

## 12.31 LoggingEventDataPtr Class

## 12.32 LoggingEventHandler Class

## 12.33 Spinnaker Headers

## 12.34 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-sup>

## 12.35 Spinnaker Definitions

Definitions file for [Spinnaker](#).

Definitions file for [Spinnaker](#).

## 12.36 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.

## 12.37 Spinnaker Video Class

## 12.38 Spinnaker Video Definitions

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

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

## 12.39 System Class

## 12.40 SystemEventHandler Class

## 12.41 SystemPtr Class

## 12.42 Spinnaker QuickSpin Classes

## 12.43 TransportLayerDefs Class

## 12.44 TransportLayerDevice Class

## 12.45 TransportLayerInterface Class

## 12.46 TransportLayerStream Class

## 12.47 TransportLayerSystem Class

## 12.48 Camera Base Interface Class

## 12.49 IChunkData Class

## 12.50 IImage Class

## 12.51 IImageStatistics Class

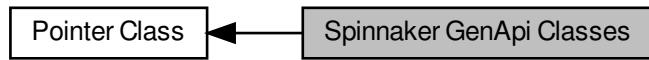
## 12.52 IInterface Class

## 12.53 **IInterfaceList** Class

## 12.54 ISystem Class

## 12.55 Spinnaker GenApi Classes

Collaboration diagram for Spinnaker GenApi Classes:



## 12.56 AutoVector Class

## 12.57 Spinnaker GenApi Interfaces

## 12.58 IBase Interface

## 12.59 BooleanNode Class

## 12.60 CategoryNode Class

## 12.61 ChunkAdapter Class

## 12.62 ChunkAdapterDcam Class

## 12.63 ChunkAdapterGeneric Class

## 12.64 ChunkAdapterGEV Class

## 12.65 ChunkPort Class

## 12.66 CommandNode Class

## 12.67 Container Class

## 12.68 Counter Class

## 12.69 **EnumClasses** Class

## 12.70 **EnumEntryNode Class**

## 12.71 EnumNode Class

## 12.72 **EnumNodeT Class**

## 12.73 EventAdapter Class

## 12.74 EventAdapter1394 Class

## 12.75 EventAdapterGeneric Class

## 12.76 EventAdapterGEV Class

## 12.77 EventAdapterU3V Class

## 12.78 EventPort Class

## 12.79 Filestream Class

## 12.80 **FloatNode Class**

## 12.81 **FloatRegNode** Class

## 12.82 GCString Class

## 12.83 GCSynch Class

## 12.84 GCTypes Class

### Typedefs

- `typedef float float32_t`  
*32 bit floating point*
- `typedef double float64_t`  
*64 bit floating point*

#### 12.84.1 Detailed Description

#### 12.84.2 Typedef Documentation

##### 12.84.2.1 `float32_t`

```
typedef float float32_t
```

32 bit floating point

##### 12.84.2.2 `float64_t`

```
typedef double float64_t
```

64 bit floating point

## 12.85 Spinnaker GenApi Utilities

## 12.86 GCUtilities Utility

## 12.87 IBoolean Interface

## 12.88 ICategory Interfaces

## 12.89 IChunkPort Interface

## 12.90 ICommand Interface

## 12.91 IDestroy Interface

## 12.92 IDeviceInfo Interface

## 12.93 IEnumEntry Interface

## 12.94 IEnumeration Interface

## 12.95 IEnumerationT Interface

## 12.96 IFloat Interface

## 12.97 IInteger Interface

## 12.98 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.*

### 12.98.1 Detailed Description

### 12.98.2 Function Documentation

**12.98.2.1 Combine() [1/3]**

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

Computes which access mode the two guards allow together.

**12.98.2.2 Combine() [2/3]**

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

Computes which CachingMode results from a combination.

**12.98.2.3 Combine() [3/3]**

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

Computes which visibility the two guards allow together.

**12.98.2.4 IsAvailable() [1/3]**

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

Checks if a node is available.

**12.98.2.5 IsAvailable() [2/3]**

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

Checks if a node is available.

**12.98.2.6 IsAvailable() [3/3]**

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

Tests if available.

**12.98.2.7 IsCacheable()**

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

Tests Cacheability.

**12.98.2.8 IsImplemented() [1/3]**

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

Checks if a node is implemented.

**12.98.2.9 IsImplemented() [2/3]**

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

Checks if a node is implemented.

**12.98.2.10 IsImplemented() [3/3]**

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

Tests if implemented.

**12.98.2.11 IsReadable() [1/3]**

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

Checks if a node is readable.

**12.98.2.12 IsReadable() [2/3]**

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

Checks if a node is readable.

**12.98.2.13 IsReadable() [3/3]**

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

Tests if readable.

**12.98.2.14 IsVisible()**

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

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

**12.98.2.15 IsWritable() [1/3]**

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

Checks if a node is writable.

**12.98.2.16 IsWritable() [2/3]**

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

Checks if a node is writable.

**12.98.2.17 IsWritable() [3/3]**

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

Tests if writable.

## 12.99 INodeMap Interface

## 12.100 INodeMapDyn Interface

## 12.101 IntegerNode Class

## 12.102 IntRegNode Class

## 12.103 IPort Interface

## 12.104 IPortConstruct Interface

## 12.105 IPortRecorder Interface

## 12.106 IRegister Interfaces

## 12.107 ISelector Interface

## 12.108 ISelectorDigit Interface

## 12.109 IString Class

## 12.110 IValue Class

## 12.111 Node Class

## 12.112 NodeCallback Class

## 12.113 NodeMap Class

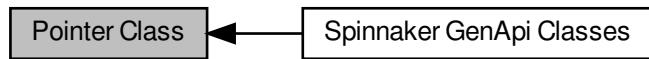
## 12.114 NodeMapFactory Class

## 12.115 NodeMapRef Class

## 12.116 Persistence Class

## 12.117 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< IPotConstruct > CPortConstructPtr**  
*SmartPointer for IPotConstruct 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.*

### 12.117.1 Detailed Description

### 12.117.2 Typedef Documentation

#### 12.117.2.1 CBasePtr

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

SmartPointer for IBase interface pointer.

### 12.117.2.2 CBooleanPtr

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

SmartPointer for IBoolean interface pointer.

### 12.117.2.3 CCategoryPtr

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

SmartPointer for ICategory interface pointer.

### 12.117.2.4 CChunkPortPtr

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

SmartPointer for IChunkPort interface pointer.

### 12.117.2.5 CCommandPtr

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

SmartPointer for ICommand interface pointer.

### 12.117.2.6 CDeviceInfoPtr

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

SmartPointer for IDeviceInfo interface pointer.

### 12.117.2.7 CEnumEntryPtr

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

SmartPointer for IEnumEntry interface pointer.

**12.117.2.8 CEnumerationPtr**

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

SmartPointer for IEnumeration interface pointer.

**12.117.2.9 CIIntegerPtr**

```
typedef CPointer<IIInteger> CIIntegerPtr
```

SmartPointer for IIInteger interface pointer.

**12.117.2.10 CNodeMapDynPtr**

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

SmartPointer for INodeMapDyn interface pointer.

**12.117.2.11 CNodeMapPtr**

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

SmartPointer for INodeMap interface pointer.

**12.117.2.12 CNodePtr**

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

SmartPointer for INode interface pointer.

**12.117.2.13 CPortConstructPtr**

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

SmartPointer for IPotConstruct interface pointer.

### 12.117.2.14 CPortPtr

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

SmartPointer for IPo

### 12.117.2.15 CPortRecorderPtr

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

SmartPointer for IPo

### 12.117.2.16 CPortReplayPtr

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

SmartPointer for IPo

### 12.117.2.17 CPortWriteListPtr

```
typedef CPointer<IPortWriteList, IPo
```

SmartPointer for IPo

### 12.117.2.18 CRegisterPtr

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

SmartPointer for IRegis

### 12.117.2.19 CSelectorPtr

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

SmartPointer for ISelec

### 12.117.2.20 CStringPtr

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

SmartPointer for IString interface pointer.

### 12.117.2.21 CValuePtr

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

SmartPointer for IValue interface pointer.

## 12.117.3 Function Documentation

### 12.117.3.1 GetInterfaceName()

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

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

### 12.117.3.2 IsAvailable()

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

Checks if a node is Available.

### 12.117.3.3 IsImplemented()

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

Checks if a node is Implemented.

### 12.117.3.4 IsReadable()

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

Checks if a node is readable.

### 12.117.3.5 IsWritable()

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

Checks if a node is Writable.

## 12.118 PortImpl Class

## 12.119 PortNode Class

## 12.120 PortRecorder Class

## 12.121 PortReplay Class

## 12.122 PortWriteList Class

## 12.123 Reference Interfaces

## 12.124 RegisterNode Class

## 12.125 RegisterPortImpl Class

## 12.126 SelectorSet Class

## 12.127 SpinTestCamera Class

## 12.128 **StringNode Class**

## 12.129 StringRegNode Class

## 12.130 StructPort Class

## 12.131 Synch Class

## 12.132 Spinnaker GenApi Enums

## 12.133 Types Enums

## 12.134 ValueNode Class

## 12.135 ChunkAdapterU3V Class



# Chapter 13

## Namespace Documentation

### 13.1 AdapterConfig Namespace Reference

#### Classes

- struct [AdapterInfo](#)
- struct [IpInfo](#)

#### Enumerations

- enum [AdapterConfigErr](#) {  
    IP\_ADDRESS\_INVALID,  
    IP\_ADDRESS\_IS\_NOT\_V4,  
    IP\_ADDRESS\_TOO\_LARGE,  
    IP\_ADDRESS\_TOO\_SMALL,  
    HOST\_ADDRESS\_ZERO,  
    SUBNET\_MASK\_INVALID,  
    VALID\_SUBNET\_NOT\_FOUND }

#### Functions

- [ADAPTERCONFIG\\_API](#) std::vector<[AdapterInfo](#)> [RetrieveAllAdapters](#) ()
- [ADAPTERCONFIG\\_API](#) void [AutoPopulateAdapterInfo](#) (std::vector<[AdapterInfo](#)> &adaptersToConfigure, const std::vector<[AdapterInfo](#)> &allAdapters)
- [ADAPTERCONFIG\\_API](#) void [AutoPopulateAdvancedProperties](#) (std::vector<[AdapterInfo](#)> &adaptersToConfigure)
- [ADAPTERCONFIG\\_API](#) void [PopulateAdapterIpInfo](#) ([IpInfo](#) startingIpInfo, std::vector<[AdapterInfo](#)> &adaptersToConfigure, const std::vector<[AdapterInfo](#)> &allAdapters)
- [ADAPTERCONFIG\\_API](#) void [ValidateIpAddress](#) (const std::string &ipAddr, unsigned int subnetMaskLength)
- [ADAPTERCONFIG\\_API](#) bool [IsValidIpAddress](#) (const std::string &ipAddr)
- [ADAPTERCONFIG\\_API](#) bool [IsValidSubnetMask](#) (const std::string &subnetMask)
- [ADAPTERCONFIG\\_API](#) bool [IsOnSameSubnet](#) (const std::string &ipAddrStr1, const std::string &ipAddrStr2, const unsigned int subnetMaskLength)
- [ADAPTERCONFIG\\_API](#) unsigned int [GetSubnetMaskLength](#) (const std::string &subnetMask)
- [ADAPTERCONFIG\\_API](#) std::string [GetEnumerationLogFileName](#) ()
- [ADAPTERCONFIG\\_API](#) std::string [GetConfigLogFileName](#) ()

- **ADAPTERCONFIG\_API** void `ConfigureAdapter` (`AdapterInfo` &adapter, bool configureIP, bool configureAdvancedProperties)
- **ADAPTERCONFIG\_API** unsigned int `GetAutoSubnetMaskLength` ()
- **ADAPTERCONFIG\_API** std::string `GetAutoSubnetMask` ()
- **ADAPTERCONFIG\_API** std::string `GetMaxIpAddress` ()
- **ADAPTERCONFIG\_API** std::string `GetMinIpAddress` ()
- **ADAPTERCONFIG\_API** std::string `GetAutoGigabitDesc` ()
- **ADAPTERCONFIG\_API** std::string `GetAuto10GDesc` ()
- **ADAPTERCONFIG\_API** std::string `GetAutoStartIp` ()

### 13.1.1 Enumeration Type Documentation

#### 13.1.1.1 AdapterConfigErr

```
enum AdapterConfigErr
```

Enumerator

IP_ADDRESS_INVALID	
IP_ADDRESS_IS_NOT_V4	
IP_ADDRESS_TOO_LARGE	
IP_ADDRESS_TOO_SMALL	
HOST_ADDRESS_ZERO	
SUBNET_MASK_INVALID	
VALID_SUBNET_NOT_FOUND	

### 13.1.2 Function Documentation

#### 13.1.2.1 AutoPopulateAdapterInfo()

```
ADAPTERCONFIG_API void AdapterConfig::AutoPopulateAdapterInfo (
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters )
```

#### 13.1.2.2 AutoPopulateAdvancedProperties()

```
ADAPTERCONFIG_API void AdapterConfig::AutoPopulateAdvancedProperties (
    std::vector< AdapterInfo > & adaptersToConfigure )
```

**13.1.2.3 ConfigureAdapter()**

```
ADAPTERCONFIG_API void AdapterConfig::ConfigureAdapter (
    AdapterInfo & adapter,
    bool configureIP,
    bool configureAdvancedProperties )
```

**13.1.2.4 GetAuto10GDesc()**

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAuto10GDesc ( )
```

**13.1.2.5 GetAutoGigabitDesc()**

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoGigabitDesc ( )
```

**13.1.2.6 GetAutoStartIp()**

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoStartIp ( )
```

**13.1.2.7 GetAutoSubnetMask()**

```
ADAPTERCONFIG_API std::string AdapterConfig::GetAutoSubnetMask ( )
```

**13.1.2.8 GetAutoSubnetMaskLength()**

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetAutoSubnetMaskLength ( )
```

**13.1.2.9 GetConfigLogFileName()**

```
ADAPTERCONFIG_API std::string AdapterConfig::GetConfigLogFileName ( )
```

### 13.1.2.10 GetEnumerationLogFileName()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetEnumerationLogFileName ( )
```

### 13.1.2.11 GetMaxIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMaxIpAddress ( )
```

### 13.1.2.12 GetMinIpAddress()

```
ADAPTERCONFIG_API std::string AdapterConfig::GetMinIpAddress ( )
```

### 13.1.2.13 GetSubnetMaskLength()

```
ADAPTERCONFIG_API unsigned int AdapterConfig::GetSubnetMaskLength ( const std::string & subnetMask )
```

### 13.1.2.14 IsOnSameSubnet()

```
ADAPTERCONFIG_API bool AdapterConfig::IsOnSameSubnet ( const std::string & ipAddrStr1, const std::string & ipAddrStr2, const unsigned int subnetMaskLength )
```

### 13.1.2.15 IsValidIpAddress()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidIpAddress ( const std::string & ipAddr )
```

### 13.1.2.16 IsValidSubnetMask()

```
ADAPTERCONFIG_API bool AdapterConfig::IsValidSubnetMask ( const std::string & subnetMask )
```

### 13.1.2.17 PopulateAdapterIpInfo()

```
ADAPTERCONFIG_API void AdapterConfig::PopulateAdapterIpInfo (
    IpInfo startingIpInfo,
    std::vector< AdapterInfo > & adaptersToConfigure,
    const std::vector< AdapterInfo > & allAdapters )
```

### 13.1.2.18 RetrieveAllAdapters()

```
ADAPTERCONFIG_API std::vector<AdapterInfo> AdapterConfig::RetrieveAllAdapters ()
```

### 13.1.2.19 ValidateIpAddress()

```
ADAPTERCONFIG_API void AdapterConfig::ValidateIpAddress (
    const std::string & ipAddr,
    unsigned int subnetMaskLength )
```

## 13.2 Conversion Namespace Reference

### Functions

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

### 13.2.1 Function Documentation

#### 13.2.1.1 NumToString() [1/3]

```
string NumToString (
    double number )
```

#### 13.2.1.2 NumToString() [2/3]

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

### 13.2.1.3 NumToCString() [3/3]

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

## 13.3 CpuUtil Namespace Reference

### Classes

- struct [CpuUsageInfo](#)

### Functions

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

### 13.3.1 Function Documentation

#### 13.3.1.1 GetCpuStats()

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

#### 13.3.1.2 StartCpuTracing()

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

#### 13.3.1.3 StopCpuTracing()

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

## 13.4 PerformanceCounter Namespace Reference

### Functions

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

## Variables

- double PCFreq
- \_\_int64 CounterStart

### 13.4.1 Function Documentation

#### 13.4.1.1 GetPerformanceCounter()

```
double GetPerformanceCounter ( )
```

#### 13.4.1.2 StartPerformanceCounter()

```
void StartPerformanceCounter ( )
```

### 13.4.2 Variable Documentation

#### 13.4.2.1 CounterStart

```
__int64 CounterStart
```

#### 13.4.2.2 PCFreq

```
double PCFreq
```

## 13.5 SecondsCounter Namespace Reference

### Functions

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

## Variables

- time\_t `startTime`
- time\_t `endTime`
- double `timeDiff`

### 13.5.1 Function Documentation

#### 13.5.1.1 `GetSecondsCounter()`

```
int GetSecondsCounter ( )
```

#### 13.5.1.2 `StartSecondsCounter()`

```
void StartSecondsCounter ( )
```

### 13.5.2 Variable Documentation

#### 13.5.2.1 `endTime`

```
time_t endTime
```

#### 13.5.2.2 `startTime`

```
time_t startTime
```

#### 13.5.2.3 `timeDiff`

```
double timeDiff
```

## 13.6 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.*
- class [ChunkData](#)  
*The chunk data which contains additional information about an image.*
- class [DeviceArrivalEventHandler](#)  
*An event handler for capturing the device arrival event.*
- class [DeviceEventHandler](#)  
*A handler to device events.*
- 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 [IImageStatistics](#)

- class [IInterface](#)

*The interface file for image statistics.*
- class [IInterfaceArrivalEventHandler](#)

*The interface file for [Interface](#).*
- class [IInterfaceEventHandler](#)
- class [IInterfaceList](#)

*The interface file for [InterfaceList](#) class.*
- class [IInterfaceRemovalEventHandler](#)
- class [ILoggingEventHandler](#)
- class [Image](#)

*The image object class.*
- class [ImageEventHandler](#)
- class [ImagePtr](#)

*A handler for capturing image arrival events.*
- class [ImageStatistics](#)

*Represents image statistics for an image.*
- class [ImageUtility](#)

*Static helper functions for the image object class.*
- 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](#)  
*Options for saving PNG images.*
  - struct [PPMOption](#)  
*Options for saving PPM images.*
  - 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_DEVICEREGISTERSENDIANCESS } }

• 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 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,
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_Deglitch,
    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_DEVICESTREAMCHANNELENDIANCESS }

• 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_BGRA32f,
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_CHUNKSCAN3DCORDINATESYSTEM }

• enum ChunkScan3dCoordinateSystemReferenceEnums {
    ChunkScan3dCoordinateSystemReference_Anchor,
    ChunkScan3dCoordinateSystemReference_Transformed,
    NUM_CHUNKSCAN3DCORDINATESYSTEMREFERENCE }

• enum ChunkScan3dCoordinateSelectorEnums {
    ChunkScan3dCoordinateSelector_CoordinateA,
    ChunkScan3dCoordinateSelector_CoordinateB,
    ChunkScan3dCoordinateSelector_CoordinateC,
    NUM_CHUNKSCAN3DCORDINATESELECTOR }

• 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 GevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum CIConfigurationEnums {
    CIConfiguration_Base,
    CIConfiguration_Medium,
    CIConfiguration_Full,
    CIConfiguration_DualBase,
    CIConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum CITimeSlotsCountEnums {
    CITimeSlotsCount_One,
    CITimeSlotsCount_Two,
    CITimeSlotsCount_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_CXP CONNECTIONTESTMODE }

• enum CxpPoCxpStatusEnums {
    CxpPoCxpStatus_Auto,
    CxpPoCxpStatus_Off,
    CxpPoCxpStatus_Tripped,
    NUM_CXPOCXPSTATUS }

• 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,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_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_LOGGING_EVENT,`
`SPINNAKER_EVENT_UNKNOWN,`
`SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }`

*Event types in Spinnaker.*

- enum `PixelFormatNamespaceID` {
`SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,`
`SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,`
`SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,`
`SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,`
`SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,`
`SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }`

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

- enum `ColorProcessingAlgorithm` {
`DEFAULT,`
`NO_COLOR_PROCESSING,`
`NEAREST_NEIGHBOR,`
`NEAREST_NEIGHBOR_AVG,`
`BILINEAR,`
`EDGE_SENSING,`
`HQ_LINEAR,`
`IPP,`
`DIRECTIONAL_FILTER,`
`RIGOROUS,`
`WEIGHTED_DIRECTIONAL_FILTER }`

*Color processing algorithms.*

- enum `ImageFileFormat` {
`FROM_FILE_EXT = -1,`
`PGM,`
`PPM,`
`BMP,`

```
JPEG,
JPEG2000,
TIFF,
PNG,
RAW,
JPEG12_C,
IMAGE_FILE_FORMAT_FORCE_32BITS = 0xFFFFFFFF }
```

*File formats to be used for saving images to disk.*

- enum `ImageStatus` {  
  `IMAGE_UNKNOWN_ERROR` = -1,  
  `IMAGE_NO_ERROR` = 0,  
  `IMAGE_CRC_CHECK_FAILED` = 1,  
  `IMAGE_DATA_OVERFLOW` = 2,  
  `IMAGE_MISSING_PACKETS` = 3,  
  `IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,  
  `IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,  
  `IMAGE_PACKETID_INCONSISTENT` = 6,  
  `IMAGE_MISSING_LEADER` = 7,  
  `IMAGE_MISSING_TRAILER` = 8,  
  `IMAGE_DATA_INCOMPLETE` = 9,  
  `IMAGE_INFO_INCONSISTENT` = 10,  
  `IMAGE_CHUNK_DATA_INVALID` = 11,  
  `IMAGE_NO_SYSTEM_RESOURCES` = 12 }

*Status of images returned from `GetNextImage()` call.*

- enum `StatisticsChannel` {  
  `GREY`,  
  `RED`,  
  `GREEN`,  
  `BLUE`,  
  `HUE`,  
  `SATURATION`,  
  `LIGHTNESS`,  
  `NUM_STATISTICS_CHANNELS` }

*Channels that allow statistics to be calculated.*

- enum `SpinnakerLogLevel` {  
  `LOG_LEVEL_OFF` = -1,  
  `LOG_LEVEL_FATAL` = 0,  
  `LOG_LEVEL_ALERT` = 100,  
  `LOG_LEVEL_CRIT` = 200,  
  `LOG_LEVEL_ERROR` = 300,  
  `LOG_LEVEL_WARN` = 400,  
  `LOG_LEVEL_NOTICE` = 500,  
  `LOG_LEVEL_INFO` = 600,  
  `LOG_LEVEL_DEBUG` = 700,  
  `LOG_LEVEL_NOTSET` = 800 }

*log levels*

- enum `PayloadTypeInfoIDs` {  
  `PAYLOAD_TYPE_UNKNOWN` = 0,  
  `PAYLOAD_TYPE_IMAGE` = 1,  
  `PAYLOAD_TYPE_RAW_DATA` = 2,  
  `PAYLOAD_TYPE_FILE` = 3,  
  `PAYLOAD_TYPE_CHUNK_DATA` = 4,  
  `PAYLOAD_TYPE_JPEG` = 5,  
  `PAYLOAD_TYPE_JPEG2000` = 6,  
  `PAYLOAD_TYPE_H264` = 7,  
  `PAYLOAD_TYPE_CHUNK_ONLY` = 8,  
  `PAYLOAD_TYPE_DEVICE_SPECIFIC` = 9,

- `PAYLOAD_TYPE_MULTI_PART` = 10,  
`PAYLOAD_TYPE_CUSTOM_ID` = 1000,  
`PAYLOAD_TYPE_EXTENDED_CHUNK` = 1001 }
- enum `ActionCommandStatus` {
 `ACTION_COMMAND_STATUS_OK` = 0,  
`ACTION_COMMAND_STATUS_NO_REF_TIME`,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE`,  
`ACTION_COMMAND_STATUS_ERROR` }

*Possible Status Codes Returned from Action Command.*

- enum `PixelFormatIntType` {
 `IntType_UINT8`,  
`IntType_INT8`,  
`IntType_UINT10`,  
`IntType_UINT10P`,  
`IntType_UINT10P`,  
`IntType_UINT12`,  
`IntType_UINT12P`,  
`IntType_UINT12P`,  
`IntType_UINT14`,  
`IntType_UINT16`,  
`IntType_INT16`,  
`IntType_FLOAT32`,  
`IntType_UNKNOWN` }

*Possible integer types and packing used in a pixel format.*

- enum `BufferOwnership` {
 `BUFFER_OWNERSHIP_SYSTEM`,  
`BUFFER_OWNERSHIP_USER` }
- 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 `StreamBufferCountModeEnum` {
 `StreamBufferCountMode_Manual`,  
`StreamBufferCountMode_Auto`,  
`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 GevCCPEnum {
GevCCP_EnumEntry_GevCCP_OpenAccess,
GevCCP_EnumEntry_GevCCP_ExclusiveAccess,
GevCCP_EnumEntry_GevCCP_ControlAccess,
NUMGEVCCP }

• enum GUIXMLLocationEnum {
GUIXMLLocation_Device,
GUIXMLLocation_Host,
NUMGUIXMLLOCATION }

• enum GenICamXMLLocationEnum {
GenICamXMLLocation_Device,
GenICamXMLLocation_Host,
NUMGENICAMXMLLOCATION }

• enum DeviceEndianessMechanismEnum {
DeviceEndianessMechanism_Legacy,
DeviceEndianessMechanism_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

### 13.6.1 Enumeration Type Documentation

#### 13.6.1.1 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         |  |

#### 13.6.1.2 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.                  |

**Enumerator**

|                                               |                                                                        |
|-----------------------------------------------|------------------------------------------------------------------------|
| 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_ACQUISITIONSTATUSSELECTOR                 |                                                                        |

**13.6.1.3 ActionCommandStatus**

```
enum ActionCommandStatus
```

Possible Status Codes Returned from Action Command.

**Enumerator**

|                                   |  |
|-----------------------------------|--|
| ACTION_COMMAND_STATUS_OK          |  |
| ACTION_COMMAND_STATUS_NO_REF_TIME |  |
| ACTION_COMMAND_STATUS_OVERFLOW    |  |
| ACTION_COMMAND_STATUS_ACTION_LATE |  |
| ACTION_COMMAND_STATUS_ERROR       |  |

**13.6.1.4 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 |                                 |

**13.6.1.5 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   |  |

**13.6.1.6 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 |                                           |

**13.6.1.7 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**

|                                          |  |
|------------------------------------------|--|
| AutoExposureControlPriority_Gain         |  |
| AutoExposureControlPriority_ExposureTime |  |
| NUM_AUTOEXPOSURECONTROLPRIORITY          |  |

**13.6.1.8 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

|                                     |  |
|-------------------------------------|--|
| AutoExposureLightingMode_AutoDetect |  |
| AutoExposureLightingMode_Backlight  |  |
| AutoExposureLightingMode_Frontlight |  |
| AutoExposureLightingMode_Normal     |  |
| NUM_AUTOEXPOSURELIGHTINGMODE        |  |

#### 13.6.1.9 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            |  |

#### 13.6.1.10 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        |                                                                                      |

### 13.6.1.11 BalanceRatioSelectorEnums

enum `BalanceRatioSelectorEnums`

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

#### Enumerator

|                                        |                                                                                                                 |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <code>BalanceRatioSelector_Red</code>  | Selects the red balance ratio control for adjustment. The red balance ratio is relative to the green channel.   |
| <code>BalanceRatioSelector_Blue</code> | Selects the blue balance ratio control for adjustment. The blue balance ratio is relative to the green channel. |
| <code>NUM_BALANCERATIOSELECTOR</code>  |                                                                                                                 |

### 13.6.1.12 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

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

### 13.6.1.13 BalanceWhiteAutoProfileEnums

enum `BalanceWhiteAutoProfileEnums`

< Selects the profile used by BalanceWhiteAuto.

#### Enumerator

|                                              |                                                                                       |
|----------------------------------------------|---------------------------------------------------------------------------------------|
| <code>BalanceWhiteAutoProfile_Indoor</code>  | Indoor auto white balance Profile. Can be used to compensate for artificial lighting. |
| <code>BalanceWhiteAutoProfile_Outdoor</code> | Outdoor auto white balance profile. Designed for scenes with natural lighting.        |
| <code>NUM_BALANCEWHITEAUTOPROFILE</code>     |                                                                                       |

### 13.6.1.14 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     |                                                                                                                                                  |

### 13.6.1.15 BinningSelectorEnums

enum [BinningSelectorEnums](#)

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

#### Enumerator

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

### 13.6.1.16 BinningVerticalModeEnums

enum [BinningVerticalModeEnums](#)

<

#### Enumerator

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

### 13.6.1.17 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 BlackLevel.                                                                            |
| <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>        |                                                                                                                                           |

### 13.6.1.18 BlackLevelAutoEnums

enum [BlackLevelAutoEnums](#)

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

Enumerator

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

### 13.6.1.19 BlackLevelSelectorEnums

enum [BlackLevelSelectorEnums](#)

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

Enumerator

|                                         |  |
|-----------------------------------------|--|
| <code>BlackLevelSelector_All</code>     |  |
| <code>BlackLevelSelector_Analog</code>  |  |
| <code>BlackLevelSelector_Digital</code> |  |
| <code>NUM_BLACKLEVELSELECTOR</code>     |  |

### 13.6.1.20 BufferOwnership

enum [BufferOwnership](#)

Enumerator

|                         |  |
|-------------------------|--|
| BUFFER_OWNERSHIP_SYSTEM |  |
| BUFFER_OWNERSHIP_USER   |  |

### 13.6.1.21 ChunkBlackLevelSelectorEnums

enum [ChunkBlackLevelSelectorEnums](#)

< Selects which black level to retrieve

Enumerator

|                             |  |
|-----------------------------|--|
| ChunkBlackLevelSelector_All |  |
| NUM_CHUNKBLACKLEVELSELECTOR |  |

### 13.6.1.22 ChunkCounterSelectorEnums

enum [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      |                        |

### 13.6.1.23 ChunkEncoderSelectorEnums

enum [ChunkEncoderSelectorEnums](#)

< Selects which Encoder to retrieve data from.

**Enumerator**

|                               |                             |
|-------------------------------|-----------------------------|
| ChunkEncoderSelector_Encoder0 | Selects the first Encoder.  |
| ChunkEncoderSelector_Encoder1 | Selects the first Encoder.  |
| ChunkEncoderSelector_Encoder2 | Selects the second Encoder. |
| NUM_CHUNKENCODERSELECTOR      |                             |

**13.6.1.24 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           |                                           |

**13.6.1.25 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         |                                        |

**13.6.1.26 ChunkGainSelectorEnums**

enum [ChunkGainSelectorEnums](#)

< Selects which gain to retrieve

**Enumerator**

|                         |  |
|-------------------------|--|
| ChunkGainSelector_All   |  |
| ChunkGainSelector_Red   |  |
| ChunkGainSelector_Green |  |
| ChunkGainSelector_Blue  |  |
| NUM_CHUNKGAINSELECTOR   |  |

**13.6.1.27 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         |                                                   |

**13.6.1.28 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                   |  |

**13.6.1.29 ChunkRegionIDEnums**

```
enum ChunkRegionIDEnums
```

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

## Enumerator

|                       |                                                |
|-----------------------|------------------------------------------------|
| ChunkRegionID_Region0 | <a href="#">Image</a> comes from the Region 0. |
| ChunkRegionID_Region1 | <a href="#">Image</a> comes from the Region 1. |
| ChunkRegionID_Region2 | <a href="#">Image</a> comes from the Region 2. |
| NUM_CHUNKREGIONID     |                                                |

**13.6.1.30 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          |                         |

### 13.6.1.31 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         |                                   |

### 13.6.1.32 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         |                                                     |

### 13.6.1.33 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_↔<br>Transformed | Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices. |
| NUM_CHUNKSCAN3DCOORDINATESYSTEMREFE<br>RENCE          |                                                                                                                                               |

### 13.6.1.34 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          |                           |

### 13.6.1.35 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        |                                                         |

### 13.6.1.36 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_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. |

## 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                   |                                                                                                                                                                                                                                                                                                             |

**13.6.1.37 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                      |  |

**13.6.1.38 ChunkSourceIDEnums**enum [ChunkSourceIDEnums](#)

&lt; Returns the identifier of Source that the image comes from.

**Enumerator**

|                       |                                                |
|-----------------------|------------------------------------------------|
| ChunkSourceID_Source0 | <a href="#">Image</a> comes from the Source 0. |
| ChunkSourceID_Source1 | <a href="#">Image</a> comes from the Source 1. |
| ChunkSourceID_Source2 | <a href="#">Image</a> comes from the Source 2. |
| NUM_CHUNKSOURCEID     |                                                |

**13.6.1.39 ChunkTimerSelectorEnums**enum [ChunkTimerSelectorEnums](#)

&lt; 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    |                           |

**13.6.1.40 ChunkTransferStreamIDEnums**enum [ChunkTransferStreamIDEnums](#)

&lt; 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     |                          |

**13.6.1.41 CIConfigurationEnums**

```
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

|                           |                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CIConfiguration_Base      | Standard base configuration described by the <a href="#">Camera</a> Link standard.                                                                                                                                                                                                                                                            |
| CIConfiguration_Medium    | Standard medium configuration described by the <a href="#">Camera</a> Link standard.                                                                                                                                                                                                                                                          |
| CIConfiguration_Full      | Standard full configuration described by the <a href="#">Camera</a> Link standard.                                                                                                                                                                                                                                                            |
| CIConfiguration_DualBase  | The camera streams the data from multiple taps (that do not fit in the standard base configuration) through two <a href="#">Camera</a> 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. |
| CIConfiguration_EightyBit | Standard 80-bit configuration with 10 taps of 8 bits or 8 taps of 10 bits, as described by the <a href="#">Camera</a> Link standard.                                                                                                                                                                                                          |
| NUM_CLCONFIGURATION       |                                                                                                                                                                                                                                                                                                                                               |

**13.6.1.42 CITimeSlotsCountEnums**

```
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

|                        |       |
|------------------------|-------|
| CITimeSlotsCount_One   | One   |
| CITimeSlotsCount_Two   | Two   |
| CITimeSlotsCount_Three | Three |
| NUM_CLTIMESLOTSCOUNT   |       |

### 13.6.1.43 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

|                             |                                                                                                                        |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------|
| DEFAULT                     | Default method.                                                                                                        |
| NO_COLOR_PROCESSING         | No color processing.                                                                                                   |
| NEAREST_NEIGHBOR            | Fastest but lowest quality. Equivalent to FLYCAPTURE_NEAREST_NEIGHBOR_FAST in FlyCapture.                              |
| NEAREST_NEIGHBOR_AVG        | Nearest Neighbor with averaged green pixels. Higher quality but slower compared to nearest neighbor without averaging. |
| BILINEAR                    | Weighted average of surrounding 4 pixels in a 2x2 neighborhood.                                                        |
| EDGE_SENSING                | Weights surrounding pixels based on localized edge orientation.                                                        |
| HQ_LINEAR                   | Well-balanced speed and quality.                                                                                       |
| IPP                         | Multi-threaded with similar results to edge sensing.                                                                   |
| DIRECTIONAL_FILTER          | Best quality but much faster than rigorous.                                                                            |
| RIGOROUS                    | Slowest but produces good results.                                                                                     |
| WEIGHTED_DIRECTIONAL_FILTER | Weighted pixel average from different directions.                                                                      |

### 13.6.1.44 ColorTransformationSelectorEnums

enum `ColorTransformationSelectorEnums`

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

Enumerator

|                                      |  |
|--------------------------------------|--|
| ColorTransformationSelector_RGBtoRGB |  |
| ColorTransformationSelector_RGBtoYUV |  |
| NUM_COLORTRANSFORMATIONSELECTOR      |  |

### 13.6.1.45 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     |  |

**13.6.1.46 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         |  |

**13.6.1.47 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   |

## Enumerator

|                                     |                  |
|-------------------------------------|------------------|
| 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              |                  |

**13.6.1.48 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         |  |

**13.6.1.49 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 |

## Enumerator

|                                     |                  |
|-------------------------------------|------------------|
| 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              |                  |

**13.6.1.50 CounterSelectorEnums**enum [CounterSelectorEnums](#)

&lt; Selects which counter to configure

## Enumerator

|                          |  |
|--------------------------|--|
| CounterSelector_Counter0 |  |
| CounterSelector_Counter1 |  |
| NUM_COUNTERSELECTOR      |  |

**13.6.1.51 CounterStatusEnums**enum [CounterStatusEnums](#)

&lt; 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                |                                                     |

### 13.6.1.52 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         |

### 13.6.1.53 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              |                  |

### 13.6.1.54 CxpConnectionTestModeEnums

enum [CxpConnectionTestModeEnums](#)

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

#### Enumerator

|                             |        |
|-----------------------------|--------|
| CxpConnectionTestMode_Off   | Off    |
| CxpConnectionTestMode_Mode1 | Mode 1 |
| NUM_CXP CONNECTIONTESTMODE  |        |

### 13.6.1.55 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).  |

## Enumerator

|                              |                                                                        |
|------------------------------|------------------------------------------------------------------------|
| 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     |                                                                        |

**13.6.1.56 CxpLinkConfigurationPreferredEnums**enum [CxpLinkConfigurationPreferredEnums](#)

&lt; 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).  |
| 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).  |

## Enumerator

|                                       |                                                      |
|---------------------------------------|------------------------------------------------------|
| 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     |                                                      |

**13.6.1.57 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).                                                                           |
| 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).                                                                           |

**Enumerator**

|                                    |                                                      |
|------------------------------------|------------------------------------------------------|
| 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     |                                                      |

**13.6.1.58 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_CXPOCXPSTATUS      |                                                         |

**13.6.1.59 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     |                                                             |

**13.6.1.60 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    |                                                                                                                                                                                                                                                   |

**13.6.1.61 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     |                                                             |

**13.6.1.62 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       |                                                                                                                     |

**13.6.1.63 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             |                                                                                 |

**13.6.1.64 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            |                                                |

**13.6.1.65 DeviceCharacterSetEnums**

```
enum DeviceCharacterSetEnums
```

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

**Enumerator**

|                          |  |
|--------------------------|--|
| DeviceCharacterSet_UTF8  |  |
| DeviceCharacterSet_ASCII |  |
| NUM_DEVICECHARACTERSET   |  |

**13.6.1.66 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 <a href="#">Camera</a> Link clock. |
| NUM_DEVICECLOCKSELECTOR                |                                                     |

**13.6.1.67 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      |                           |

**13.6.1.68 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           |                |

**13.6.1.69 DeviceEndianessMechanismEnum**

```
enum DeviceEndianessMechanismEnum
```

< Identifies the endianness handling mode.

**Enumerator**

|                                   |                                                                                          |
|-----------------------------------|------------------------------------------------------------------------------------------|
| DeviceEndianessMechanism_Legacy   | Handling the device endianness according to <a href="#">GenICam Schema 1.0</a>           |
| DeviceEndianessMechanism_Standard | Handling the device endianness according to <a href="#">GenICam Schema 1.1 and later</a> |
| NUMDEVICEENDIANESSMECHANISM       |                                                                                          |

**13.6.1.70 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         |  |

**13.6.1.71 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 |                              |

**13.6.1.72 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 |                                                 |

**13.6.1.73 DevicePowerSupplySelectorEnums**

```
enum DevicePowerSupplySelectorEnums
```

< Selects the power supply source to control or read.

**Enumerator**

|                                    |  |
|------------------------------------|--|
| DevicePowerSupplySelector_External |  |
| NUM_DEVICEPOWERSUPPLYSELECTOR      |  |

**13.6.1.74 DeviceRegistersEndiannessEnums**

```
enum DeviceRegistersEndiannessEnums
```

< Endianess of the registers of the device.

**Enumerator**

|                                  |  |
|----------------------------------|--|
| DeviceRegistersEndianness_Little |  |
| DeviceRegistersEndianness_Big    |  |
| NUM_DEVICEREGISTERSENDIANNESST   |  |

**13.6.1.75 DeviceScanTypeEnums**

```
enum DeviceScanTypeEnums
```

< Scan type of the sensor of the device.

**Enumerator**

|                         |  |
|-------------------------|--|
| DeviceScanType_Areascan |  |
| NUM_DEVICESCANTYPE      |  |

### 13.6.1.76 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_DEVICESERIALPORTBAUDRATE        |                                   |

### 13.6.1.77 DeviceSerialPortSelectorEnums

enum [DeviceSerialPortSelectorEnums](#)

< Selects which serial port of the device to control.

Enumerator

|                                     |                                                                       |
|-------------------------------------|-----------------------------------------------------------------------|
| DeviceSerialPortSelector_CameraLink | Serial port associated to the <a href="#">Camera</a> link connection. |
| NUM_DEVICESERIALPORTSELECTOR        |                                                                       |

### 13.6.1.78 DeviceStreamChannelEndiannessEnums

enum [DeviceStreamChannelEndiannessEnums](#)

< Endianess 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_DEVICESTREAMCHANNELENDIANNES     |                                       |

### 13.6.1.79 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         |                                  |

### 13.6.1.80 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                  |
| 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 |

## Enumerator

|                                     |                   |
|-------------------------------------|-------------------|
| 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_DEVICETAPGEOMETRY               |                   |

## 13.6.1.81 DeviceTemperatureSelectorEnums

```
enum DeviceTemperatureSelectorEnums
```

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

## Enumerator

|                                  |  |
|----------------------------------|--|
| DeviceTemperatureSelector_Sensor |  |
| NUM_DEVICETEMPERATURESELECTOR    |  |

## 13.6.1.82 DeviceTLTypeEnums

```
enum DeviceTLTypeEnums
```

< Transport Layer type of the device.

**Enumerator**

|                           |  |
|---------------------------|--|
| DeviceTLType_GigEVision   |  |
| DeviceTLType_CameraLink   |  |
| DeviceTLType_CameraLinkHS |  |
| DeviceTLType_CoaXPress    |  |
| DeviceTLType_USB3Vision   |  |
| DeviceTLType_Custom       |  |
| NUM_DEVICETYPE            |  |

**13.6.1.83 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           |                        |

**13.6.1.84 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         |                                                     |

### 13.6.1.85 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            |                                                                                                                         |

### 13.6.1.86 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           |                                                                                                                                                                                                      |

### 13.6.1.87 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         |                                                                          |

**13.6.1.88 EncoderResetSourceEnums**enum [EncoderResetSourceEnums](#)

&lt; 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.         |
| 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.       |

## Enumerator

|                                    |                                                                                               |
|------------------------------------|-----------------------------------------------------------------------------------------------|
| 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             |                                                                                               |

**13.6.1.89 EncoderSelectorEnums**enum [EncoderSelectorEnums](#)

&lt; Selects which Encoder to configure.

## Enumerator

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

**13.6.1.90 EncoderSourceAEnums**enum [EncoderSourceAEnums](#)

&lt; 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   |                                                          |

### 13.6.1.91 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   |                                                           |

### 13.6.1.92 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           |                                           |

### 13.6.1.93 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 |  |
|-----------------------|--|

## Enumerator

|                                   |
|-----------------------------------|
| 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                |
| GENICAM_ERR_INVALID_ARGUMENT      |
| GENICAM_ERR_OUT_OF_RANGE          |
| GENICAM_ERR_PROPERTY              |
| GENICAM_ERR_RUN_TIME              |
| GENICAM_ERR_LOGICAL               |
| GENICAM_ERR_ACCESS                |
| GENICAM_ERR_TIMEOUT               |
| GENICAM_ERR_DYNAMIC_CAST          |
| GENICAM_ERR_GENERIC               |
| GENICAM_ERR_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           |

## 13.6.1.94 EventNotificationEnums

```
enum EventNotificationEnums
```

< Enables/Disables the selected event.

**Enumerator**

|                       |  |
|-----------------------|--|
| EventNotification_On  |  |
| EventNotification_Off |  |
| NUM_EVENTNOTIFICATION |  |

**13.6.1.95 EventSelectorEnums**

```
enum EventSelectorEnums
```

< Selects which Event to enable or disable.

**Enumerator**

|                                 |  |
|---------------------------------|--|
| EventSelector_Error             |  |
| EventSelector_ExposureEnd       |  |
| EventSelector_SerialPortReceive |  |
| NUM_EVENTSELECTOR               |  |

**13.6.1.96 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_LOGGING_EVENT             |  |
| SPINNAKER_EVENT_UNKNOWN                   |  |
| SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL |  |

**13.6.1.97 ExposureActiveModeEnums**

```
enum ExposureActiveModeEnums
```

< Control sensor active exposure mode.

#### Enumerator

|                              |  |
|------------------------------|--|
| ExposureActiveMode_Line1     |  |
| ExposureActiveMode_AnyPixels |  |
| ExposureActiveMode_AllPixels |  |
| NUM_EXPOSUREACTIVEMODE       |  |

### 13.6.1.98 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        |                                                                                                  |

### 13.6.1.99 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          |                                                                                                                                                    |

### 13.6.1.100 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        |                                                                                                                                                              |

**13.6.1.101 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         |                                        |

**13.6.1.102 FileModeEnums**

```
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       |  |

### 13.6.1.103 FileOperationSelectorEnums

enum `FileOperationSelectorEnums`

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

Enumerator

|                                           |  |
|-------------------------------------------|--|
| <code>FileOperationSelector_Open</code>   |  |
| <code>FileOperationSelector_Close</code>  |  |
| <code>FileOperationSelector_Read</code>   |  |
| <code>FileOperationSelector_Write</code>  |  |
| <code>FileOperationSelector_Delete</code> |  |
| <code>NUM_FILEOPERATIONSELECTOR</code>    |  |

### 13.6.1.104 FileOperationStatusEnums

enum `FileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

|                                           |                                                          |
|-------------------------------------------|----------------------------------------------------------|
| <code>FileOperationStatus_Success</code>  | File Operation was sucessful.                            |
| <code>FileOperationStatus_Failure</code>  | File Operation failed.                                   |
| <code>FileOperationStatus_Overflow</code> | An overflow occurred while executing the File Operation. |
| <code>NUM_FILEOPERATIONSTATUS</code>      |                                                          |

### 13.6.1.105 FileSelectorEnums

enum `FileSelectorEnums`

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

Enumerator

|                                          |  |
|------------------------------------------|--|
| <code>FileSelector_UserSetDefault</code> |  |
| <code>FileSelector_UserSet0</code>       |  |
| <code>FileSelector_UserSet1</code>       |  |
| <code>FileSelector_UserFile1</code>      |  |
| <code>FileSelector_SerialPort0</code>    |  |
| <code>NUM_FILESELECTOR</code>            |  |

**13.6.1.106 FilterDriverStatusEnum**

enum [FilterDriverStatusEnum](#)

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

Enumerator

|                                              |                                             |
|----------------------------------------------|---------------------------------------------|
| <code>FilterDriverStatus_NotSupported</code> | Not Supported                               |
| <code>FilterDriverStatus_Disabled</code>     | FLIR Light Weight Filter Driver is disabled |
| <code>FilterDriverStatus_Enabled</code>      | FLIR Light Weight Filter Driver is enabled  |
| <code>NUMFILTERDRIVERSTATUS</code>           |                                             |

**13.6.1.107 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

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

**13.6.1.108 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

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

### 13.6.1.109 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 |  |

### 13.6.1.110 GenICamXMLLocationEnum

enum [GenICamXMLLocationEnum](#)

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

Enumerator

|                           |                                              |
|---------------------------|----------------------------------------------|
| GenICamXMLLocation_Device | Load <a href="#">GenICam</a> XML from device |
| GenICamXMLLocation_Host   | Load <a href="#">GenICam</a> XML from host   |
| NUMGENICAMXMLLOCATION     |                                              |

### 13.6.1.111 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                               |                             |

### 13.6.1.112 GevCCPEnums

enum [GevCCPEnums](#)

< Controls the device access privilege of an application.

#### Enumerator

|                        |  |
|------------------------|--|
| GevCCP_OpenAccess      |  |
| GevCCP_ExclusiveAccess |  |
| GevCCP_ControlAccess   |  |
| NUM_GEVCCP             |  |

### 13.6.1.113 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        |             |

### 13.6.1.114 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_GEVGVCPEXTENDEDSTATUSCODESELECTOR         |             |

### 13.6.1.115 GevGVSPExtendedIDModeEnums

enum [GevGVSPExtendedIDModeEnums](#)

< Enables the extended IDs mode.

**Enumerator**

|                            |     |
|----------------------------|-----|
| GevGVSPExtendedIDMode_Off  | Off |
| GevGVSPExtendedIDMode_On   | On  |
| NUM_GEVGVSPEXTENDEDDIDMODE |     |

**13.6.1.116 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     |                  |

**13.6.1.117 GevIEEE1588ModeEnums**

```
enum GevIEEE1588ModeEnums
```

< Provides the mode of the IEEE 1588 clock.

**Enumerator**

|                           |            |
|---------------------------|------------|
| GevIEEE1588Mode_Auto      | Automatic  |
| GevIEEE1588Mode_SlaveOnly | Slave Only |
| NUM_GEVIEEE1588MODE       |            |

**13.6.1.118 GevIEEE1588StatusEnums**

```
enum GevIEEE1588StatusEnums
```

< Provides the status of the IEEE 1588 clock.

**Enumerator**

|                                |              |
|--------------------------------|--------------|
| GevIEEE1588Status_Initializing | Initializing |
| GevIEEE1588Status_Faulty       | Faulty       |
| GevIEEE1588Status_Disabled     | Disabled     |

Enumerator

|                                |              |
|--------------------------------|--------------|
| GevIEEE1588Status_Listening    | Listening    |
| GevIEEE1588Status_PreMaster    | Pre Master   |
| GevIEEE1588Status_Master       | Master       |
| GevIEEE1588Status_Passive      | Passive      |
| GevIEEE1588Status_Uncalibrated | Uncalibrated |
| GevIEEE1588Status_Slave        | Slave        |
| NUM_GEVIEEE1588STATUS          |              |

### 13.6.1.119 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          |               |

### 13.6.1.120 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        |             |

### 13.6.1.121 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                         |

### 13.6.1.122 GUIXMLLocationEnum

enum [GUIXMLLocationEnum](#)

< Sets the location to load GUI XML.

Enumerator

|                       |                      |
|-----------------------|----------------------|
| GUIXMLLocation_Device | Load XML from device |
| GUIXMLLocation_Host   | Load XML from host   |
| NUMGUIXMLLOCATION     |                      |

### 13.6.1.123 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. |
| 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         |                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### 13.6.1.124 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<br>Standard  | Indicates this is a baseline sequential (single-scan) DCT-based JPEG.                                                                                                     |
| ImageCompressionJPEGFormatOption_Baseline<br>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<br>ION               |                                                                                                                                                                           |

### 13.6.1.125 ImageCompressionModeEnums

enum [ImageCompressionModeEnums](#)

<

Enumerator

|                               |  |
|-------------------------------|--|
| ImageCompressionMode_Off      |  |
| ImageCompressionMode_Lossless |  |
| NUM_IMAGECOMPRESSIONMODE      |  |

### 13.6.1.126 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        |                                                                                                                                                                   |

### 13.6.1.127 ImageFileFormat

enum [ImageFileFormat](#)

File formats to be used for saving images to disk.

Enumerator

|               |                                            |
|---------------|--------------------------------------------|
| FROM_FILE_EXT | Determine file format from file extension. |
| PGM           | Portable gray map.                         |
| PPM           | Portable pixmap.                           |
| BMP           | Bitmap.                                    |
| JPEG          | JPEG.                                      |
| JPEG2000      | JPEG 2000.                                 |

## Enumerator

|  |                                |                              |
|--|--------------------------------|------------------------------|
|  | TIFF                           | Tagged image file format.    |
|  | PNG                            | Portable network graphics.   |
|  | RAW                            | Raw data.                    |
|  | JPEG12_C                       | 12 bit compressed JPEG data. |
|  | IMAGE_FILE_FORMAT_FORCE_32BITS |                              |

**13.6.1.128 ImageStatus**

```
enum ImageStatus
```

Status of images returned from GetNextImage() call.

## Enumerator

|                                        |                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IMAGE_UNKNOWN_ERROR                    | <a href="#">Image</a> has an unknown error.                                                                                                                                                                                                                                                                                                                                                      |
| IMAGE_NO_ERROR                         | <a href="#">Image</a> is returned from GetNextImage() call without any errors.                                                                                                                                                                                                                                                                                                                   |
| IMAGE_CRC_CHECK_FAILED                 | <a href="#">Image</a> failed CRC check.                                                                                                                                                                                                                                                                                                                                                          |
| IMAGE_DATA_OVERFLOW                    | Received more data than the size of the image.                                                                                                                                                                                                                                                                                                                                                   |
| IMAGE_MISSING_PACKETS                  | <a href="#">Image</a> has missing packets. Potential fixes include enabling jumbo packets and adjusting packet size/delay. For more information see<br><a href="https://www.flir.com/support-center/iis/machine-vision/application-note/missing-packets-in-image-data-streams">https://www.flir.com/support-center/iis/machine-vision/application-note/missing-packets-in-image-data-streams</a> |
| IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT  | <a href="#">Image</a> leader is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                                |
| IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT | <a href="#">Image</a> trailer is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                               |
| IMAGE_PACKETID_INCONSISTENT            | <a href="#">Image</a> has an inconsistent packet id. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                       |
| IMAGE_MISSING_LEADER                   | <a href="#">Image</a> leader is missing. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                                   |
| IMAGE_MISSING_TRAILER                  | <a href="#">Image</a> trailer is missing. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                                  |
| IMAGE_DATA_INCOMPLETE                  | <a href="#">Image</a> data is incomplete. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                                  |
| IMAGE_INFO_INCONSISTENT                | <a href="#">Image</a> info is corrupted. Could be caused by missing packet(s). See link above.                                                                                                                                                                                                                                                                                                   |
| IMAGE_CHUNK_DATA_INVALID               | <a href="#">Image</a> chunk data is invalid.                                                                                                                                                                                                                                                                                                                                                     |
| IMAGE_NO_SYSTEM_RESOURCES              | <a href="#">Image</a> cannot be processed due to lack of system resources.                                                                                                                                                                                                                                                                                                                       |

### 13.6.1.129 InferenceBoxType

enum [InferenceBoxType](#)

Inference Bounding Box Type.

Enumerator

|                                                       |  |
|-------------------------------------------------------|--|
| <a href="#">INFEERENCE_BOX_TYPE_RECTANGLE</a>         |  |
| <a href="#">INFEERENCE_BOX_TYPE_CIRCLE</a>            |  |
| <a href="#">INFEERENCE_BOX_TYPE_ROTATED_RECTANGLE</a> |  |

### 13.6.1.130 InterfaceTypeEnum

enum [InterfaceTypeEnum](#)

< Transport layer type of the interface.

Enumerator

|                                            |                                        |
|--------------------------------------------|----------------------------------------|
| <a href="#">InterfaceType_GigEVision</a>   | GigE Vision                            |
| <a href="#">InterfaceType_CameraLink</a>   | <a href="#">Camera Link</a>            |
| <a href="#">InterfaceType_CameraLinkHS</a> | <a href="#">Camera Link High Speed</a> |
| <a href="#">InterfaceType_CoaXPress</a>    | CoaXPress                              |
| <a href="#">InterfaceType_USB3Vision</a>   | USB3 Vision                            |
| <a href="#">InterfaceType_Custom</a>       | Custom transport layer                 |
| <a href="#">NUMINTERFACETYPE</a>           |                                        |

### 13.6.1.131 LineFormatEnums

enum [LineFormatEnums](#)

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

Enumerator

|                                        |  |
|----------------------------------------|--|
| <a href="#">LineFormat_NoConnect</a>   |  |
| <a href="#">LineFormat_TriState</a>    |  |
| <a href="#">LineFormat_TTL</a>         |  |
| <a href="#">LineFormat_LVDS</a>        |  |
| <a href="#">LineFormat_RS422</a>       |  |
| <a href="#">LineFormat_OptoCoupled</a> |  |
| <a href="#">LineFormat_OpenDrain</a>   |  |
| <a href="#">NUM_LINEFORMAT</a>         |  |

### 13.6.1.132 LineInputFilterSelectorEnums

enum [LineInputFilterSelectorEnums](#)

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

Enumerator

|                                  |  |
|----------------------------------|--|
| LineInputFilterSelector_Deglitch |  |
| LineInputFilterSelector_Debounce |  |
| NUM_LINEINPUTFILTERSELECTOR      |  |

### 13.6.1.133 LineModeEnums

enum [LineModeEnums](#)

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

Enumerator

|                 |  |
|-----------------|--|
| LineMode_Input  |  |
| LineMode_Output |  |
| NUM_LINEMODE    |  |

### 13.6.1.134 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   |  |

### 13.6.1.135 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              |

### 13.6.1.136 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         |

**13.6.1.137 LogicBlockLUTInputSelectorEnums**

```
enum LogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

**Enumerator**

|                                                |  |
|------------------------------------------------|--|
| <code>LogicBlockLUTInputSelector_Input0</code> |  |
| <code>LogicBlockLUTInputSelector_Input1</code> |  |
| <code>LogicBlockLUTInputSelector_Input2</code> |  |
| <code>LogicBlockLUTInputSelector_Input3</code> |  |
| <code>NUM_LOGICBLOCKLUTINPUTSELECTOR</code>    |  |

**13.6.1.138 LogicBlockLUTInputSourceEnums**

```
enum LogicBlockLUTInputSourceEnums
```

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

**Enumerator**

|                                                         |                                |
|---------------------------------------------------------|--------------------------------|
| <code>LogicBlockLUTInputSource_Zero</code>              | <code>Zero</code>              |
| <code>LogicBlockLUTInputSource_Line0</code>             | <code>Line0</code>             |
| <code>LogicBlockLUTInputSource_Line1</code>             | <code>Line1</code>             |
| <code>LogicBlockLUTInputSource_Line2</code>             | <code>Line2</code>             |
| <code>LogicBlockLUTInputSource_Line3</code>             | <code>Line3</code>             |
| <code>LogicBlockLUTInputSource_UserOutput0</code>       | <code>UserOutput0</code>       |
| <code>LogicBlockLUTInputSource_UserOutput1</code>       | <code>UserOutput1</code>       |
| <code>LogicBlockLUTInputSource_UserOutput2</code>       | <code>UserOutput2</code>       |
| <code>LogicBlockLUTInputSource_UserOutput3</code>       | <code>UserOutput3</code>       |
| <code>LogicBlockLUTInputSource_Counter0Start</code>     | <code>Counter0Start</code>     |
| <code>LogicBlockLUTInputSource_Counter1Start</code>     | <code>Counter1Start</code>     |
| <code>LogicBlockLUTInputSource_Counter0End</code>       | <code>Counter0End</code>       |
| <code>LogicBlockLUTInputSource_Counter1End</code>       | <code>Counter1End</code>       |
| <code>LogicBlockLUTInputSource_LogicBlock0</code>       | <code>LogicBlock0</code>       |
| <code>LogicBlockLUTInputSource_LogicBlock1</code>       | <code>LogicBlock1</code>       |
| <code>LogicBlockLUTInputSource_ExposureStart</code>     | <code>ExposureStart</code>     |
| <code>LogicBlockLUTInputSource_ExposureEnd</code>       | <code>ExposureEnd</code>       |
| <code>LogicBlockLUTInputSource_FrameTriggerWait</code>  | <code>FrameTriggerWait</code>  |
| <code>LogicBlockLUTInputSource_AcquisitionActive</code> | <code>AcquisitionActive</code> |
| <code>NUM_LOGICBLOCKLUTINPUTSOURCE</code>               |                                |

### 13.6.1.139 LogicBlockLUTSelectorEnums

enum [LogicBlockLUTSelectorEnums](#)

< Selects which LogicBlock LUT to configure

Enumerator

|                              |  |
|------------------------------|--|
| LogicBlockLUTSelector_Value  |  |
| LogicBlockLUTSelector_Enable |  |
| NUM_LOGICBLOCKLUTSELECTOR    |  |

### 13.6.1.140 LogicBlockSelectorEnums

enum [LogicBlockSelectorEnums](#)

< Selects which LogicBlock to configure

Enumerator

|                                |  |
|--------------------------------|--|
| LogicBlockSelector_LogicBlock0 |  |
| LogicBlockSelector_LogicBlock1 |  |
| NUM_LOGICBLOCKSELECTOR         |  |

### 13.6.1.141 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  |                                                                                      |

### 13.6.1.142 PayloadTypeInfoIDs

enum [PayloadTypeInfoIDs](#)

## Enumerator

|                              |  |
|------------------------------|--|
| PAYLOAD_TYPE_UNKNOWN         |  |
| PAYLOAD_TYPE_IMAGE           |  |
| PAYLOAD_TYPE_RAW_DATA        |  |
| PAYLOAD_TYPE_FILE            |  |
| PAYLOAD_TYPE_CHUNK_DATA      |  |
| PAYLOAD_TYPE_JPEG            |  |
| PAYLOAD_TYPE_JPEG2000        |  |
| PAYLOAD_TYPE_H264            |  |
| PAYLOAD_TYPE_CHUNK_ONLY      |  |
| PAYLOAD_TYPE_DEVICE_SPECIFIC |  |
| PAYLOAD_TYPE_MULTI_PART      |  |
| PAYLOAD_TYPE_CUSTOM_ID       |  |
| PAYLOAD_TYPE_EXTENDED_CHUNK  |  |

**13.6.1.143 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     |                          |

**13.6.1.144 PixelFormatEnums**

```
enum PixelFormatEnums
```

< Format of the pixel provided by the camera.

## Enumerator

|                        |  |
|------------------------|--|
| PixelFormat_Mono8      |  |
| PixelFormat_Mono16     |  |
| PixelFormat_RGB8Packed |  |
| PixelFormat_BayerGR8   |  |
| PixelFormat_BayerRG8   |  |

## Enumerator

|                             |                                  |
|-----------------------------|----------------------------------|
| 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          | 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 |

## Enumerator

|                          |                                          |
|--------------------------|------------------------------------------|
| 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          |
| 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                             |

## Enumerator

|                                   |                                                  |
|-----------------------------------|--------------------------------------------------|
| PixelFormat_G12                   | Green 12-bit                                     |
| PixelFormat_G16                   | Green 16-bit                                     |
| PixelFormat_B8                    | Blue 8-bit                                       |
| PixelFormat_B10                   | Blue 10-bit                                      |
| 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            |

## Enumerator

|                                 |                                                               |
|---------------------------------|---------------------------------------------------------------|
| 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                 |
| 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                                     |

## Enumerator

|                                     |                                                |
|-------------------------------------|------------------------------------------------|
| 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                      |
| 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 |

## Enumerator

|                                 |                                                   |
|---------------------------------|---------------------------------------------------|
| 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 |
| 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.                          |
| UNKNOWN_PIXELFORMAT             |                                                   |
| NUM_PIXELFORMAT                 |                                                   |

**13.6.1.145 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   |

## Enumerator

|                                      |                                          |
|--------------------------------------|------------------------------------------|
| PixelFormatInfoSelector_BayerGB12    | Bayer Green-Blue 12-bit unpacked         |
| PixelFormatInfoSelector_BayerGB12p   | Bayer Green-Blue 12-bit packed           |
| 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        |

## Enumerator

|                                               |                                                  |
|-----------------------------------------------|--------------------------------------------------|
| PixelFormatInfoSelector_BGR8                  | Blue-Green-Red 8-bit                             |
| PixelFormatInfoSelector_BGR10                 | Blue-Green-Red 10-bit unpacked                   |
| 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                    |

## Enumerator

|                                        |                                                               |
|----------------------------------------|---------------------------------------------------------------|
| PixelFormatInfoSelector_Coord3D_B16    | 3D coordinate B 16-bit                                        |
| PixelFormatInfoSelector_Coord3D_B32f   | 3D coordinate B 32-bit floating point                         |
| 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            |

## Enumerator

|                                                      |                                                              |
|------------------------------------------------------|--------------------------------------------------------------|
| PixelFormatInfoSelector_SCF1WGWR10                   | Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked |
| 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_CbYY←<br>CrYY | YCbCr 4:1:1 8-bit BT.601                                     |
| PixelFormatInfoSelector_YCbCr601_422_8               | YCbCr 4:2:2 8-bit BT.601                                     |

## Enumerator

|                                                  |                                                   |
|--------------------------------------------------|---------------------------------------------------|
| 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                |
| 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                                  |

## Enumerator

|                             |
|-----------------------------|
| NUM_PIXELFORMATINFOSELECTOR |
|-----------------------------|

**13.6.1.146 PixelFormatIntType**

```
enum PixelFormatIntType
```

Possible integer types and packing used in a pixel format.

## Enumerator

|                 |
|-----------------|
| IntType_UINT8   |
| IntType_INT8    |
| IntType_UINT10  |
| IntType_UINT10p |
| IntType_UINT10P |
| IntType_UINT12  |
| IntType_UINT12p |
| IntType_UINT12P |
| IntType_UINT14  |
| IntType_UINT16  |
| IntType_INT16   |
| IntType_FLOAT32 |
| IntType_UNKNOWN |

**13.6.1.147 PixelFormatNamespaceID**

```
enum PixelFormatNamespaceID
```

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_PIXELFORMAT_NAMESPACE_UNKNOWN |
| SPINNAKER_PIXELFORMAT_NAMESPACE_GEV     |

## Enumerator

|                                            |  |
|--------------------------------------------|--|
| SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC       |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID  |  |

**13.6.1.148 PixelSizeEnums**enum [PixelSizeEnums](#)

&lt; 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   |                    |

**13.6.1.149 POEStatusEnum**enum [POEStatusEnum](#)

&lt; 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           |               |

### 13.6.1.150 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     |                                                     |

### 13.6.1.151 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 |                                  |

### 13.6.1.152 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     |                                                                  |

### 13.6.1.153 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 fluorescent light with color temperature 3000K.                               |
| RgbTransformLightSource_CoolFluorescent4000K | Uses a matrix optimized for a typical cool fluorescent light with color temperature 4000K.                               |
| 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                  |                                                                                                                          |

### 13.6.1.154 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          |                         |

**13.6.1.155 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         |                                   |

**13.6.1.156 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         |                                                     |

**13.6.1.157 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         |                                                                                                                                               |

### 13.6.1.158 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          |                           |

### 13.6.1.159 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        |                                                    |

### 13.6.1.160 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. |

## Enumerator

|                                        |                                                                                                                                                                                                                                                                                                             |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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. |
| 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                   |                                                                                                                                                                                                                                                                                                             |

**13.6.1.161 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   |          |

### 13.6.1.162 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         |                                                                                                                                                                    |

### 13.6.1.163 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   |          |

### 13.6.1.164 SequencerConfigurationModeEnums

enum [SequencerConfigurationModeEnums](#)

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

Enumerator

|                                |  |
|--------------------------------|--|
| SequencerConfigurationMode_Off |  |
| SequencerConfigurationMode_On  |  |
| NUM_SEQUENCERCONFIGURATIONMODE |  |

### 13.6.1.165 SequencerConfigurationValidEnums

```
enum SequencerConfigurationValidEnums
```

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

Enumerator

|                                 |  |
|---------------------------------|--|
| SequencerConfigurationValid_No  |  |
| SequencerConfigurationValid_Yes |  |
| NUM_SEQUENCERCONFIGURATIONVALID |  |

### 13.6.1.166 SequencerModeEnums

```
enum SequencerModeEnums
```

< Controls whether or not a sequencer is active.

Enumerator

|                   |  |
|-------------------|--|
| SequencerMode_Off |  |
| SequencerMode_On  |  |
| NUM_SEQUENCERMODE |  |

### 13.6.1.167 SequencerSetValidEnums

```
enum SequencerSetValidEnums
```

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

Enumerator

|                       |  |
|-----------------------|--|
| SequencerSetValid_No  |  |
| SequencerSetValid_Yes |  |
| NUM_SEQUENCERSETVALID |  |

### 13.6.1.168 SequencerTriggerActivationEnums

```
enum SequencerTriggerActivationEnums
```

< Specifies the activation mode of the sequencer trigger.

#### Enumerator

|                                        |  |
|----------------------------------------|--|
| SequencerTriggerActivation_RisingEdge  |  |
| SequencerTriggerActivation_FallingEdge |  |
| SequencerTriggerActivation_AnyEdge     |  |
| SequencerTriggerActivation_LevelHigh   |  |
| SequencerTriggerActivation_LevelLow    |  |
| NUM_SEQUENCERTRIGGERACTIVATION         |  |

### 13.6.1.169 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        |  |

### 13.6.1.170 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        |  |

### 13.6.1.171 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   |  |

### 13.6.1.172 SerialPortSelectorEnums

enum [SerialPortSelectorEnums](#)

< Selects which serial port of the device to control.

Enumerator

|                                |  |
|--------------------------------|--|
| SerialPortSelector_SerialPort0 |  |
| NUM_SERIALPORTSELECTOR         |  |

### 13.6.1.173 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   |  |

### 13.6.1.174 SerialPortStopBitsEnums

enum `SerialPortStopBitsEnums`

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

Enumerator

|                                               |  |
|-----------------------------------------------|--|
| <code>SerialPortStopBits_Bits1</code>         |  |
| <code>SerialPortStopBits_Bits1AndAHalf</code> |  |
| <code>SerialPortStopBits_Bits2</code>         |  |
| <code>NUM_SERIALPORTSTOPBITS</code>           |  |

### 13.6.1.175 SoftwareSignalSelectorEnums

enum `SoftwareSignalSelectorEnums`

< Selects which Software Signal features to control.

Enumerator

|                                                     |                                                   |
|-----------------------------------------------------|---------------------------------------------------|
| <code>SoftwareSignalSelector_SoftwareSignal0</code> | Selects the software generated signal to control. |
| <code>SoftwareSignalSelector_SoftwareSignal1</code> | Selects the software generated signal to control. |
| <code>SoftwareSignalSelector_SoftwareSignal2</code> | Selects the software generated signal to control. |
| <code>NUM_SOFTWARESIGNALSELECTOR</code>             |                                                   |

### 13.6.1.176 SourceSelectorEnums

enum `SourceSelectorEnums`

< Selects the source to control.

Enumerator

|                                     |                               |
|-------------------------------------|-------------------------------|
| <code>SourceSelector_Source0</code> | Selects the data source 0.    |
| <code>SourceSelector_Source1</code> | Selects the data source 1.    |
| <code>SourceSelector_Source2</code> | Selects the data source 2.    |
| <code>SourceSelector_All</code>     | Selects all the data sources. |
| <code>NUM_SOURCESELECTOR</code>     |                               |

### 13.6.1.177 SpinnakerLogLevel

enum [SpinnakerLogLevel](#)

log levels

Enumerator

|                  |  |
|------------------|--|
| LOG_LEVEL_OFF    |  |
| LOG_LEVEL_FATAL  |  |
| LOG_LEVEL_ALERT  |  |
| LOG_LEVEL_CRIT   |  |
| LOG_LEVEL_ERROR  |  |
| LOG_LEVEL_WARN   |  |
| LOG_LEVEL_NOTICE |  |
| LOG_LEVEL_INFO   |  |
| LOG_LEVEL_DEBUG  |  |
| LOG_LEVEL_NOTSET |  |

### 13.6.1.178 StatisticsChannel

enum [StatisticsChannel](#)

Channels that allow statistics to be calculated.

Enumerator

|                         |  |
|-------------------------|--|
| GREY                    |  |
| RED                     |  |
| GREEN                   |  |
| BLUE                    |  |
| HUE                     |  |
| SATURATION              |  |
| LIGHTNESS               |  |
| NUM_STATISTICS_CHANNELS |  |

### 13.6.1.179 StreamBufferCountModeEnum

enum [StreamBufferCountModeEnum](#)

< Controls access to setting the number of buffers used for the stream. Locked to Manual mode on 32-bit Windows due to memory constraints.

Enumerator

|                              |                                                                |
|------------------------------|----------------------------------------------------------------|
| StreamBufferCountMode_Manual | The number of buffers used for the stream are set by the user. |
|------------------------------|----------------------------------------------------------------|

## Enumerator

|                            |                                                                                                                   |
|----------------------------|-------------------------------------------------------------------------------------------------------------------|
| StreamBufferCountMode_Auto | DEPRECATED. The number of buffers used for the stream is automatically calculated based on the device frame rate. |
| NUMSTREAMBUFFERCOUNTMODE   |                                                                                                                   |

**13.6.1.180 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                   |                                                                                                                                                                                                                                                                                                                                                                |

**13.6.1.181 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           |                        |

**13.6.1.182 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               |                                                                                               |

**13.6.1.183 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           |                                                                                            |

**13.6.1.184 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    |                      |

**13.6.1.185 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              |                                                   |

**13.6.1.186 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         |                                                                               |

**13.6.1.187 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          |                                                        |

**13.6.1.188 TLTypeEnum**enum [TLTypeEnum](#)

&lt; Transport layer type of the GenTL Producer implementation.

## Enumerator

|                     |                                                                                          |
|---------------------|------------------------------------------------------------------------------------------|
| TLType_GigEVision   | GigE Vision                                                                              |
| TLType_CameraLink   | <a href="#">Camera</a> Link                                                              |
| TLType_CameraLinkHS | <a href="#">Camera</a> Link High Speed                                                   |
| TLType_CoaxPress    | CoaXPress                                                                                |
| TLType_USB3Vision   | USB3 Vision                                                                              |
| TLType_Mixed        | Different <a href="#">Interface</a> modules of the GenTL Producer are of different types |
| TLType_Custom       | Custom transport layer                                                                   |
| NUMTLTYPE           |                                                                                          |

**13.6.1.189 TransferComponentSelectorEnums**enum [TransferComponentSelectorEnums](#)

&lt; 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.                                                |
| 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. |
| Generated by Doxygen<br>NUM_TRANSFERCOMPONENTSELECTOR |                                                                                                                                                                                        |

### 13.6.1.190 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

|                                                 |                 |
|-------------------------------------------------|-----------------|
| <code>TransferControlMode_Basic</code>          | Basic           |
| <code>TransferControlMode_Automatic</code>      | Automatic       |
| <code>TransferControlMode_UserControlled</code> | User Controlled |
| <code>NUM_TRANSFERCONTROLMODE</code>            |                 |

### 13.6.1.191 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

|                                               |             |
|-----------------------------------------------|-------------|
| <code>TransferOperationMode_Continuous</code> | Continuous  |
| <code>TransferOperationMode_MultiBlock</code> | Multi Block |
| <code>NUM_TRANSFEROPERATIONMODE</code>        |             |

### 13.6.1.192 TransferQueueModeEnums

enum `TransferQueueModeEnums`

< Specifies the operation mode of the transfer queue.

Enumerator

|                                                |                                            |
|------------------------------------------------|--------------------------------------------|
| <code>TransferQueueMode_FirstInFirstOut</code> | Blocks first In are transferred Out first. |
| <code>NUM_TRANSFERQUEUEMODE</code>             |                                            |

**13.6.1.193 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 simultaneously. |
| NUM_TRANSFERSELECTOR     |                                                                    |

**13.6.1.194 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           |                                                                                                                          |

**13.6.1.195 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. |

**Enumerator**

|                                     |                                                                                                                                                               |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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       |                                                                                                                                                               |

**13.6.1.196 TransferTriggerModeEnums**

```
enum TransferTriggerModeEnums
```

< Controls if the selected trigger is active.

**Enumerator**

|                          |                                |
|--------------------------|--------------------------------|
| TransferTriggerMode_Off  | Disables the selected trigger. |
| TransferTriggerMode_On   | Enable the selected trigger.   |
| NUM_TRANSFERTRIGGERTMODE |                                |

**13.6.1.197 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                |                                                                                                                             |

**13.6.1.198 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             |                                                                                                                                            |

### 13.6.1.199 TriggerActivationEnums

enum `TriggerActivationEnums`

< Specifies the activation mode of the trigger.

Enumerator

|                                            |  |
|--------------------------------------------|--|
| <code>TriggerActivation_LevelLow</code>    |  |
| <code>TriggerActivation_LevelHigh</code>   |  |
| <code>TriggerActivation_FallingEdge</code> |  |
| <code>TriggerActivation_RisingEdge</code>  |  |
| <code>TriggerActivation_AnyEdge</code>     |  |
| <code>NUM_TRIGGERACTIVATION</code>         |  |

### 13.6.1.200 TriggerModeEnums

enum `TriggerModeEnums`

< Controls whether or not trigger is active.

Enumerator

|                              |  |
|------------------------------|--|
| <code>TriggerMode_Off</code> |  |
| <code>TriggerMode_On</code>  |  |
| <code>NUM_TRIGGERMODE</code> |  |

### 13.6.1.201 TriggerOverlapEnums

enum `TriggerOverlapEnums`

< Specifies the overlap mode of the trigger.

Enumerator

|                                           |  |
|-------------------------------------------|--|
| <code>TriggerOverlap_Off</code>           |  |
| <code>TriggerOverlap_ReadOut</code>       |  |
| <code>TriggerOverlap_PreviousFrame</code> |  |
| <code>NUM_TRIGGEROVERLAP</code>           |  |

### 13.6.1.202 TriggerSelectorEnums

```
enum TriggerSelectorEnums
```

< Selects the type of trigger to configure.

Enumerator

|                                  |  |
|----------------------------------|--|
| TriggerSelector_AcquisitionStart |  |
| TriggerSelector_FrameStart       |  |
| TriggerSelector_FrameBurstStart  |  |
| NUM_TRIGGERSELECTOR              |  |

### 13.6.1.203 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           |  |

### 13.6.1.204 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         |  |

**13.6.1.205 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      |                          |

**13.6.1.206 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      |                          |

**13.6.1.207 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   |                                                     |

**13.6.2 Function Documentation****13.6.2.1 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>plImage</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\( ImagePtr plImage\)](#)

**13.6.2.2 operator==( )**

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

Pointer equal.

### 13.6.3 Variable Documentation

#### 13.6.3.1 EVENT\_TIMEOUT\_INFINITE

```
const uint64_t EVENT_TIMEOUT_INFINITE = 0xFFFFFFFFFFFFFF
```

#### 13.6.3.2 EVENT\_TIMEOUT\_NONE

```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

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

## 13.7 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](#)

- class [CEventAdapterU3V](#)  
*Connects a GigE Event to a node map.*
- class [CEventPort](#)  
*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\\_automvector\\_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](#)

- 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 IDDeviceInfo 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 `_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,  
  I2C,  
  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` = 0x8000000L }  
*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` {  
  intf1Value,

```

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.

```

## 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_auvector_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 `IInteger & 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 IsCachable () 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 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`

- **virtual bool IsStreamable () const =0**  
*True if the node is streamable.*
- **virtual void GetPropertyNames (GenICam::gcstring\_vector &PropertyName) 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 `GetSwapEndianess` ()=0  
*Determines if the port adapter must perform an endianess 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)  
*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](#) (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 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 IPotConstruct](#)  
*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.*

### 13.7.1 Typedef Documentation

#### 13.7.1.1 CallbackHandleType

```
typedef intptr_t CallbackHandleType
```

the callback handle for nodes

#### 13.7.1.2 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

### 13.7.1.3 CCategoryRef

```
typedef CategoryNode CCategoryRef
```

### 13.7.1.4 CCommandRef

```
typedef CommandNode CCommandRef
```

### 13.7.1.5 CEnumEntryRef

```
typedef EnumEntryNode CEnumEntryRef
```

### 13.7.1.6 CEnumerationRef

```
typedef EnumNode CEnumerationRef
```

### 13.7.1.7 CFloatRef

```
typedef FloatNode CFloatRef
```

### 13.7.1.8 CIintegerRef

```
typedef IntegerNode CIintegerRef
```

### 13.7.1.9 CNodeMapRef

```
typedef NodeMap CNodeMapRef
```

### 13.7.1.10 CNodeRef

```
typedef Node CNodeRef
```

### 13.7.1.11 CPortRecorderRef

```
typedef PortRecorder CPortRecorderRef
```

Reference to an IPortRecorder pointer.

### 13.7.1.12 CPortRef

```
typedef PortNode CPortRef
```

### 13.7.1.13 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

### 13.7.1.14 CSelectorRef

```
typedef Node CSelectorRef
```

### 13.7.1.15 CStringRef

```
typedef StringNode CStringRef
```

### 13.7.1.16 CValueRef

```
typedef ValueNode CValueRef
```

### 13.7.1.17 IDevFileStream

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

### 13.7.1.18 NodeList\_t

```
typedef node_vector NodeList_t
```

a list of node references

### 13.7.1.19 ODevFileStream

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

### 13.7.1.20 StringList\_t

```
typedef GenICam::gcstring_vector StringList_t
```

A list of strings.

## 13.7.2 Enumeration Type Documentation

### 13.7.2.1 \_EAccessMode

```
enum _EAccessMode
```

access mode of a node

Enumerator

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

### 13.7.2.2 \_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 |

**13.7.2.3 \_ECallbackType**

enum [\\_ECallbackType](#)

the type of callback

**Enumerator**

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

**13.7.2.4 \_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  |

**13.7.2.5 \_EEndianess**

enum [\\_EEndianess](#)

Endianess of a value in a register.

Enumerator

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

### 13.7.2.6 \_EGenApiSchemaVersion

enum [\\_EGenApiSchemaVersion](#)

GenApi schema version.

Enumerator

|            |  |
|------------|--|
| v1_0       |  |
| v1_1       |  |
| _Undefined |  |

### 13.7.2.7 \_EIncMode

enum [\\_EIncMode](#)

typedef for increment mode

Enumerator

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

### 13.7.2.8 \_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 |

### 13.7.2.9 \_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. IPot interface |

### 13.7.2.10 \_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 |

### 13.7.2.11 \_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 |

### 13.7.2.12 \_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.                       |

### 13.7.2.13 \_ESign

enum [\\_ESign](#)

signed or unsigned integers

Enumerator

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

### 13.7.2.14 \_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 |

**13.7.2.15 \_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 |

**13.7.2.16 \_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.                  |

**13.7.2.17 \_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                                                       |

### 13.7.2.18 \_EYesNo

enum [\\_EYesNo](#)

Defines the choices of a Yes/No alternative.

#### Enumerator

|                 |     |
|-----------------|-----|
| Yes             |     |
| No              | yes |
| _UndefinedYesNo | no  |

### 13.7.2.19 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.                                                                          |

### 13.7.2.20 EContentType\_t

enum `EContentType_t`

Lists the processable file types.

Enumerator

|                                    |                                          |
|------------------------------------|------------------------------------------|
| <code>ContentType_Xml</code>       | XML camera description file text.        |
| <code>ContentType_ZippedXml</code> | Zipped XML camera description file text. |

### 13.7.2.21 GVCP\_MESSAGE\_TAGS

enum `GVCP_MESSAGE_TAGS`

Enumerator

|                                |  |
|--------------------------------|--|
| <code>TAG_EVENT_CMD</code>     |  |
| <code>TAG_EVENTDATA_CMD</code> |  |

## 13.7.3 Function Documentation

### 13.7.3.1 `_ClearXMLCache()`

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

### 13.7.3.2 `_Connect() [1/2]`

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

### 13.7.3.3 `_Connect() [2/2]`

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

#### 13.7.3.4 `_GetDeviceName()`

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

#### 13.7.3.5 `_GetNode()`

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

#### 13.7.3.6 `_GetNodes()`

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

#### 13.7.3.7 `_GetSupportedSchemaVersions()`

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

#### 13.7.3.8 `_InvalidateNodes()`

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

#### 13.7.3.9 `_LoadXMLFromFile()`

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

#### 13.7.3.10 `_LoadXMLFromFileInject()`

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

### 13.7.3.11 `_LoadXMLFromString()`

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

### 13.7.3.12 `_LoadXMLFromStringInject()`

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

### 13.7.3.13 `_LoadXMLFromZIPData()`

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

### 13.7.3.14 `_LoadXMLFromZIPFile()`

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

### 13.7.3.15 `_Poll()`

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

### 13.7.3.16 `CacheChunkData()`

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

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

### 13.7.3.17 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)

### 13.7.3.18 Connect() [1/2]

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

Connects a port to the standard port "Device".

### 13.7.3.19 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.

### 13.7.3.20 Deregister()

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

Unregistering callback by handle.

### 13.7.3.21 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

### 13.7.3.22 EatComments()

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

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

### 13.7.3.23 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. |

### 13.7.3.24 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) |

### 13.7.3.25 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

### 13.7.3.26 GetAddress()

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

Retrieves the Address of the register.

### 13.7.3.27 GetAlias()

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

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

### 13.7.3.28 GetCachingMode()

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

Get Caching Mode.

### 13.7.3.29 GetCastAlias()

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

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

### 13.7.3.30 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**

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

**13.7.3.31 GetCookie()**

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

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

**13.7.3.32 GetCurrentEntry()**

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

Get the current entry.

**13.7.3.33 GetDescription()**

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

Get a long description of the node.

**13.7.3.34 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".

**13.7.3.35 GetDeviceVersion()**

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

Get the version of the device description file.

**13.7.3.36 GetDisplayName()**

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

Get a name string for display.

**13.7.3.37 GetDisplayNotation()**

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

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

**13.7.3.38 GetDisplayPrecision()**

```
virtual int64_t Spinnaker::GenApi::GetDisplayPrecision ( ) const [pure virtual]
```

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

**13.7.3.39 GetDocuURL()**

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

Gets a URL pointing to the documentation of that feature.

**13.7.3.40 GetEntries()**

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

Get list of entry nodes.

**13.7.3.41 GetEntry() [1/2]**

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

returns the [EnumEntry](#) object belonging to the Value

### 13.7.3.42 GetEntry() [2/2]

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

Get an entry node by its IntValue.

### 13.7.3.43 GetEntryByName()

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

Get an entry node by name.

### 13.7.3.44 GetEventID()

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

Get the EventId of the node.

### 13.7.3.45 GetGenApiVersion()

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

Get the version of the DLL's [GenApi](#) implementation.

### 13.7.3.46 GetInc()

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

Get the constant increment if there is any.

Get increment.

### 13.7.3.47 GetIncMode()

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

Get increment mode.

### 13.7.3.48 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

**13.7.3.49 GetLength()**

```
virtual int64_t Spinnaker::GenApi::GetLength ( ) const [pure virtual]
```

Retrieves the Length of the register [Bytes].

**13.7.3.50 GetListOfValidValues()**

```
int64_autovector_t GetListOfValidValues (
    bool bounded = true ) const [pure virtual]
```

Get list of valid value.

**13.7.3.51 GetLock()**

```
virtual Clock& Spinnaker::GenApi::GetLock ( ) const [pure virtual]
```

Returns the lock which guards the node map.

**13.7.3.52 GetMax()**

```
int64_t GetMax ( ) const [pure virtual]
```

Get maximum value allowed.

### 13.7.3.53 GetMaxLength()

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

Retrieves the maximum length of the string in bytes.

### 13.7.3.54 GetMin()

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

Get minimum value allowed.

### 13.7.3.55 GetNameSpace()

```
virtual GenApi::ENameSpace Spinnaker::GenApi::GetNameSpace ( ) const [pure virtual]
```

Get name space.

### 13.7.3.56 GetNode()

```
virtual INode* Spinnaker::GenApi::GetNode ( const GenICam::gcstring & Name ) const [pure virtual]
```

Retrieves the node from the central map by Name.

### 13.7.3.57 GetNodeMap()

```
virtual INodeMap* Spinnaker::GenApi::GetNodeMap ( ) const [pure virtual]
```

Retrieves the central node map.

### 13.7.3.58 GetNumericValue()

```
virtual double Spinnaker::GenApi::GetNumericValue ( ) [pure virtual]
```

Get double number associated with the entry.

### 13.7.3.59 GetNumNodes()

```
virtual uint64_t Spinnaker::GenApi::GetNumNodes ( ) const [pure virtual]
```

Get the number of nodes in the map.

### 13.7.3.60 GetParents()

```
virtual void Spinnaker::GenApi::GetParents ( GenApi::NodeList_t & Parents ) const [pure virtual]
```

Gets all nodes this node is directly depending on.

**Parameters**

|                  |                |                      |
|------------------|----------------|----------------------|
| <code>out</code> | <i>Parents</i> | List of parent nodes |
|------------------|----------------|----------------------|

**13.7.3.61 GetPollingTime()**

```
virtual int64_t Spinnaker::GenApi::GetPollingTime ( ) const [pure virtual]
```

recommended polling time (for non-cacheable nodes)

**13.7.3.62 GetPrincipalInterfaceType()**

```
virtual EInterfaceType Spinnaker::GenApi::GetPrincipalInterfaceType ( ) const [pure virtual]
```

Get the type of the main interface of a node.

**13.7.3.63 GetProductGuid()**

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

Get the Guid describing the product.

**13.7.3.64 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.

**13.7.3.65 GetPropertyNames()**

```
virtual void Spinnaker::GenApi::GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [pure virtual]
```

Returns a list of the names all properties set during initialization.

### 13.7.3.66 GetRepresentation()

```
ERepresentation GetRepresentation () [pure virtual]
```

Get recommended representation.

### 13.7.3.67 GetSchemaVersion()

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

Get the schema version number.

### 13.7.3.68 GetSelectedFeatures()

```
virtual void Spinnaker::GenApi::GetSelectedFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of selected features

### 13.7.3.69 GetSelectingFeatures()

```
virtual void Spinnaker::GenApi::GetSelectingFeatures (
    FeatureList_t & ) const [pure virtual]
```

retrieve the group of features selecting this node

### 13.7.3.70 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 |
|--------------------|------------------------------------------------------------------------------------------------------------|

**13.7.3.71 GetStandardNameSpace()**

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

Get the standard name space.

**13.7.3.72 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>" were <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

**13.7.3.73 GetSwapEndianess()**

```
virtual EYesNo Spinnaker::GenApi::GetSwapEndianess ( ) [pure virtual]
```

Determines if the port adapter must perform an endianess swap.

**13.7.3.74 GetSymbolic()**

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

Get symbolic enum value.

**13.7.3.75 GetToolTip()**

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

Get tool tip.

Get a short description of the node.

**13.7.3.76 GetUnit()**

```
GenICam::gcstring GetUnit ( ) const [pure virtual]
```

Get the physical unit name.

**13.7.3.77 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

**13.7.3.78 GetVendorName()**

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

Get the vendor name.

**13.7.3.79 GetVersionGuid()**

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

Get the Guid describing the product version.

**13.7.3.80 GetVisibility()**

```
virtual EVisibility Spinnaker::GenApi::GetVisibility ( ) const [pure virtual]
```

Get the recommended visibility of the node.

**13.7.3.81 HasInc()**

```
virtual bool Spinnaker::GenApi::HasInc ( ) [pure virtual]
```

True if the float has a constant increment.

**13.7.3.82 ImposeAccessMode()**

```
virtual void Spinnaker::GenApi::ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [pure virtual]
```

Imposes an access mode to the natural access mode of the node.

**13.7.3.83 ImposeMax() [1/2]**

```
virtual void Spinnaker::GenApi::ImposeMax (
    double Value ) [pure virtual]
```

Restrict maximum value.

**13.7.3.84 ImposeMax() [2/2]**

```
virtual void Spinnaker::GenApi::ImposeMax (
    int64_t Value ) [pure virtual]
```

Restrict maximum value.

**13.7.3.85 ImposeMin() [1/2]**

```
virtual void Spinnaker::GenApi::ImposeMin (
    double Value ) [pure virtual]
```

Restrict minimum value.

**13.7.3.86 ImposeMin() [2/2]**

```
virtual void Spinnaker::GenApi::ImposeMin (
    int64_t Value ) [pure virtual]
```

Restrict minimum value.

**13.7.3.87 ImposeVisibility()**

```
virtual void Spinnaker::GenApi::ImposeVisibility (
    EVisibility ImposedVisibility ) [pure virtual]
```

Imposes a visibility to the natural visibility of the node.

**13.7.3.88 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

### 13.7.3.89 InvalidateNodes()

```
virtual void Spinnaker::GenApi::InvalidateNodes( ) const [pure virtual]
```

Invalidates all nodes.

### 13.7.3.90 IsAccessModeCacheable()

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

True if the AccessMode can be cached.

### 13.7.3.91 IsCachable()

```
virtual bool Spinnaker::GenApi::IsCachable( ) const [pure virtual]
```

Is the node value cacheable.

### 13.7.3.92 IsDeprecated()

```
virtual bool Spinnaker::GenApi::IsDeprecated( ) const [pure virtual]
```

True if the node should not be used any more.

### 13.7.3.93 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

### 13.7.3.94 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".

### 13.7.3.95 IsSelfClearing()

```
virtual bool Spinnaker::GenApi::IsSelfClearing ( ) [pure virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

### 13.7.3.96 IsStreamable()

```
virtual bool Spinnaker::GenApi::IsStreamable ( ) const [pure virtual]
```

True if the node is streamable.

### 13.7.3.97 IsValueCacheValid()

```
virtual bool Spinnaker::GenApi::IsValueCacheValid ( ) const [pure virtual]
```

Checks if the value comes from cache or is requested from another node.

### 13.7.3.98 LoadXMLFromFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromFile (
    const GenICam::gcstring & FileName ) [pure virtual]
```

Loads an XML from a file.

### 13.7.3.99 LoadXMLFromFileInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromFileInject (
    const GenICam::gcstring & TargetFileName,
    const GenICam::gcstring & InjectFileName ) [pure virtual]
```

Loads an XML from a file with injection.

### 13.7.3.100 LoadXMLFromString()

```
virtual void Spinnaker::GenApi::LoadXMLFromString (
    const GenICam::gcstring & XMLData ) [pure virtual]
```

Loads an XML from a string.

### 13.7.3.101 LoadXMLFromStringInject()

```
virtual void Spinnaker::GenApi::LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData,
    const GenICam::gcstring & InjectXMLData ) [pure virtual]
```

Loads an XML from a string with injection.

### 13.7.3.102 LoadXMLFromZIPData()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize ) [pure virtual]
```

Loads an XML from a ZIP data buffer.

### 13.7.3.103 LoadXMLFromZIPFile()

```
virtual void Spinnaker::GenApi::LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName ) [pure virtual]
```

Loads an XML from a ZIP file.

### 13.7.3.104 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

**13.7.3.105 make\_NodeCallback() [2/2]**

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
    INode * pNode,
    Function function,
    ECallbackType CallbackType )
```

make a new callback object for C functions

**13.7.3.106 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 |

**13.7.3.107 operator"!=()**

```
virtual bool Spinnaker::GenApi::operator!= (
    int nullPtr ) const [pure virtual]
```

**13.7.3.108 operator()()**

```
GenICam::gcstring operator() ( ) const [virtual]
```

Get node value.

Execute the command.

**13.7.3.109 operator\*()**

```
GenICam::gcstring operator* ( ) [pure virtual]
```

Get string node value.

Get node value.

**13.7.3.110 operator<<()**

```
SPINNAKER_API std::ostream& Spinnaker::GenApi::operator<< (
    std::ostream & os,
    const CFeatureBag & FeatureBag )
```

writes out persistent data to a stream

**13.7.3.111 operator=() [1/5]**

```
virtual void Spinnaker::GenApi::operator= (
    bool Value ) [virtual]
```

Set node value.

**13.7.3.112 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

**13.7.3.113 operator=() [3/5]**

```
virtual IFloat& Spinnaker::GenApi::operator= (
    double Value ) [pure virtual]
```

Set node value.

**13.7.3.114 operator=() [4/5]**

```
virtual IEnumeration& Spinnaker::GenApi::operator= (
    EnumT Value ) [pure virtual]
```

Set node value.

**13.7.3.115 operator=() [5/5]**

```
virtual IInteger& Spinnaker::GenApi::operator= (
    int64_t Value ) [pure virtual]
```

Set node value.

**13.7.3.116 operator==( )**

```
virtual bool Spinnaker::GenApi::operator== (
    int nullPtr ) const [pure virtual]
```

**13.7.3.117 operator>>()**

```
SPINNAKER_API std::istream& Spinnaker::GenApi::operator>> (
    std::istream & is,
    CFeatureBag & FeatureBag )
```

Reads in persistent data from a stream.

**13.7.3.118 PersistFeature()**

```
virtual void Spinnaker::GenApi::PersistFeature (
    IValue & item ) [pure virtual]
```

Stores a feature.

**13.7.3.119 Poll()**

```
virtual void Spinnaker::GenApi::Poll (
    int64_t ElapsedTime ) [pure virtual]
```

Fires nodes which have a polling time.

**13.7.3.120 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 |

**13.7.3.121 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 |

**13.7.3.122 Register() [1/2]**

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Client & c,
    Member m,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C++-member function a callback.

**13.7.3.123 Register() [2/2]**

```
intptr_t Spinnaker::GenApi::Register (
    INode * pNode,
    Function f,
    ECallbackType CallbackType = cbPostInsideLock )
```

Register a C-function as a callback.

**13.7.3.124 RegisterCallback()**

```
virtual CallbackHandleType Spinnaker::GenApi::RegisterCallback (
    CNodeCallback * pCallback ) [pure virtual]
```

Register change callback Takes ownership of the **CNodeCallback** object.

**13.7.3.125 Replay()**

```
virtual void Spinnaker::GenApi::Replay (
    IPort * pPort ) [pure virtual]
```

Replays the write command to the given port interface.

**13.7.3.126 Restore()**

```
virtual void Spinnaker::GenApi::Restore ( ) [pure virtual]
```

Restores the selectors' values found at creation.

**13.7.3.127 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 )
```

**13.7.3.128 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.

**13.7.3.129 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) |

**13.7.3.130 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

**13.7.3.131 SetNumEnums()**

```
virtual void Spinnaker::GenApi::SetNumEnums (
    int NumEnums ) [pure virtual]
```

sets the number of enum values

**13.7.3.132 StopRecording()**

```
virtual void Spinnaker::GenApi::StopRecording () [pure virtual]
```

Stops recording.

**13.7.3.133 ToString() [1/2]**

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

Returns a string representation of the digit.

**13.7.3.134 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

**13.7.3.135 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.

**13.7.4 Variable Documentation****13.7.4.1 Address**

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

**13.7.4.2 COMMAND\_MAGIC**

```
const uint8_t COMMAND_MAGIC = 0x42
```

**13.7.4.3 GENCP\_COMMAND\_HEADER\_SIZE**

```
const size_t GENCP_COMMAND_HEADER_SIZE = sizeof(U3V_COMMAND_HEADER)
```

**13.7.4.4 GENCP\_EVENT\_BASIC\_SIZE**

```
const size_t GENCP_EVENT_BASIC_SIZE = sizeof(U3V_EVENT_MESSAGE)
```

#### 13.7.4.5 GENCP\_EVENT\_CMD\_ID

```
const uint16_t GENCP_EVENT_CMD_ID = 0x0C00
```

#### 13.7.4.6 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

##### Initial value:

```
{  
    virtual EAccessMode GetAccessMode() const = 0
```

Base interface common to all nodes.

#### 13.7.4.7 IBoolean

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

[Interface](#) for Boolean properties.

#### 13.7.4.8 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

#### 13.7.4.9 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

[Interface](#) for ports attached to a chunk.

#### 13.7.4.10 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

[Interface](#) for command like properties.

### 13.7.4.11 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

**Initial value:**

```
{  
    virtual void Destroy() = 0
```

**Interface** to destroy an object.

### 13.7.4.12 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

**Initial value:**

```
{  
    virtual GenICam::gcstring GetModelName() = 0
```

**Interface** to get information about the device (= nodemap)

### 13.7.4.13 IEnumEntry

```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

**Interface** of single enum value.

Maps of Enum Values to symbolic values

### 13.7.4.14 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

**Interface** for enumeration properties.

### 13.7.4.15 IEnumerationT

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

**Interface** for enumeration properties.

#### 13.7.4.16 IEnumReference

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

**Initial value:**

```
{  
    virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

[Interface](#) to construct an enum reference.

#### 13.7.4.17 IFloat

```
interface SPINNAKER_API_ABSTRACT IFloat
```

[Interface](#) for float properties.

#### 13.7.4.18 IInteger

```
interface SPINNAKER_API_ABSTRACT IInteger
```

[Interface](#) for integer properties.

#### 13.7.4.19 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

[Interface](#) common to all nodes.

#### 13.7.4.20 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

**Initial value:**

```
{  
    virtual void GetNodes(NodeList_t & Nodes) const = 0
```

[Interface](#) to access the node map.

#### 13.7.4.21 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

[Interface](#) to access the node map.

#### 13.7.4.22 Invalidate

```
interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0
```

#### 13.7.4.23 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

**Initial value:**

```
{  
    virtual void SetInfo(GenICam::gcstring & Info) = 0
```

Basic interface to persist values to.

#### 13.7.4.24 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

[Interface](#) for ports.

#### 13.7.4.25 IPortConstruct

```
interface SPINNAKER_API IPortConstruct
```

[Interface](#) for ports.

#### 13.7.4.26 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

[Interface](#) for recording write commands on a port.

### 13.7.4.27 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

[Interface](#) for replaying write commands on a port.

### 13.7.4.28 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

**Initial value:**

```
{  
    virtual void Write(const void* pBuffer, int64_t Address, int64_t Length) = 0
```

### 13.7.4.29 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

**Initial value:**

```
{  
    virtual GenICam::gcstring GetName(bool FullQualified = false) const = 0
```

[Interface](#) to construct a reference.

### 13.7.4.30 IRegister

```
interface SPINNAKER_API_ABSTRACT IRegister
```

[Interface](#) for registers.

### 13.7.4.31 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
```

[Interface](#) for groups of features selected by a single one.

### 13.7.4.32 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

**Initial value:**

```
{  
    virtual bool SetFirst() = 0
```

**Interface** of a "digit" of the "counter" formed by the selector set.

### 13.7.4.33 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

**Interface** for string properties.

### 13.7.4.34 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

**Interface** for value properties.

### 13.7.4.35 Length

```
interface SPINNAKER_API_ABSTRACT int64_t Length = 0
```

### 13.7.4.36 U3V\_EVENT\_PREFIX

```
const uint32_t U3V_EVENT_PREFIX = 0x45563355
```

### 13.7.4.37 Verify

```
interface SPINNAKER_API_ABSTRACT bool Verify = true) = 0
```

## 13.8 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 \(void\)](#)

- **SPINNAKER\_API gcstring GetGenICamCLProtocolFolder (void)**

*Retrieve the path of the GenICam logging properties file.*
- **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.*

### 13.8.1 Function Documentation

#### 13.8.1.1 DoesEnvironmentVariableExist()

```
SPINNAKER_API bool Spinnaker::GenICam::DoesEnvironmentVariableExist (
    const Spinnaker::GenICam::gcstring & VariableName )
```

Returns true if an environment variable exists.

#### 13.8.1.2 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            |

### 13.8.1.3 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.

### 13.8.1.4 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.

### 13.8.1.5 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.

### 13.8.1.6 getline() [1/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

STL getline.

### 13.8.1.7 getline() [2/2]

```
std::istream& Spinnaker::GenICam::getline (
    std::istream & is,
    Spinnaker::GenICam::gcstring & str,
    char delim ) [inline]
```

STL getline.

### 13.8.1.8 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

### 13.8.1.9 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 |
|--------------------------|--------------|

### 13.8.1.10 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

### 13.8.1.11 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`)

### 13.8.1.12 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. int64\_t) was downcast to type Td (e.g. int32\_t)

### 13.8.1.13 ReplaceEnvironmentVariables()

```
SPINNAKER_API void Spinnaker:::GenICam::ReplaceEnvironmentVariables (
    gcstring & Buffer,
    bool ReplaceBlankBy20 = false )
```

Replaces in a string and replace '' with %20.

### 13.8.1.14 SetGenICamCacheFolder()

```
SPINNAKER_API void Spinnaker:::GenICam::SetGenICamCacheFolder (
    const gcstring & path )
```

Stores the path of the [GenICam](#) cache folder.

### 13.8.1.15 SetGenICamCLProtocolFolder()

```
SPINNAKER_API void Spinnaker:::GenICam::SetGenICamCLProtocolFolder (
    const gcstring & path )
```

Stores the path of the CLProtocol folder.

### 13.8.1.16 SetGenICamLogConfig()

```
SPINNAKER_API void Spinnaker:::GenICam::SetGenICamLogConfig (
    const gcstring & path )
```

Stores the path of the [GenICam](#) logging properties file.

### 13.8.1.17 ThrowBadAlloc()

```
SPINNAKER_API void Spinnaker:::GenICam::ThrowBadAlloc ( )
```

### 13.8.1.18 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      |

**13.8.1.19 UrlDecode()**

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlDecode (
    const gcstring & Input )
```

Replaces xx escapes by their char equivalent.

**13.8.1.20 UrlEncode()**

```
SPINNAKER_API gcstring Spinnaker::GenICam::UrlEncode (
    const gcstring & Input )
```

Converts \ to / and replaces all unsave characters by their xx equivalent.

## 13.9 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 14

## Class Documentation

### 14.1 ActionCommandResult Struct Reference

Action Command Result.

#### Public Attributes

- unsigned int [DeviceAddress](#)
- [ActionCommandStatus](#) [Status](#)

#### 14.1.1 Detailed Description

Action Command Result.

#### 14.1.2 Member Data Documentation

##### 14.1.2.1 DeviceAddress

unsigned int [DeviceAddress](#)

##### 14.1.2.2 Status

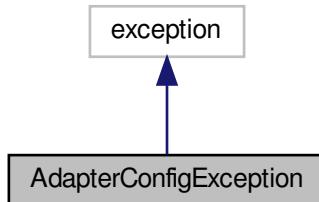
[ActionCommandStatus](#) [Status](#)

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

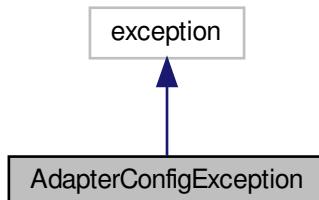
- include/SpinnakerDefs.h

## 14.2 AdapterConfigException Class Reference

Inheritance diagram for AdapterConfigException:



Collaboration diagram for AdapterConfigException:



### Public Member Functions

- `AdapterConfigException (const AdapterConfig::AdapterConfigErr errCode)`
- `AdapterConfigException (const AdapterConfig::AdapterConfigErr errCode, std::string param)`
- `AdapterConfig::AdapterConfigErr ErrCode () const`
- `std::string GetParamStr () const`

#### 14.2.1 Constructor & Destructor Documentation

##### 14.2.1.1 AdapterConfigException() [1/2]

```
AdapterConfigException (
    const AdapterConfig::AdapterConfigErr errCode ) [inline]
```

#### 14.2.1.2 AdapterConfigException() [2/2]

```
AdapterConfigException (
    const AdapterConfig::AdapterConfigErr errCode,
    std::string param ) [inline]
```

### 14.2.2 Member Function Documentation

#### 14.2.2.1 ErrCode()

```
AdapterConfig::AdapterConfigErr ErrCode () const [inline]
```

#### 14.2.2.2 GetParamStr()

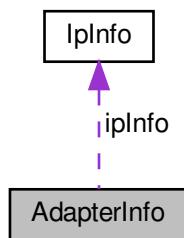
```
std::string GetParamStr () const [inline]
```

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

- [include/AdapterConfig.h](#)

## 14.3 AdapterInfo Struct Reference

Collaboration diagram for AdapterInfo:



### Public Member Functions

- [AdapterInfo \(\)](#)

## Public Attributes

- std::string [adapterName](#)
- std::string [adapterGUID](#)
- std::string [adapterMACAddress](#)
- std::string [adapterDescription](#)
- bool [dhcpEnabled](#)
- [IplInfo ipInfo](#)
- std::string [receiveBuffersRegKey](#)
- std::string [transmitBuffersRegKey](#)
- std::string [jumboPacketsRegKey](#)
- unsigned int [transmitBuffers](#)
- unsigned int [receiveBuffers](#)
- unsigned int [jumboPackets](#)
- unsigned int [receiveBuffersMin](#)
- unsigned int [receiveBuffersMax](#)
- unsigned int [receiveBuffersStep](#)
- unsigned int [transmitBuffersMin](#)
- unsigned int [transmitBuffersMax](#)
- unsigned int [transmitBuffersStep](#)
- std::vector< unsigned int > [jumboPacketValidValues](#)

### 14.3.1 Constructor & Destructor Documentation

#### 14.3.1.1 AdapterInfo()

```
AdapterInfo ( ) [inline]
```

### 14.3.2 Member Data Documentation

#### 14.3.2.1 adapterDescription

```
std::string adapterDescription
```

#### 14.3.2.2 adapterGUID

```
std::string adapterGUID
```

**14.3.2.3 adapterMACAddress**

```
std::string adapterMACAddress
```

**14.3.2.4 adapterName**

```
std::string adapterName
```

**14.3.2.5 dhcpEnabled**

```
bool dhcpEnabled
```

**14.3.2.6 ipInfo**

```
IpInfo ipInfo
```

**14.3.2.7 jumboPackets**

```
unsigned int jumboPackets
```

**14.3.2.8 jumboPacketsRegKey**

```
std::string jumboPacketsRegKey
```

**14.3.2.9 jumboPacketValidValues**

```
std::vector<unsigned int> jumboPacketValidValues
```

**14.3.2.10 receiveBuffers**

```
unsigned int receiveBuffers
```

**14.3.2.11 receiveBuffersMax**

```
unsigned int receiveBuffersMax
```

**14.3.2.12 receiveBuffersMin**

```
unsigned int receiveBuffersMin
```

**14.3.2.13 receiveBuffersRegKey**

```
std::string receiveBuffersRegKey
```

**14.3.2.14 receiveBuffersStep**

```
unsigned int receiveBuffersStep
```

**14.3.2.15 transmitBuffers**

```
unsigned int transmitBuffers
```

**14.3.2.16 transmitBuffersMax**

```
unsigned int transmitBuffersMax
```

**14.3.2.17 transmitBuffersMin**

```
unsigned int transmitBuffersMin
```

**14.3.2.18 transmitBuffersRegKey**

```
std::string transmitBuffersRegKey
```

### 14.3.2.19 transmitBuffersStep

```
unsigned int transmitBuffersStep
```

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

- [include/AdapterConfig.h](#)

## 14.4 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.*

### 14.4.1 Detailed Description

Delivers information about the attached chunks and nodes.

### 14.4.2 Member Data Documentation

#### 14.4.2.1 NumAttachedChunks

```
int NumAttachedChunks
```

Number of chunks from the buffer attached to a chunk port.

#### 14.4.2.2 NumChunkPorts

```
int NumChunkPorts
```

Number of chunk ports found in the node map.

#### 14.4.2.3 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](#)

## 14.5 AutoLock Class Reference

### Public Member Functions

- [AutoLock \(CLock &lock\)](#)
- [~AutoLock \(\)](#)

#### 14.5.1 Constructor & Destructor Documentation

##### 14.5.1.1 AutoLock()

```
AutoLock (
```

```
    CLock & lock ) [inline]
```

##### 14.5.1.2 ~AutoLock()

```
~AutoLock ( ) [inline]
```

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

- include/SpinGenApi/[GCSynch.h](#)

## 14.6 AutoLock Class Reference

### Public Member Functions

- [AutoLock \(CLock &lock\)](#)
- [~AutoLock \(\)](#)

### 14.6.1 Constructor & Destructor Documentation

#### 14.6.1.1 AutoLock()

```
AutoLock (   
    CLock & lock ) [inline]
```

#### 14.6.1.2 ~AutoLock()

```
~AutoLock ( ) [inline]
```

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

- include/SpinGenApi/Synch.h

## 14.7 AVIOption Struct Reference

Options for saving AVI files.

### Public Member Functions

- [AVIOption \(\)](#)

### Public Attributes

- float [frameRate](#)  
*Frame rate of the stream.*
- unsigned int [reserved](#) [256]  
*Reserved for future use.*

### 14.7.1 Detailed Description

Options for saving AVI files.

### 14.7.2 Constructor & Destructor Documentation

#### 14.7.2.1 AVIOption()

```
AVIOption ( ) [inline]
```

### 14.7.3 Member Data Documentation

#### 14.7.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

#### 14.7.3.2 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

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

- [include/SpinVideoDefs.h](#)

## 14.8 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 T \* [get](#) () const
 

*get()*

## Protected Attributes

- PointerData \* [m\\_pT](#)

*Underlying raw pointer.*

### 14.8.1 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.

### 14.8.2 Constructor & Destructor Documentation

#### 14.8.2.1 [BasePtr\(\)](#) [1/2]

```
BasePtr (
    void ) throw ( )
```

Default constructor.

#### 14.8.2.2 [~BasePtr\(\)](#)

```
virtual ~BasePtr (
    void ) [virtual]
```

#### 14.8.2.3 BasePtr() [2/2]

```
BasePtr (   
    const BasePtr< T, B > & other ) throw ( )
```

Copy constructor.

### 14.8.3 Member Function Documentation

#### 14.8.3.1 get()

```
virtual T* get ( ) const [virtual]
```

**get()**

#### 14.8.3.2 IsValid()

```
virtual bool IsValid ( ) const throw ( ) [virtual]
```

True if the pointer is valid.

#### 14.8.3.3 operator bool()

```
virtual operator bool (   
    void ) const throw ( ) [virtual]
```

True if the pointer is valid.

#### 14.8.3.4 operator T\*()

```
virtual operator T* (   
    void ) const [virtual]
```

Dereferencing.

**14.8.3.5 operator->()**

```
virtual T* operator-> (
    void ) const [virtual]
```

Dereferencing.

**14.8.3.6 operator=() [1/4]**

```
virtual BasePtr& operator= (
    const BasePtr< T, B > & rhs ) [virtual]
```

Assign INode Pointer.

**14.8.3.7 operator=() [2/4]**

```
virtual BasePtr& operator= (
    const int nMustBeNull ) [virtual]
```

**14.8.3.8 operator=() [3/4]**

```
virtual BasePtr& operator= (
    const long nMustBeNull ) [virtual]
```

**14.8.3.9 operator=() [4/4]**

```
virtual BasePtr& operator= (
    const std::nullptr_t nullPtr ) [virtual]
```

**14.8.3.10 operator==( ) [1/4]**

```
virtual bool operator== (
    const BasePtr< T, B > & rT ) const [virtual]
```

Pointer equal.

**14.8.3.11 operator==( ) [2/4]**

```
virtual bool operator== (
    int nMustBeNull ) const [virtual]
```

Pointer equal.

**14.8.3.12 operator==( ) [3/4]**

```
virtual bool operator== (
    long nMustBeNull ) const [virtual]
```

Pointer equal.

**14.8.3.13 operator==( ) [4/4]**

```
virtual bool operator== (
    std::nullptr_t ) const [virtual]
```

Pointer equal.

**14.8.4 Member Data Documentation****14.8.4.1 m\_pT**

```
PointerData* m_pT [protected]
```

Underlying raw pointer.

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

- [include/BasePtr.h](#)

**14.9 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.*

### 14.9.1 Detailed Description

Options for saving Bitmap image.

### 14.9.2 Constructor & Destructor Documentation

#### 14.9.2.1 BMPOption()

[BMPOption](#) ( ) [inline]

### 14.9.3 Member Data Documentation

#### 14.9.3.1 indexedColor\_8bit

bool indexedColor\_8bit

#### 14.9.3.2 reserved

unsigned int reserved[16]

Reserved for future use.

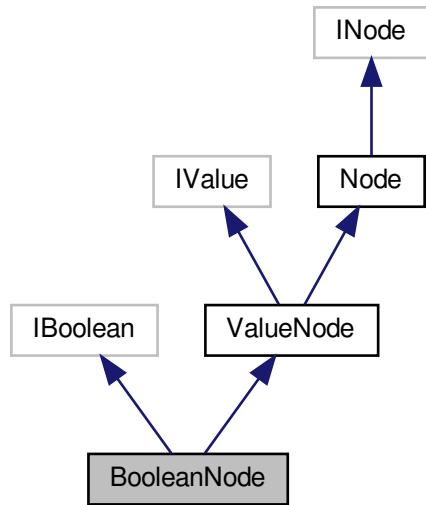
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

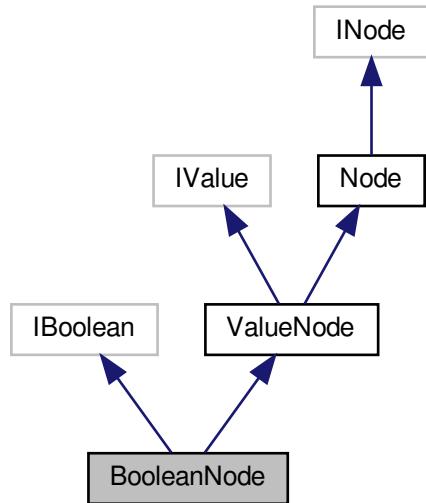
## 14.10 BooleanNode Class Reference

[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

### 14.10.1 Detailed Description

[Interface](#) for string properties.

### 14.10.2 Constructor & Destructor Documentation

#### 14.10.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

#### 14.10.2.2 BooleanNode() [2/2]

```
BooleanNode ( std::shared_ptr< Node::NodeImpl > pBoolean )
```

#### 14.10.2.3 ~BooleanNode()

```
virtual ~BooleanNode ( ) [virtual]
```

### 14.10.3 Member Function Documentation

#### 14.10.3.1 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.

**14.10.3.2 operator=( )**

```
virtual void operator= (
    bool Value ) [virtual]
```

Set node value.

**14.10.3.3 SetReference( )**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

**14.10.3.4 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). |

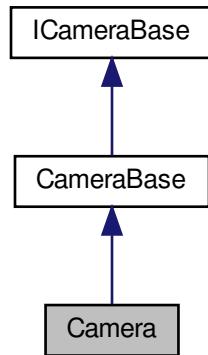
The documentation for this class was generated from the following file:

- include/SpinGenApi/[BooleanNode.h](#)

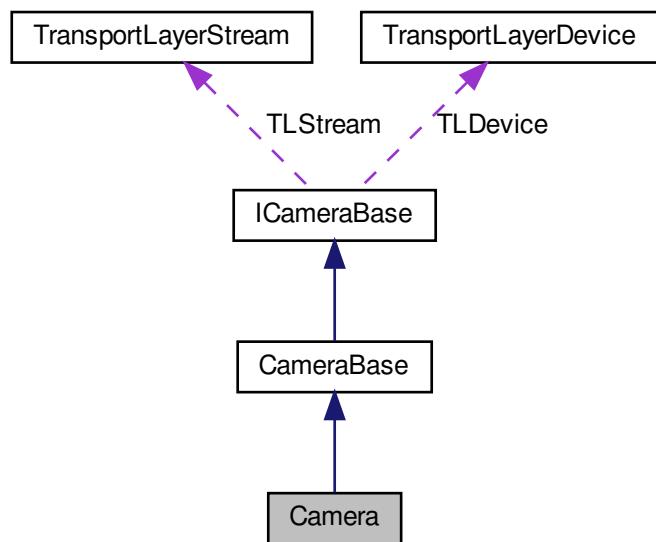
## 14.11 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



### Public Member Functions

- `~Camera ()`
- `void Init ()`

## Public Attributes

- `GenApi::IInteger & LUTIndex`
- `GenApi::IBoolean & LUTEnable`
- `GenApi::IInteger & LUTValue`
- `GenApi::IEnumerationT< LUTSelectorEnums > & LUTSelector`
- `GenApi::IFloat & ExposureTime`
- `GenApi:: ICommand & AcquisitionStop`

*Description: This command stops the acquisition of images.*

- `GenApi::IFloat & AcquisitionResultingFrameRate`

*Description: Resulting frame rate in Hertz.*

- `GenApi::IFloat & AcquisitionLineRate`

*Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.*

- `GenApi:: ICommand & AcquisitionStart`

*Description: This command starts the acquisition of images.*

- `GenApi:: ICommand & TriggerSoftware`

- `GenApi::IEnumerationT< ExposureModeEnums > & ExposureMode`

- `GenApi::IEnumerationT< AcquisitionModeEnums > & AcquisitionMode`

*Description: Sets the acquisition mode of the device.*

- `GenApi::IInteger & AcquisitionFrameCount`

- `GenApi::IEnumerationT< 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::IInteger & AcquisitionBurstFrameCount`

- `GenApi::IInteger & EventTest`

*Description: Returns the unique identifier of the Test type of Event.*

- `GenApi::IInteger & EventTestTimestamp`

*Description: Returns the Timestamp of the Test Event.*

- `GenApi::IInteger & EventExposureEndFrameID`

*Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.*

- `GenApi::IInteger & EventExposureEnd`

*Description: Returns the unique identifier of the Exposure End type of Event.*

- `GenApi::IInteger & EventExposureEndTimestamp`

*Description: Returns the Timestamp of the Exposure End Event.*

- `GenApi::IInteger & EventError`

*Description: Returns the unique identifier of the Error type of Event.*

- `GenApi::IInteger & EventErrorTimestamp`

*Description: Returns the Timestamp of the Error Event.*

- [GenApi::IInteger & EventErrorCode](#)  
*Description:* Returns the error code for the error that happened *Visibility*:
- [GenApi::IInteger & 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::IInteger & EventSerialPortReceive](#)  
*Description:* Returns the unique identifier of the Serial Port Receive type of Event.
- [GenApi::IInteger & EventSerialPortReceiveTimestamp](#)  
*Description:* Returns the Timestamp of the Serial Port Receive Event.
- [GenApi::IString & EventSerialData](#)  
*Description:* Returns the serial data that was received.
- [GenApi::IInteger & 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::IInteger & 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 & LogicBlockLUTOOutputValue](#)  
*Description:* Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.
- [GenApi::IInteger & LogicBlockLUTOOutputValueAll](#)  
*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](#)

- **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: Endianess 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::IInteger & DeviceLinkThroughputLimit**
- **GenApi::IString & DeviceFirmwareVersion**  
*Description: Version of the firmware on the device.*
- **GenApi::IInteger & 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::IInteger & DeviceTLVersionMajor**
- **GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > & DeviceTemperatureSelector**
- **GenApi::IInteger & EnumerationCount**  
*Description: Number of enumerations since uptime.*
- **GenApi::IFloat & PowerSupplyCurrent**
- **GenApi::IString & DeviceID**  
*Description: Device identifier (serial number).*
- **GenApi::IInteger & DeviceUptime**  
*Description: Total time since the device was powered up in seconds.*
- **GenApi::IInteger & DeviceLinkCurrentThroughput**  
*Description: Current bandwidth of streamed data.*
- **GenApi::IInteger & 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::IInteger & AasRoiOffsetY
- GenApi::IInteger & AasRoiOffsetX
- GenApi::IEnumerationT< AutoExposureControlPriorityEnums > & AutoExposureControlPriority
- GenApi::IFloat & BalanceWhiteAutoLowerLimit
- GenApi::IFloat & BalanceWhiteAutoDamping
- GenApi::IInteger & 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 & GevSCPSDoNotFragment

*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::IInteger & GevSCPIfaceIndex](#)

*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::IInteger & 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::IInteger & 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::IInteger & GevHeartbeatTimeout](#)

*Description:* Indicates the current heartbeat timeout in milliseconds.
  - [GenApi::IInteger & 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::IInteger & GevMCDA](#)

*Description:* Controls the destination IP address of the message channel Visibility:
  - [GenApi::IInteger & 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::IInteger & 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::IEnumeratorT< GevSupportedOptionSelectorEnums > & GevSupportedOptionSelector](#)

*Description:* Selects the GEV option to interrogate for existing support.

- [GenApi::IBoolean & GevGVCPHeartbeatDisable](#)

*Description:* Disables the GVCP heartbeat.

- [GenApi::IInteger & GevMCRC](#)

*Description:* Indicates the number of retries of the message channel.

- [GenApi::IBoolean & GevSCPSBigEndian](#)

*Description:* Endianess of multi-byte pixel data for this stream.

- [GenApi::IInteger & GevNumberOfInterfaces](#)

*Description:* Indicates the number of physical network interfaces supported by this device.

- [GenApi::IInteger & TLParamsLocked](#)

*Description:* Visibility:

- [GenApi::IInteger & PayloadSize](#)

*Description:* Provides the number of bytes transferred for each image or chunk on the stream channel.

- [GenApi::IInteger & PacketResendRequestCount](#)

*Description:* Counts the number of resend requests received from the host.

- [GenApi::IBoolean & SharpeningEnable](#)

- [GenApi::IEnumeratorT< BlackLevelSelectorEnums > & BlackLevelSelector](#)

- [GenApi::IBoolean & GammaEnable](#)

*Description:* Enables/disables gamma correction.

- [GenApi::IBoolean & SharpeningAuto](#)

- [GenApi::IBoolean & BlackLevelClampingEnable](#)

- [GenApi::IFloat & BalanceRatio](#)

- [GenApi::IEnumeratorT< BalanceWhiteAutoEnums > & BalanceWhiteAuto](#)

- [GenApi::IFloat & SharpeningThreshold](#)

- [GenApi::IEnumeratorT< GainAutoEnums > & GainAuto](#)

- [GenApi::IFloat & Sharpening](#)

- [GenApi::IFloat & Gain](#)

- [GenApi::IEnumeratorT< BalanceRatioSelectorEnums > & BalanceRatioSelector](#)

- [GenApi::IEnumeratorT< GainSelectorEnums > & GainSelector](#)

*Description:* Selects which gain to control.

- [GenApi::IFloat & BlackLevel](#)

- [GenApi::IInteger & BlackLevelRaw](#)

- [GenApi::IFloat & Gamma](#)

*Description:* Controls the gamma correction of pixel intensity.

- [GenApi::IInteger & DefectTableIndex](#)

- [GenApi::ICommand & DefectTableFactoryRestore](#)

*Description:* Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.

- [GenApi::IInteger & DefectTableCoordinateY](#)

- [GenApi::ICommand & DefectTableSave](#)

*Description:* Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.

- [GenApi::IEnumeratorT< DefectCorrectionModeEnums > & DefectCorrectionMode](#)

*Description:* Controls the method used for replacing defective pixels.

- [GenApi::IInteger & DefectTableCoordinateX](#)  
*Description: Enables/Disables table-based defective pixel correction.*
- [GenApi::ICommand & DefectTableApply](#)  
*Description: Applies the current defect table, so that any changes made affect images captured by the camera.*
- [GenApi::IBoolean & UserSetFeatureEnable](#)  
*Description: Whether or not the selected feature is saved to user sets.*
- [GenApi::ICommand & UserSetSave](#)
- [GenApi::IEnumerationT< UserSetSelectorEnums > & UserSetSelector](#)
- [GenApi::ICommand & UserSetLoad](#)
- [GenApi::IEnumerationT< UserSetDefaultEnums > & UserSetDefault](#)
- [GenApi::IEnumerationT< SerialPortBaudRateEnums > & SerialPortBaudRate](#)  
*Description: This feature controls the baud rate used by the selected serial port.*
- [GenApi::IInteger & SerialPortDataBits](#)  
*Description: This feature controls the number of data bits used by the selected serial port.*
- [GenApi::IEnumerationT< SerialPortParityEnums > & SerialPortParity](#)  
*Description: This feature controls the parity used by the selected serial port.*
- [GenApi::IInteger & SerialTransmitQueueMaxCharacterCount](#)  
*Description: >Returns the maximum number of characters in the serial port transmit queue.*
- [GenApi::IInteger & 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::IInteger & SerialReceiveFramingErrorCount](#)  
*Description: Returns the number of framing errors that have occurred on the serial port.*
- [GenApi::IInteger & SerialTransmitQueueCurrentCharacterCount](#)  
*Description: Returns the number of characters currently in the serial port transmit queue.*
- [GenApi::IInteger & 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::IInteger & SerialReceiveQueueMaxCharacterCount](#)  
*Description: >Returns the maximum number of characters in the serial port receive queue.*
- [GenApi::IInteger & 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::IInteger & SequencerSetSelector](#)
- [GenApi::IEnumerationT< SequencerTriggerActivationEnums > & SequencerTriggerActivation](#)
- [GenApi::IEnumerationT< SequencerConfigurationModeEnums > & SequencerConfigurationMode](#)
- [GenApi::ICommand & SequencerSetSave](#)
- [GenApi::IEnumerationT< SequencerTriggerSourceEnums > & SequencerTriggerSource](#)
- [GenApi::IInteger & SequencerSetActive](#)  
*Description: Displays the currently active sequencer set.*
- [GenApi::IInteger & SequencerSetNext](#)

- Description:* Specifies the next sequencer set.

  - [GenApi::ICommand & SequencerSetLoad](#)
  - [GenApi::IInteger & SequencerPathSelector](#)
  - [GenApi::IBoolean & SequencerFeatureEnable](#)
  - [GenApi::IInteger & 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::IInteger & TransferQueueMaxBlockCount](#)
- Description:* Returns the maximum number of data blocks (images) in the transfer queue Visibility:

  - [GenApi::IInteger & TransferQueueCurrentBlockCount](#)
- Description:* Returns number of data blocks (images) currently in the transfer queue.

  - [GenApi::IEnumeratorT<TransferQueueModeEnums> & TransferQueueMode](#)
- Description:* Specifies the operation mode of the transfer queue.

  - [GenApi::IEnumeratorT<TransferOperationModeEnums> & TransferOperationMode](#)
- Description:* Selects the operation mode of the transfer.

  - [GenApi::ICommand & TransferStop](#)
- Description:* Stops the streaming of data block (images).

  - [GenApi::IInteger & TransferQueueOverflowCount](#)
- Description:* Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

  - [GenApi::IEnumeratorT<TransferControlModeEnums> & TransferControlMode](#)
- Description:* Selects the control method for the transfers.

  - [GenApi::IFloat & ChunkBlackLevel](#)
- Description:* Returns the black level used to capture the image.

  - [GenApi::IString & 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::IBoolean & ChunkSerialReceiveOverflow](#)
- Description:* Returns the status of the chunk serial receive overflow.

  - [GenApi::IInteger & 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::IInteger & ChunkExposureEndLineStatusAll](#)
- Description:* Returns the status of all the I/O lines at the end of exposure event.

  - [GenApi::IEnumeratorT<ChunkGainSelectorEnums> & ChunkGainSelector](#)
- Description:* Selects which gain to retrieve Visibility:

  - [GenApi::IEnumeratorT<ChunkSelectorEnums> & ChunkSelector](#)
- Description:* Selects which chunk data to enable or disable.

  - [GenApi::IEnumeratorT<ChunkBlackLevelSelectorEnums> & ChunkBlackLevelSelector](#)
- Description:* Selects which black level to retrieve Visibility:

  - [GenApi::IInteger & ChunkWidth](#)
- Description:* Returns the width of the image included in the payload.

  - [GenApi::IInteger & ChunkImage](#)
- Description:* Returns the image payload.

  - [GenApi::IInteger & 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::IInteger & **ChunkSequencerSetActive**  
*Description: Returns the index of the active set of the running sequencer included in the payload.*
- GenApi::IInteger & **ChunkCRC**  
*Description: Returns the CRC of the image payload.*
- GenApi::IInteger & **ChunkOffsetX**  
*Description: Returns the Offset X of the image included in the payload.*
- GenApi::IInteger & **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::IInteger & **ChunkSerialDataLength**  
*Description: Returns the length of the received serial data that was included in the payload.*
- GenApi::IInteger & **FileAccessOffset**  
*Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.*
- GenApi::IInteger & **FileAccessLength**  
*Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.*
- GenApi::IEnumerationT< FileOperationStatusEnums > & **FileOperationStatus**  
*Description: Represents the file operation execution status.*
- GenApi::ICommand & **FileOperationExecute**
- GenApi::IEnumerationT< FileModeEnums > & **FileOpenMode**
- GenApi::IInteger & **FileOperationResult**  
*Description: Represents the file operation result.*
- GenApi::IEnumerationT< FileOperationSelectorEnums > & **FileOperationSelector**
- GenApi::IEnumerationT< FileSelectorEnums > & **FileSelector**
- GenApi::IInteger & **FileSize**  
*Description: Represents the size of the selected file in bytes.*
- GenApi::IEnumerationT< BinningSelectorEnums > & **BinningSelector**
- GenApi::IInteger & **PixelDynamicRangeMin**  
*Description: Minimum value that can be returned during the digitization process.*
- GenApi::IInteger & **PixelDynamicRangeMax**  
*Description: Maximum value that can be returned during the digitization process.*
- GenApi::IInteger & **OffsetY**
- GenApi::IInteger & **BinningHorizontal**
- GenApi::IInteger & **Width**
- GenApi::IEnumerationT< TestPatternGeneratorSelectorEnums > & **TestPatternGeneratorSelector**
- GenApi::IFloat & **CompressionRatio**  
*Description: Reports the ratio between the uncompressed image size and compressed image size.*
- GenApi::IBoolean & **ReverseX**  
*Description: Horizontally flips the image sent by the device.*
- GenApi::IBoolean & **ReverseY**  
*Description: Vertically flips the image sent by the device.*
- GenApi::IEnumerationT< TestPatternEnums > & **TestPattern**
- GenApi::IEnumerationT< PixelColorFilterEnums > & **PixelColorFilter**  
*Description: Type of color filter that is applied to the image.*
- GenApi::IInteger & **WidthMax**
- GenApi::IEnumerationT< AdcBitDepthEnums > & **AdcBitDepth**
- GenApi::IInteger & **BinningVertical**
- GenApi::IEnumerationT< DecimationHorizontalModeEnums > & **DecimationHorizontalMode**

- [GenApi::IEnumerationT< BinningVerticalModeEnums > & BinningVerticalMode](#)  
*Description: Visibility:*
  - [GenApi::IInteger & OffsetX](#)
  - [GenApi::IInteger & HeightMax](#)
  - [GenApi::IInteger & DecimationHorizontal](#)
  - [GenApi::IEnumerationT< PixelSizeEnums > & PixelSize](#)  
*Description: Total size in bits of a pixel of the image.*
  - [GenApi::IInteger & SensorHeight](#)  
*Description: Effective height of the sensor in pixels.*
  - [GenApi::IEnumerationT< DecimationSelectorEnums > & DecimationSelector](#)  
*Description: Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.*
  - [GenApi::IBoolean & IspEnable](#)
  - [GenApi::IBoolean & AdaptiveCompressionEnable](#)  
*Description: Controls whether lossless compression adapts to the image content.*
  - [GenApi::IEnumerationT< ImageCompressionModeEnums > & ImageCompressionMode](#)  
*Description: Visibility:*
    - [GenApi::IInteger & DecimationVertical](#)
    - [GenApi::IInteger & Height](#)
    - [GenApi::IEnumerationT< BinningHorizontalModeEnums > & BinningHorizontalMode](#)  
*Description: Visibility:*
      - [GenApi::IEnumerationT< PixelFormatEnums > & PixelFormat](#)  
*Description: Format of the pixel provided by the camera.*
      - [GenApi::IInteger & 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::IInteger & GuiXmlManifestAddress](#)  
*Description: Location of the GUI XML manifest table.*
      - [GenApi::IInteger & 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::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).*
      - [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::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).
  - [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::IFloat & 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::IInteger & CounterValue](#)

*Description:* Current counter value Visibility:
  - [GenApi::IEnumerationT< CounterSelectorEnums > & CounterSelector](#)

*Description:* Selects which counter to configure Visibility:
  - [GenApi::IInteger & CounterValueAtReset](#)

*Description:* Value of the selected Counter when it was reset by a trigger.
  - [GenApi::IEnumerationT< CounterStatusEnums > & CounterStatus](#)

*Description:* Returns the current status of the Counter.
  - [GenApi::IEnumerationT< CounterTriggerSourceEnums > & CounterTriggerSource](#)

*Description:* Selects the source of the trigger to start the counter Visibility:
  - [GenApi::IInteger & CounterDelay](#)

*Description:* Sets the delay (or number of events) before the CounterStart event is generated.
  - [GenApi::IEnumerationT< CounterResetSourceEnums > & CounterResetSource](#)

*Description:* Selects the signal that will be the source to reset the Counter.
  - [GenApi::IEnumerationT< CounterEventSourceEnums > & CounterEventSource](#)

*Description:* Selects the event that will increment the counter Visibility:
  - [GenApi::IEnumerationT< CounterEventActivationEnums > & CounterEventActivation](#)

*Description:* Selects the activation mode of the event to increment the Counter.
  - [GenApi::IInteger & CounterDuration](#)

*Description:* Sets the duration (or number of events) before the CounterEnd event is generated.
  - [GenApi::IEnumerationT< CounterResetActivationEnums > & CounterResetActivation](#)

*Description:* Selects the Activation mode of the Counter Reset Source signal.
  - [GenApi::IEnumerationT< DeviceTypeEnums > & DeviceType](#)

*Description:* Returns the device type.
  - [GenApi::IString & DeviceFamilyName](#)

*Description:* Identifier of the product family of the device.
  - [GenApi::IInteger & DeviceSFNCVersionMajor](#)

*Description:* Major version of the Standard Features Naming Convention that was used to create the device's GenICam XML.
  - [GenApi::IInteger & DeviceSFNCVersionMinor](#)

*Description:* Minor version of the Standard Features Naming Convention that was used to create the device's GenICam XML.
  - [GenApi::IInteger & DeviceSFNCVersionSubMinor](#)

*Description:* Sub minor version of Standard Features Naming Convention that was used to create the device's GenICam XML.
  - [GenApi::IInteger & DeviceManifestEntrySelector](#)

*Description:* Selects the manifest entry to reference.

- **GenApi::IInteger & DeviceManifestXMLMajorVersion**  
*Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.*
- **GenApi::IInteger & DeviceManifestXMLMinorVersion**  
*Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.*
- **GenApi::IInteger & DeviceManifestXMLSubMinorVersion**  
*Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.*
- **GenApi::IInteger & DeviceManifestSchemaMajorVersion**  
*Description: Indicates the major version number of the schema file of the selected manifest entry.*
- **GenApi::IInteger & 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::IInteger & DeviceTLVersionSubMinor**  
*Description: Sub minor version of the Transport Layer of the device.*
- **GenApi::IInteger & DeviceGenCPVersionMajor**  
*Description: Major version of the GenCP protocol supported by the device.*
- **GenApi::IInteger & DeviceGenCPVersionMinor**  
*Description: Minor version of the GenCP protocol supported by the device.*
- **GenApi::IInteger & DeviceConnectionSelector**  
*Description: Selects which Connection of the device to control.*
- **GenApi::IInteger & DeviceConnectionSpeed**  
*Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.*
- **GenApi::IEnumerationT< DeviceConnectionStatusEnums > & DeviceConnectionStatus**  
*Description: Indicates the status of the specified Connection.*
- **GenApi::IInteger & DeviceLinkSelector**  
*Description: Selects which Link of the device to control.*
- **GenApi::IEnumerationT< DeviceLinkThroughputLimitModeEnums > & DeviceLinkThroughputLimitMode**  
*Description: Controls if the DeviceLinkThroughputLimit is active.*
- **GenApi::IInteger & DeviceLinkConnectionCount**  
*Description: Returns the number of physical connection of the device used by a particular Link.*
- **GenApi::IEnumerationT< DeviceLinkHeartbeatModeEnums > & DeviceLinkHeartbeatMode**  
*Description: Activate or deactivate the Link's heartbeat.*
- **GenApi::IFloat & DeviceLinkHeartbeatTimeout**  
*Description: Controls the current heartbeat timeout of the specific Link.*
- **GenApi::IFloat & DeviceLinkCommandTimeout**  
*Description: Indicates the command timeout of the specified Link.*
- **GenApi::IInteger & DeviceStreamChannelSelector**  
*Description: Selects the stream channel to control.*
- **GenApi::IEnumerationT< DeviceStreamChannelTypeEnums > & DeviceStreamChannelType**  
*Description: Reports the type of the stream channel.*
- **GenApi::IInteger & DeviceStreamChannelLink**  
*Description: Index of device's Link to use for streaming the specified stream channel.*
- **GenApi::IEnumerationT< DeviceStreamChannelEndiannessEnums > & DeviceStreamChannelEndianness**  
*Description: Endianness of multi-byte pixel data for this stream.*
- **GenApi::IInteger & 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::IBoolean & DeviceRegistersValid](#)  
*Description: Returns if the current register set is valid and consistent.*
- [GenApi::IEnumerationT< DeviceClockSelectorEnums > & DeviceClockSelector](#)  
*Description: Selects the clock frequency to access from the device.*
- [GenApi::IFloat & 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::IInteger & 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::IBoolean & ImageComponentEnable](#)  
*Description: Controls if the selected component streaming is active.*
- [GenApi::IInteger & 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::IInteger & 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::IInteger & ImageCompressionQuality](#)  
*Description: Control the quality of the produced compressed stream.*
- [GenApi::IFloat & 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::IBoolean & AcquisitionStatus](#)  
*Description:* Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.
- [GenApi::IInteger & TriggerDivider](#)  
*Description:* Specifies a division factor for the incoming trigger pulses.
- [GenApi::IInteger & 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::IInteger & 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::IInteger & 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::IInteger & EncoderValue](#)  
*Description: Reads or writes the current value of the position counter of the selected Encoder.*
- [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.*
- [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::IInteger & ActionDeviceKey](#)  
*Description: Provides the device key that allows the device to check the validity of action commands.*
- [GenApi::IInteger & ActionQueueSize](#)  
*Description: Indicates the size of the scheduled action commands queue.*
- [GenApi::IInteger & ActionSelector](#)  
*Description: Selects to which Action Signal further Action settings apply.*
- [GenApi::IInteger & ActionGroupMask](#)  
*Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::IInteger & ActionGroupKey](#)  
*Description: Provides the key that the device will use to validate the action on reception of the action protocol message.*
- [GenApi::IInteger & EventAcquisitionTrigger](#)  
*Description: Returns the unique Identifier of the Acquisition Trigger type of Event.*
- [GenApi::IInteger & EventAcquisitionTriggerTimestamp](#)  
*Description: Returns the Timestamp of the Acquisition Trigger Event.*

- [GenApi::IInteger & EventAcquisitionTriggerFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.
- [GenApi::IInteger & EventAcquisitionStart](#)  
*Description:* Returns the unique Identifier of the Acquisition Start type of Event.
- [GenApi::IInteger & EventAcquisitionStartTimestamp](#)  
*Description:* Returns the Timestamp of the Acquisition Start Event.
- [GenApi::IInteger & EventAcquisitionStartFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.
- [GenApi::IInteger & EventAcquisitionEnd](#)  
*Description:* Returns the unique Identifier of the Acquisition End type of Event.
- [GenApi::IInteger & EventAcquisitionEndTimestamp](#)  
*Description:* Returns the Timestamp of the Acquisition End Event.
- [GenApi::IInteger & EventAcquisitionEndFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.
- [GenApi::IInteger & EventAcquisitionTransferStart](#)  
*Description:* Returns the unique Identifier of the Acquisition Transfer Start type of Event.
- [GenApi::IInteger & EventAcquisitionTransferStartTimestamp](#)  
*Description:* Returns the Timestamp of the Acquisition Transfer Start Event.
- [GenApi::IInteger & EventAcquisitionTransferStartFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.
- [GenApi::IInteger & EventAcquisitionTransferEnd](#)  
*Description:* Returns the unique Identifier of the Acquisition Transfer End type of Event.
- [GenApi::IInteger & EventAcquisitionTransferEndTimestamp](#)  
*Description:* Returns the Timestamp of the Acquisition Transfer End Event.
- [GenApi::IInteger & EventAcquisitionTransferEndFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.
- [GenApi::IInteger & EventAcquisitionError](#)  
*Description:* Returns the unique Identifier of the Acquisition Error type of Event.
- [GenApi::IInteger & EventAcquisitionErrorTimestamp](#)  
*Description:* Returns the Timestamp of the Acquisition Error Event.
- [GenApi::IInteger & EventAcquisitionErrorFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.
- [GenApi::IInteger & EventFrameTrigger](#)  
*Description:* Returns the unique Identifier of the FrameTrigger type of Event.
- [GenApi::IInteger & EventFrameTriggerTimestamp](#)  
*Description:* Returns the Timestamp of the FrameTrigger Event.
- [GenApi::IInteger & EventFrameTriggerFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.
- [GenApi::IInteger & EventFrameStart](#)  
*Description:* Returns the unique Identifier of the Frame Start type of Event.
- [GenApi::IInteger & EventFrameStartTimestamp](#)  
*Description:* Returns the Timestamp of the Frame Start Event.
- [GenApi::IInteger & EventFrameStartFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.
- [GenApi::IInteger & EventFrameEnd](#)  
*Description:* Returns the unique Identifier of the Frame End type of Event.
- [GenApi::IInteger & EventFrameEndTimestamp](#)  
*Description:* Returns the Timestamp of the Frame End Event.
- [GenApi::IInteger & EventFrameEndFrameID](#)  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.
- [GenApi::IInteger & EventFrameBurstStart](#)

*Description:* Returns the unique Identifier of the Frame Burst Start type of Event.

- `GenApi::IInteger & EventFrameBurstStartTimestamp`

*Description:* Returns the Timestamp of the Frame Burst Start Event.

- `GenApi::IInteger & EventFrameBurstStartFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.

- `GenApi::IInteger & EventFrameBurstEnd`

*Description:* Returns the unique Identifier of the Frame Burst End type of Event.

- `GenApi::IInteger & EventFrameBurstEndTimestamp`

*Description:* Returns the Timestamp of the Frame Burst End Event.

- `GenApi::IInteger & EventFrameBurstEndFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.

- `GenApi::IInteger & EventFrameTransferStart`

*Description:* Returns the unique Identifier of the Frame Transfer Start type of Event.

- `GenApi::IInteger & EventFrameTransferStartTimestamp`

*Description:* Returns the Timestamp of the Frame Transfer Start Event.

- `GenApi::IInteger & EventFrameTransferStartFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.

- `GenApi::IInteger & EventFrameTransferEnd`

*Description:* Returns the unique Identifier of the Frame Transfer End type of Event.

- `GenApi::IInteger & EventFrameTransferEndTimestamp`

*Description:* Returns the Timestamp of the Frame Transfer End Event.

- `GenApi::IInteger & EventFrameTransferEndFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.

- `GenApi::IInteger & EventExposureStart`

*Description:* Returns the unique Identifier of the Exposure Start type of Event.

- `GenApi::IInteger & EventExposureStartTimestamp`

*Description:* Returns the Timestamp of the Exposure Start Event.

- `GenApi::IInteger & EventExposureStartFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.

- `GenApi::IInteger & EventStream0TransferStart`

*Description:* Returns the unique Identifier of the Stream 0 Transfer Start type of Event.

- `GenApi::IInteger & EventStream0TransferStartTimestamp`

*Description:* Returns the Timestamp of the Stream 0 Transfer Start Event.

- `GenApi::IInteger & EventStream0TransferStartFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.

- `GenApi::IInteger & EventStream0TransferEnd`

*Description:* Returns the unique Identifier of the Stream 0 Transfer End type of Event.

- `GenApi::IInteger & EventStream0TransferEndTimestamp`

*Description:* Returns the Timestamp of the Stream 0 Transfer End Event.

- `GenApi::IInteger & EventStream0TransferEndFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.

- `GenApi::IInteger & EventStream0TransferPause`

*Description:* Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.

- `GenApi::IInteger & EventStream0TransferPauseTimestamp`

*Description:* Returns the Timestamp of the Stream 0 Transfer Pause Event.

- `GenApi::IInteger & EventStream0TransferPauseFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.

- `GenApi::IInteger & EventStream0TransferResume`

*Description:* Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.

- `GenApi::IInteger & EventStream0TransferResumeTimestamp`

*Description:* Returns the Timestamp of the Stream 0 Transfer Resume Event.

- **GenApi::IInteger & EventStream0TransferResumeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.
- **GenApi::IInteger & EventStream0TransferBlockStart**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.
- **GenApi::IInteger & EventStream0TransferBlockStartTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Block Start Event.
- **GenApi::IInteger & EventStream0TransferBlockStartFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.
- **GenApi::IInteger & EventStream0TransferBlockEnd**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.
- **GenApi::IInteger & EventStream0TransferBlockEndTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Block End Event.
- **GenApi::IInteger & EventStream0TransferBlockEndFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.
- **GenApi::IInteger & EventStream0TransferBlockTrigger**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.
- **GenApi::IInteger & EventStream0TransferBlockTriggerTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.
- **GenApi::IInteger & EventStream0TransferBlockTriggerFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.
- **GenApi::IInteger & EventStream0TransferBurstStart**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.
- **GenApi::IInteger & EventStream0TransferBurstStartTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Burst Start Event.
- **GenApi::IInteger & EventStream0TransferBurstStartFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.
- **GenApi::IInteger & EventStream0TransferBurstEnd**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.
- **GenApi::IInteger & EventStream0TransferBurstEndTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Burst End Event.
- **GenApi::IInteger & EventStream0TransferBurstEndFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.
- **GenApi::IInteger & EventStream0TransferOverflow**  
*Description:* Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.
- **GenApi::IInteger & EventStream0TransferOverflowTimestamp**  
*Description:* Returns the Timestamp of the Stream 0 Transfer Overflow Event.
- **GenApi::IInteger & EventStream0TransferOverflowFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.
- **GenApi::IInteger & EventSequencerSetChange**  
*Description:* Returns the unique Identifier of the Sequencer Set Change type of Event.
- **GenApi::IInteger & EventSequencerSetChangeTimestamp**  
*Description:* Returns the Timestamp of the Sequencer Set Change Event.
- **GenApi::IInteger & EventSequencerSetChangeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.
- **GenApi::IInteger & EventCounter0Start**

- `GenApi::IInteger & EventCounter0StartTimestamp`  
*Description: Returns the unique Identifier of the Counter 0 Start type of Event.*
- `GenApi::IInteger & EventCounter0StartFrameID`  
*Description: Returns the Timestamp of the Counter 0 Start Event.*
- `GenApi::IInteger & EventCounter0StartFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.*
- `GenApi::IInteger & EventCounter1Start`  
*Description: Returns the unique Identifier of the Counter 1 Start type of Event.*
- `GenApi::IInteger & EventCounter1StartTimestamp`  
*Description: Returns the Timestamp of the Counter 1 Start Event.*
- `GenApi::IInteger & EventCounter1StartFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.*
- `GenApi::IInteger & EventCounter0End`  
*Description: Returns the unique Identifier of the Counter 0 End type of Event.*
- `GenApi::IInteger & EventCounter0EndTimestamp`  
*Description: Returns the Timestamp of the Counter 0 End Event.*
- `GenApi::IInteger & EventCounter0EndFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.*
- `GenApi::IInteger & EventCounter1End`  
*Description: Returns the unique Identifier of the Counter 1 End type of Event.*
- `GenApi::IInteger & EventCounter1EndTimestamp`  
*Description: Returns the Timestamp of the Counter 1 End Event.*
- `GenApi::IInteger & EventCounter1EndFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.*
- `GenApi::IInteger & EventTimer0Start`  
*Description: Returns the unique Identifier of the Timer 0 Start type of Event.*
- `GenApi::IInteger & EventTimer0StartTimestamp`  
*Description: Returns the Timestamp of the Timer 0 Start Event.*
- `GenApi::IInteger & EventTimer0StartFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.*
- `GenApi::IInteger & EventTimer1Start`  
*Description: Returns the unique Identifier of the Timer 1 Start type of Event.*
- `GenApi::IInteger & EventTimer1StartTimestamp`  
*Description: Returns the Timestamp of the Timer 1 Start Event.*
- `GenApi::IInteger & EventTimer1StartFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.*
- `GenApi::IInteger & EventTimer0End`  
*Description: Returns the unique Identifier of the Timer 0 End type of Event.*
- `GenApi::IInteger & EventTimer0EndTimestamp`  
*Description: Returns the Timestamp of the Timer 0 End Event.*
- `GenApi::IInteger & EventTimer0EndFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.*
- `GenApi::IInteger & EventTimer1End`  
*Description: Returns the unique Identifier of the Timer 1 End type of Event.*
- `GenApi::IInteger & EventTimer1EndTimestamp`  
*Description: Returns the Timestamp of the Timer 1 End Event.*
- `GenApi::IInteger & EventTimer1EndFrameID`  
*Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.*
- `GenApi::IInteger & EventEncoder0Stopped`  
*Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.*
- `GenApi::IInteger & EventEncoder0StoppedTimestamp`  
*Description: Returns the Timestamp of the Encoder 0 Stopped Event.*

- **GenApi::IInteger & EventEncoder0StoppedFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.
- **GenApi::IInteger & EventEncoder1Stopped**  
*Description:* Returns the unique Identifier of the Encoder 1 Stopped type of Event.
- **GenApi::IInteger & EventEncoder1StoppedTimestamp**  
*Description:* Returns the Timestamp of the Encoder 1 Stopped Event.
- **GenApi::IInteger & EventEncoder1StoppedFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.
- **GenApi::IInteger & EventEncoder0Restarted**  
*Description:* Returns the unique Identifier of the Encoder 0 Restarted type of Event.
- **GenApi::IInteger & EventEncoder0RestartedTimestamp**  
*Description:* Returns the Timestamp of the Encoder 0 Restarted Event.
- **GenApi::IInteger & EventEncoder0RestartedFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.
- **GenApi::IInteger & EventEncoder1Restarted**  
*Description:* Returns the unique Identifier of the Encoder 1 Restarted type of Event.
- **GenApi::IInteger & EventEncoder1RestartedTimestamp**  
*Description:* Returns the Timestamp of the Encoder 1 Restarted Event.
- **GenApi::IInteger & EventEncoder1RestartedFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.
- **GenApi::IInteger & EventLine0RisingEdge**  
*Description:* Returns the unique Identifier of the Line 0 Rising Edge type of Event.
- **GenApi::IInteger & EventLine0RisingEdgeTimestamp**  
*Description:* Returns the Timestamp of the Line 0 Rising Edge Event.
- **GenApi::IInteger & EventLine0RisingEdgeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.
- **GenApi::IInteger & EventLine1RisingEdge**  
*Description:* Returns the unique Identifier of the Line 1 Rising Edge type of Event.
- **GenApi::IInteger & EventLine1RisingEdgeTimestamp**  
*Description:* Returns the Timestamp of the Line 1 Rising Edge Event.
- **GenApi::IInteger & EventLine1RisingEdgeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.
- **GenApi::IInteger & EventLine0FallingEdge**  
*Description:* Returns the unique Identifier of the Line 0 Falling Edge type of Event.
- **GenApi::IInteger & EventLine0FallingEdgeTimestamp**  
*Description:* Returns the Timestamp of the Line 0 Falling Edge Event.
- **GenApi::IInteger & EventLine0FallingEdgeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.
- **GenApi::IInteger & EventLine1FallingEdge**  
*Description:* Returns the unique Identifier of the Line 1 Falling Edge type of Event.
- **GenApi::IInteger & EventLine1FallingEdgeTimestamp**  
*Description:* Returns the Timestamp of the Line 1 Falling Edge Event.
- **GenApi::IInteger & EventLine1FallingEdgeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.
- **GenApi::IInteger & EventLine0AnyEdge**  
*Description:* Returns the unique Identifier of the Line 0 Any Edge type of Event.
- **GenApi::IInteger & EventLine0AnyEdgeTimestamp**  
*Description:* Returns the Timestamp of the Line 0 Any Edge Event.
- **GenApi::IInteger & EventLine0AnyEdgeFrameID**  
*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.
- **GenApi::IInteger & EventLine1AnyEdge**

- Description:* Returns the unique Identifier of the Line 1 Any Edge type of Event.

  - `GenApi::IInteger & EventLine1AnyEdgeTimestamp`

*Description:* Returns the Timestamp of the Line 1 Any Edge Event.
  - `GenApi::IInteger & EventLine1AnyEdgeFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.
  - `GenApi::IInteger & EventLinkTrigger0`

*Description:* Returns the unique Identifier of the Link Trigger 0 type of Event.
  - `GenApi::IInteger & EventLinkTrigger0Timestamp`

*Description:* Returns the Timestamp of the Link Trigger 0 Event.
  - `GenApi::IInteger & EventLinkTrigger0FrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.
  - `GenApi::IInteger & EventLinkTrigger1`

*Description:* Returns the unique Identifier of the Link Trigger 1 type of Event.
  - `GenApi::IInteger & EventLinkTrigger1Timestamp`

*Description:* Returns the Timestamp of the Link Trigger 1 Event.
  - `GenApi::IInteger & EventLinkTrigger1FrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.
  - `GenApi::IInteger & EventActionLate`

*Description:* Returns the unique Identifier of the Action Late type of Event.
  - `GenApi::IInteger & EventActionLateTimestamp`

*Description:* Returns the Timestamp of the Action Late Event.
  - `GenApi::IInteger & EventActionLateFrameID`

*Description:* Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.
  - `GenApi::IInteger & EventLinkSpeedChange`

*Description:* Returns the unique Identifier of the Link Speed Change type of Event.
  - `GenApi::IInteger & EventLinkSpeedChangeTimestamp`

*Description:* Returns the Timestamp of the Link Speed Change Event.
  - `GenApi::IInteger & 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::IInteger & 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::IInteger & 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::IInteger](#) & [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](#) & [Scan3dInvalidHeaderValue](#)

*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::IInteger](#) & [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::IInteger](#) & [ChunkPixelDynamicRangeMin](#)

*Description:* Returns the minimum value of dynamic range of the image included in the payload.
- [GenApi::IInteger](#) & [ChunkPixelDynamicRangeMax](#)

*Description:* Returns the maximum value of dynamic range of the image included in the payload.
- [GenApi::IInteger](#) & [ChunkTimestampLatchValue](#)

- Description:* Returns the last Timestamp latched with the *TimestampLatch* command.

  - **GenApi::IInteger & 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::IInteger & 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::IInteger & ChunkScanLineSelector**

*Description:* Index for vector representation of one chunk value per line in an image.
  - **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.
  - **GenApi::IEnumerationT< ChunkEncoderStatusEnums > & ChunkEncoderStatus**

*Description:* Returns the motion status of the selected encoder.
  - **GenApi::IEnumerationT< ChunkExposureTimeSelectorEnums > & ChunkExposureTimeSelector**

*Description:* Selects which exposure time is read by the *ChunkExposureTime* feature.
  - **GenApi::IInteger & ChunkLinePitch**

*Description:* Returns the *LinePitch* of the image included in the payload.
  - **GenApi::IEnumerationT< ChunkSourceIDEnums > & ChunkSourceID**

*Description:* Returns the identifier of *Source* that the image comes from.
  - **GenApi::IEnumerationT< ChunkRegionIDEnums > & ChunkRegionID**

*Description:* Returns the identifier of *Region* that the image comes from.
  - **GenApi::IInteger & ChunkTransferBlockID**

*Description:* Returns the unique identifier of the transfer block used to transport the payload.
  - **GenApi::IEnumerationT< ChunkTransferStreamIDEnums > & ChunkTransferStreamID**

*Description:* Returns identifier of the stream that generated this block.
  - **GenApi::IInteger & ChunkTransferQueueCurrentBlockCount**

*Description:* Returns the current number of blocks in the transfer queue.
  - **GenApi::IInteger & ChunkStreamChannelID**

*Description:* Returns identifier of the stream channel used to carry the block.
  - **GenApi::IEnumerationT< ChunkScan3dDistanceUnitEnums > & ChunkScan3dDistanceUnit**

*Description:* Returns the Distance Unit of the payload image.
  - **GenApi::IEnumerationT< ChunkScan3dOutputModeEnums > & ChunkScan3dOutputMode**

*Description:* Returns the Calibrated Mode of the payload image.
  - **GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums > & ChunkScan3dCoordinateSystem**

*Description:* Returns the Coordinate *System* of the image included in the payload.
  - **GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > & ChunkScan3dCoordinateSystemReference**

*Description:* Returns the Coordinate *System* Position of the image included in the payload.
  - **GenApi::IEnumerationT< 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::IBoolean & ChunkScan3dInvalidDataFlag**

- **GenApi::IFloat & ChunkScan3dInvalidDataValue**

Description: Returns if a specific non-valid data flag is used in the data stream.
- **GenApi::IFloat & ChunkScan3dAxisMin**

Description: Returns the Invalid Data Value used for the image included in the payload.
- **GenApi::IFloat & ChunkScan3dAxisMax**

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.
- **GenApi::IFloat & ChunkScan3dTransformValue**

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.
- **GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums > & ChunkScan3dCoordinateTransformSelector**

Description: Selector for transform values.
- **GenApi::IFloat & ChunkScan3dCoordinateReferenceValue**

Description: Returns the transform value.
- **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.
- **GenApi::IFloat & 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::IInteger & GevActiveLinkCount**

Description: Indicates the current number of active logical links.
- **GenApi::IBoolean & GevPAUSEFrameReception**

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.
- **GenApi::IBoolean & 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::IInteger & GevDiscoveryAckDelay**

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.
- **GenApi::IEnumerationT< GevGVCPExtendedStatusCodesSelectorEnums > & GevGVCPExtendedStatusCodesSelector**

Description: Selects the GigE Vision version to control extended status codes for.
- **GenApi::IBoolean & GevGVCPExtendedStatusCodes**

Description: Enables the generation of extended status codes.
- **GenApi::IInteger & GevPrimaryApplicationSwitchoverKey**

Description: Controls the key to use to authenticate primary application switchover requests.
- **GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums > & GevGVSPExtendedIDMode**

Description: Enables the extended IDs mode.
- **GenApi::IInteger & GevPrimaryApplicationSocket**

Description: Returns the UDP source port of the primary application.
- **GenApi::IInteger & GevPrimaryApplicationIPAddress**

Description: Returns the address of the primary application.
- **GenApi::IBoolean & GevSCCFGPacketResendDestination**

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.
- **GenApi::IBoolean & GevSCCFGAllInTransmission**

Description:

- Description:* Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

  - [GenApi::IInteger & GevSCZoneCount](#)

*Description:* Reports the number of zones per block transmitted on the selected stream channel.
  - [GenApi::IInteger & GevSCZoneDirectionAll](#)

*Description:* Reports the transmission direction of each zone transmitted on the selected stream channel.
  - [GenApi::IBoolean & GevSCZoneConfigurationLock](#)

*Description:* Controls whether the selected stream channel multi-zone configuration is locked.
  - [GenApi::IInteger & aPAUSEMACCtrlFramesTransmitted](#)

*Description:* Reports the number of transmitted PAUSE frames.
  - [GenApi::IInteger & 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::IInteger & 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::IInteger & CxpConnectionTestErrorCount](#)

*Description:* Reports the current connection error count for test packets received by the device on the connection selected by CxpConnectionSelector.
  - [GenApi::IInteger & CxpConnectionTestPacketCount](#)

*Description:* Reports the current count for test packets received 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::IInteger & ChunkInferenceFrameId](#)

*Description:* Returns the frame ID associated with the most recent inference result.
  - [GenApi::IInteger & 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

### 14.11.1 Detailed Description

The camera object class.

### 14.11.2 Constructor & Destructor Documentation

#### 14.11.2.1 ~Camera()

`~Camera ()`

#### 14.11.2.2 Camera()

`Camera () [protected]`

### 14.11.3 Member Function Documentation

#### 14.11.3.1 Init()

`void Init () [virtual]`

Implements [ICameraBase](#).

### 14.11.4 Member Data Documentation

#### 14.11.4.1 AasRoiEnable

`GenApi::IBoolean& AasRoiEnable`

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 14.11.4.2 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:

#### 14.11.4.3 AasRoiOffsetX

`GenApi::IInteger& AasRoiOffsetX`

Description:

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 14.11.4.4 AasRoiOffsetY

`GenApi::IInteger& AasRoiOffsetY`

Description:

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 14.11.4.5 AasRoiWidth

`GenApi::IInteger& AasRoiWidth`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

#### 14.11.4.6 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

#### 14.11.4.7 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

#### 14.11.4.8 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 MultiFrame and ignored if AcquisitionMode is Single.

Visibility:

#### 14.11.4.9 AcquisitionFrameCount

`GenApi::IInteger& AcquisitionFrameCount`

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

#### 14.11.4.10 AcquisitionFrameRate

`GenApi::IFloat& AcquisitionFrameRate`

Description: User controlled acquisition frame rate in Hertz Visibility:

#### 14.11.4.11 AcquisitionFrameRateEnable

`GenApi::IBoolean& AcquisitionFrameRateEnable`

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

#### 14.11.4.12 AcquisitionLineRate

`GenApi::IFloat& AcquisitionLineRate`

Description: Controls the rate (in Hertz) at which the Lines in a Frame are captured.

Visibility:

#### 14.11.4.13 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:

**14.11.4.14 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:

**14.11.4.15 AcquisitionStart**

```
GenApi::ICommand& AcquisitionStart
```

Description: This command starts the acquisition of images.

Visibility:

**14.11.4.16 AcquisitionStatus**

```
GenApi::IBoolean& AcquisitionStatus
```

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

**14.11.4.17 AcquisitionStatusSelector**

```
GenApi::IEnumeration<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector
```

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

**14.11.4.18 AcquisitionStop**

```
GenApi::ICommand& AcquisitionStop
```

Description: This command stops the acquisition of images.

Visibility:

**14.11.4.19 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

#### 14.11.4.20 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

#### 14.11.4.21 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

#### 14.11.4.22 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

#### 14.11.4.23 ActionSelector

`GenApi::IInteger& ActionSelector`

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

#### 14.11.4.24 ActionUnconditionalMode

`GenApi::IEnumeration<ActionUnconditionalModeEnums>& ActionUnconditionalMode`

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

Visibility: Guru

#### 14.11.4.25 AdaptiveCompressionEnable

`GenApi::IBoolean& AdaptiveCompressionEnable`

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

**14.11.4.26 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:

**14.11.4.27 aPAUSEMACCtrlFramesReceived**

```
GenApi::IInteger& aPAUSEMACCtrlFramesReceived
```

Description: Reports the number of received PAUSE frames.

Visibility: Guru

**14.11.4.28 aPAUSEMACCtrlFramesTransmitted**

```
GenApi::IInteger& aPAUSEMACCtrlFramesTransmitted
```

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

**14.11.4.29 AutoAlgorithmSelector**

```
GenApi::IEnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector
```

Description:

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

Visibility:

**14.11.4.30 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:

#### 14.11.4.31 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:

#### 14.11.4.32 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:

#### 14.11.4.33 AutoExposureExposureTimeLowerLimit

`GenApi::IFloat& AutoExposureExposureTimeLowerLimit`

Description:

The smallest exposure time that auto exposure can set.

Visibility:

#### 14.11.4.34 AutoExposureExposureTimeUpperLimit

`GenApi::IFloat& AutoExposureExposureTimeUpperLimit`

Description:

The largest exposure time that auto exposure can set.

Visibility:

**14.11.4.35 AutoExposureGainLowerLimit**

`GenApi::IFloat& AutoExposureGainLowerLimit`

Description:

The smallest gain that auto exposure can set.

Visibility:

**14.11.4.36 AutoExposureGainUpperLimit**

`GenApi::IFloat& AutoExposureGainUpperLimit`

Description:

The largest gain that auto exposure can set.

Visibility:

**14.11.4.37 AutoExposureGreyValueLowerLimit**

`GenApi::IFloat& AutoExposureGreyValueLowerLimit`

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

**14.11.4.38 AutoExposureGreyValueUpperLimit**

`GenApi::IFloat& AutoExposureGreyValueUpperLimit`

Description:

The highest value in percentage that the target mean may reach.

Visibility:

#### 14.11.4.39 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:

#### 14.11.4.40 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:

#### 14.11.4.41 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:

**14.11.4.42 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:

**14.11.4.43 BalanceRatio**

`GenApi::IFloat& BalanceRatio`

Description:

Controls the balance ratio of the selected color relative to green. Used for white balancing.

Visibility:

**14.11.4.44 BalanceRatioSelector**

`GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector`

Description:

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

Visibility:

**14.11.4.45 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:

#### 14.11.4.46 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:

#### 14.11.4.47 BalanceWhiteAutoLowerLimit

`GenApi::IFloat& BalanceWhiteAutoLowerLimit`

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

#### 14.11.4.48 BalanceWhiteAutoProfile

`GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile`

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

#### 14.11.4.49 BalanceWhiteAutoUpperLimit

`GenApi::IFloat& BalanceWhiteAutoUpperLimit`

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

#### 14.11.4.50 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:

#### 14.11.4.51 BinningHorizontalMode

```
GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode
```

Description: Visibility:

#### 14.11.4.52 BinningSelector

```
GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector
```

Description:

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

Visibility:

#### 14.11.4.53 BinningVertical

```
GenApi::IInteger& 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:

#### 14.11.4.54 BinningVerticalMode

```
GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode
```

Description: Visibility:

#### 14.11.4.55 BlackLevel

```
GenApi::IFloat& BlackLevel
```

Description:

Controls the offset of the video signal in percent.

Visibility:

#### 14.11.4.56 BlackLevelAuto

```
GenApi::IEnumerationT<BlackLevelAutoEnums>& BlackLevelAuto
```

Description: Controls the mode for automatic black level adjustment.

The exact algorithm used to implement this adjustment is device-specific. Visibility: Expert

#### 14.11.4.57 BlackLevelAutoBalance

```
GenApi::IEnumerationT<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

#### 14.11.4.58 BlackLevelClampingEnable

```
GenApi::IBoolean& BlackLevelClampingEnable
```

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

**14.11.4.59 BlackLevelRaw**

```
GenApi::IInteger& BlackLevelRaw
```

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

**14.11.4.60 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:

**14.11.4.61 ChunkBlackLevel**

```
GenApi::IFloat& ChunkBlackLevel
```

Description: Returns the black level used to capture the image.

Visibility:

**14.11.4.62 ChunkBlackLevelSelector**

```
GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector
```

Description: Selects which black level to retrieve Visibility:

**14.11.4.63 ChunkCounterSelector**

```
GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector
```

Description: Selects which counter to retrieve data from.

Visibility: Expert

#### 14.11.4.64 ChunkCounterValue

```
GenApi::IInteger& ChunkCounterValue
```

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

#### 14.11.4.65 ChunkCRC

```
GenApi::IInteger& ChunkCRC
```

Description: Returns the CRC of the image payload.

Visibility:

#### 14.11.4.66 ChunkEnable

```
GenApi::IBoolean& ChunkEnable
```

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

#### 14.11.4.67 ChunkEncoderSelector

```
GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector
```

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

#### 14.11.4.68 ChunkEncoderStatus

```
GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus
```

Description: Returns the motion status of the selected encoder.

Visibility: Expert

#### 14.11.4.69 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

**14.11.4.70 ChunkExposureEndLineStatusAll**

```
GenApi::IInteger& ChunkExposureEndLineStatusAll
```

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

**14.11.4.71 ChunkExposureTime**

```
GenApi::IFloat& ChunkExposureTime
```

Description: Returns the exposure time used to capture the image.

Visibility:

**14.11.4.72 ChunkExposureTimeSelector**

```
GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector
```

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

**14.11.4.73 ChunkFrameID**

```
GenApi::IInteger& ChunkFrameID
```

Description: Returns the image frame ID.

Visibility:

**14.11.4.74 ChunkGain**

```
GenApi::IFloat& ChunkGain
```

Description: Returns the gain used to capture the image.

Visibility:

**14.11.4.75 ChunkGainSelector**

```
GenApi::IEnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector
```

Description: Selects which gain to retrieve Visibility:

#### 14.11.4.76 ChunkHeight

`GenApi::IInteger& ChunkHeight`

Description: Returns the height of the image included in the payload.

Visibility:

#### 14.11.4.77 ChunkImage

`GenApi::IInteger& ChunkImage`

Description: Returns the image payload.

Visibility:

#### 14.11.4.78 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

#### 14.11.4.79 ChunkInferenceBoundingBoxResult

`GenApi::IRegister& ChunkInferenceBoundingBoxResult`

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

#### 14.11.4.80 ChunkInferenceConfidence

`GenApi::IFloat& ChunkInferenceConfidence`

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

#### 14.11.4.81 ChunkInferenceFrameId

`GenApi::IInteger& ChunkInferenceFrameId`

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

**14.11.4.82 ChunkInferenceResult**

```
GenApi::IInteger& ChunkInferenceResult
```

Description: Returns the chunk data inference result.

Visibility: Expert

**14.11.4.83 ChunkLinePitch**

```
GenApi::IInteger& ChunkLinePitch
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

**14.11.4.84 ChunkLineStatusAll**

```
GenApi::IInteger& ChunkLineStatusAll
```

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

**14.11.4.85 ChunkModeActive**

```
GenApi::IBoolean& ChunkModeActive
```

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

**14.11.4.86 ChunkOffsetX**

```
GenApi::IInteger& ChunkOffsetX
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

**14.11.4.87 ChunkOffsetY**

```
GenApi::IInteger& ChunkOffsetY
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

#### 14.11.4.88 ChunkPartSelector

`GenApi::IInteger& ChunkPartSelector`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

#### 14.11.4.89 ChunkPixelDynamicRangeMax

`GenApi::IInteger& ChunkPixelDynamicRangeMax`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

#### 14.11.4.90 ChunkPixelDynamicRangeMin

`GenApi::IInteger& ChunkPixelDynamicRangeMin`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

#### 14.11.4.91 ChunkPixelFormat

`GenApi::IEnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat`

Description: Format of the pixel provided by the camera Visibility:

#### 14.11.4.92 ChunkRegionID

`GenApi::IEnumerationT<ChunkRegionIDEnums>& ChunkRegionID`

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

#### 14.11.4.93 ChunkScan3dAxisMax

`GenApi::IFloat& ChunkScan3dAxisMax`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**14.11.4.94 ChunkScan3dAxisMin**

```
GenApi::IFloat& ChunkScan3dAxisMin
```

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**14.11.4.95 ChunkScan3dCoordinateOffset**

```
GenApi::IFloat& ChunkScan3dCoordinateOffset
```

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**14.11.4.96 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

**14.11.4.97 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

**14.11.4.98 ChunkScan3dCoordinateScale**

```
GenApi::IFloat& ChunkScan3dCoordinateScale
```

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

**14.11.4.99 ChunkScan3dCoordinateSelector**

```
GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector
```

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

#### 14.11.4.100 ChunkScan3dCoordinateSystem

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem`

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

Visibility: Expert

#### 14.11.4.101 ChunkScan3dCoordinateSystemReference

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinateSystemReference`

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

Visibility: Expert

#### 14.11.4.102 ChunkScan3dCoordinateTransformSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinateTransformSelector`

Description: Selector for transform values.

Visibility: Expert

#### 14.11.4.103 ChunkScan3dDistanceUnit

`GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit`

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

#### 14.11.4.104 ChunkScan3dInvalidDataFlag

`GenApi::IBoolean& ChunkScan3dInvalidDataFlag`

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

#### 14.11.4.105 ChunkScan3dInvalidDataValue

`GenApi::IFloat& ChunkScan3dInvalidDataValue`

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

**14.11.4.106 ChunkScan3dOutputMode**

```
GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode
```

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

**14.11.4.107 ChunkScan3dTransformValue**

```
GenApi::IFloat& ChunkScan3dTransformValue
```

Description: Returns the transform value.

Visibility: Expert

**14.11.4.108 ChunkScanLineSelector**

```
GenApi::IInteger& ChunkScanLineSelector
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

**14.11.4.109 ChunkSelector**

```
GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector
```

Description: Selects which chunk data to enable or disable.

Visibility:

**14.11.4.110 ChunkSequencerSetActive**

```
GenApi::IInteger& ChunkSequencerSetActive
```

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

**14.11.4.111 ChunkSerialData**

```
GenApi::IString& ChunkSerialData
```

Description: Returns the serial data that was received.

Visibility:

#### 14.11.4.112 ChunkSerialDataLength

`GenApi::IInteger& ChunkSerialDataLength`

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

#### 14.11.4.113 ChunkSerialReceiveOverflow

`GenApi::IBoolean& ChunkSerialReceiveOverflow`

Description: Returns the status of the chunk serial receive overflow.

Visibility:

#### 14.11.4.114 ChunkSourceID

`GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID`

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

#### 14.11.4.115 ChunkStreamChannelID

`GenApi::IInteger& ChunkStreamChannelID`

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

#### 14.11.4.116 ChunkTimerSelector

`GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector`

Description: Selects which Timer to retrieve data from.

Visibility: Expert

#### 14.11.4.117 ChunkTimerValue

`GenApi::IFloat& ChunkTimerValue`

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

**14.11.4.118 ChunkTimestamp**

```
GenApi::IInteger& ChunkTimestamp
```

Description: Returns the Timestamp of the image.

Visibility:

**14.11.4.119 ChunkTimestampLatchValue**

```
GenApi::IInteger& ChunkTimestampLatchValue
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

**14.11.4.120 ChunkTransferBlockID**

```
GenApi::IInteger& ChunkTransferBlockID
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

**14.11.4.121 ChunkTransferQueueCurrentBlockCount**

```
GenApi::IInteger& ChunkTransferQueueCurrentBlockCount
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

**14.11.4.122 ChunkTransferStreamID**

```
GenApi::IEnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID
```

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

**14.11.4.123 ChunkWidth**

```
GenApi::IInteger& ChunkWidth
```

Description: Returns the width of the image included in the payload.

Visibility:

#### 14.11.4.124 CIConfiguration

`GenApi::IEnumerationT<CIConfigurationEnums>& CIConfiguration`

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

#### 14.11.4.125 CLTimeSlotsCount

`GenApi::IEnumerationT<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

#### 14.11.4.126 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:

#### 14.11.4.127 ColorTransformationSelector

`GenApi::IEnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector`

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

#### 14.11.4.128 ColorTransformationValue

`GenApi::IFloat& ColorTransformationValue`

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

**14.11.4.129 ColorTransformationValueSelector**

```
GenApi::IEnumeration<ColorTransformationValueSelectorEnums>& ColorTransformationValueSelector
```

Description:

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

Visibility:

**14.11.4.130 CompressionRatio**

```
GenApi::IFloat& CompressionRatio
```

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

**14.11.4.131 CounterDelay**

```
GenApi::IInteger& CounterDelay
```

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

**14.11.4.132 CounterDuration**

```
GenApi::IInteger& CounterDuration
```

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

**14.11.4.133 CounterEventActivation**

```
GenApi::IEnumeration<CounterEventActivationEnums>& CounterEventActivation
```

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

#### 14.11.4.134 CounterEventSource

```
GenApi::IEnumerationT<CounterEventSourceEnums>& CounterEventSource
```

Description: Selects the event that will increment the counter Visibility:

#### 14.11.4.135 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

#### 14.11.4.136 CounterResetActivation

```
GenApi::IEnumerationT<CounterResetActivationEnums>& CounterResetActivation
```

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

#### 14.11.4.137 CounterResetSource

```
GenApi::IEnumerationT<CounterResetSourceEnums>& CounterResetSource
```

Description: Selects the signal that will be the source to reset the Counter.

Visibility:

#### 14.11.4.138 CounterSelector

```
GenApi::IEnumerationT<CounterSelectorEnums>& CounterSelector
```

Description: Selects which counter to configure Visibility:

#### 14.11.4.139 CounterStatus

```
GenApi::IEnumerationT<CounterStatusEnums>& CounterStatus
```

Description: Returns the current status of the Counter.

Visibility:

**14.11.4.140 CounterTriggerActivation**

```
GenApi::IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation
```

Description: Selects the activation mode of the trigger to start the Counter.

Visibility:

**14.11.4.141 CounterTriggerSource**

```
GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource
```

Description: Selects the source of the trigger to start the counter Visibility:

**14.11.4.142 CounterValue**

```
GenApi::IInteger& CounterValue
```

Description: Current counter value Visibility:

**14.11.4.143 CounterValueAtReset**

```
GenApi::IInteger& CounterValueAtReset
```

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

**14.11.4.144 CxpConnectionSelector**

```
GenApi::IInteger& CxpConnectionSelector
```

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

**14.11.4.145 CxpConnectionTestErrorCount**

```
GenApi::IInteger& CxpConnectionTestErrorCount
```

Description: Reports the current connection error count for test packets received by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

#### 14.11.4.146 CxpConnectionTestMode

`GenApi::IEnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode`

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

Visibility: Expert

#### 14.11.4.147 CxpConnectionTestPacketCount

`GenApi::IInteger& CxpConnectionTestPacketCount`

Description: Reports the current count for test packets received by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

#### 14.11.4.148 CxpLinkConfiguration

`GenApi::IEnumerationT<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

#### 14.11.4.149 CxpLinkConfigurationPreferred

`GenApi::IEnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred`

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

Visibility: Expert

#### 14.11.4.150 CxpLinkConfigurationStatus

`GenApi::IEnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus`

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

Visibility: Beginner

#### 14.11.4.151 CxpPoCxpAuto

`GenApi:: ICommand& CxpPoCxpAuto`

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

**14.11.4.152 CxpPoCxpStatus**

```
GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus
```

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

Visibility: Expert

**14.11.4.153 CxpPoCxpTripReset**

```
GenApi:: ICommand& CxpPoCxpTripReset
```

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

**14.11.4.154 CxpPoCxpTurnOff**

```
GenApi:: ICommand& CxpPoCxpTurnOff
```

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

**14.11.4.155 DecimationHorizontal**

```
GenApi::I 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:

**14.11.4.156 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:

#### 14.11.4.157 DecimationSelector

```
GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector
```

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

Visibility:

#### 14.11.4.158 DecimationVertical

```
GenApi::IInteger& 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:

#### 14.11.4.159 DecimationVerticalMode

```
GenApi::IEnumerationT<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:

#### 14.11.4.160 DefectCorrectionMode

```
GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode
```

Description: Controls the method used for replacing defective pixels.

Visibility:

**14.11.4.161 DefectCorrectStaticEnable**

```
GenApi::IBoolean& DefectCorrectStaticEnable
```

Description: Enables/Disables table-based defective pixel correction.

Visibility:

**14.11.4.162 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:

**14.11.4.163 DefectTableCoordinateX**

```
GenApi::IInteger& 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:

**14.11.4.164 DefectTableCoordinateY**

```
GenApi::IInteger& 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:

#### 14.11.4.165 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:

#### 14.11.4.166 DefectTableIndex

`GenApi:: IInteger& DefectTableIndex`

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

#### 14.11.4.167 DefectTablePixelCount

`GenApi:: IInteger& DefectTablePixelCount`

Description:

The number of defective pixel locations in the current table.

Visibility:

#### 14.11.4.168 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:

**14.11.4.169 Deinterlacing**

```
GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing
```

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

**14.11.4.170 DeviceCharacterSet**

```
GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet
```

Description:

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

Visibility:

**14.11.4.171 DeviceClockFrequency**

```
GenApi::IFloat& DeviceClockFrequency
```

Description: Returns the frequency of the selected Clock.

Visibility: Expert

**14.11.4.172 DeviceClockSelector**

```
GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector
```

Description: Selects the clock frequency to access from the device.

Visibility: Expert

**14.11.4.173 DeviceConnectionSelector**

```
GenApi::IInteger& DeviceConnectionSelector
```

Description: Selects which Connection of the device to control.

Visibility: Beginner

**14.11.4.174 DeviceConnectionSpeed**

`GenApi::IInteger& DeviceConnectionSpeed`

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

**14.11.4.175 DeviceConnectionStatus**

`GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus`

Description: Indicates the status of the specified Connection.

Visibility: Expert

**14.11.4.176 DeviceEventChannelCount**

`GenApi::IInteger& DeviceEventChannelCount`

Description:

Indicates the number of event channels supported by the device.

Visibility:

**14.11.4.177 DeviceFamilyName**

`GenApi::IString& DeviceFamilyName`

Description: Identifier of the product family of the device.

Visibility: Beginner

**14.11.4.178 DeviceFeaturePersistenceEnd**

`GenApi:: ICommand& DeviceFeaturePersistenceEnd`

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

**14.11.4.179 DeviceFeaturePersistenceStart**

`GenApi:: ICommand& DeviceFeaturePersistenceStart`

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

**14.11.4.180 DeviceFirmwareVersion**

`GenApi:: IString& DeviceFirmwareVersion`

Description: Version of the firmware on the device.

Visibility:

**14.11.4.181 DeviceGenCPVersionMajor**

`GenApi:: IInteger& DeviceGenCPVersionMajor`

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

**14.11.4.182 DeviceGenCPVersionMinor**

`GenApi:: IInteger& DeviceGenCPVersionMinor`

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

**14.11.4.183 DeviceID**

`GenApi:: IString& DeviceID`

Description: Device identifier (serial number).

Visibility:

**14.11.4.184 DeviceIndicatorMode**

`GenApi:: IEnumeration<DeviceIndicatorModeEnums>& DeviceIndicatorMode`

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

Visibility:

#### 14.11.4.185 DeviceLinkBandwidthReserve

`GenApi::IFloat& DeviceLinkBandwidthReserve`

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

#### 14.11.4.186 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

#### 14.11.4.187 DeviceLinkConnectionCount

`GenApi::IInteger& DeviceLinkConnectionCount`

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

#### 14.11.4.188 DeviceLinkCurrentThroughput

`GenApi::IInteger& DeviceLinkCurrentThroughput`

Description: Current bandwidth of streamed data.

Visibility:

#### 14.11.4.189 DeviceLinkHeartbeatMode

`GenApi::IEnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode`

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

**14.11.4.190 DeviceLinkHeartbeatTimeout**

`GenApi::IFloat& DeviceLinkHeartbeatTimeout`

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

**14.11.4.191 DeviceLinkSelector**

`GenApi::IInteger& DeviceLinkSelector`

Description: Selects which Link of the device to control.

Visibility: Beginner

**14.11.4.192 DeviceLinkSpeed**

`GenApi::IInteger& DeviceLinkSpeed`

Description:

Indicates the speed of transmission negotiated on the specified Link. (Bps)

Visibility:

**14.11.4.193 DeviceLinkThroughputLimit**

`GenApi::IInteger& 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:

**14.11.4.194 DeviceLinkThroughputLimitMode**

`GenApi::IEnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode`

Description: 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. Visibility: Expert

#### 14.11.4.195 DeviceManifestEntrySelector

`GenApi::IInteger& DeviceManifestEntrySelector`

Description: Selects the manifest entry to reference.

Visibility: Guru

#### 14.11.4.196 DeviceManifestPrimaryURL

`GenApi::IString& DeviceManifestPrimaryURL`

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

#### 14.11.4.197 DeviceManifestSchemaMajorVersion

`GenApi::IInteger& DeviceManifestSchemaMajorVersion`

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

#### 14.11.4.198 DeviceManifestSchemaMinorVersion

`GenApi::IInteger& DeviceManifestSchemaMinorVersion`

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

#### 14.11.4.199 DeviceManifestSecondaryURL

`GenApi::IString& DeviceManifestSecondaryURL`

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

#### 14.11.4.200 DeviceManifestXMLMajorVersion

`GenApi::IInteger& DeviceManifestXMLMajorVersion`

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**14.11.4.201 DeviceManifestXMLMinorVersion**

`GenApi::IInteger& DeviceManifestXMLMinorVersion`

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**14.11.4.202 DeviceManifestXMLSubMinorVersion**

`GenApi::IInteger& DeviceManifestXMLSubMinorVersion`

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

**14.11.4.203 DeviceManufacturerInfo**

`GenApi::IString& DeviceManufacturerInfo`

Description: Manufacturer information about the device.

Visibility:

**14.11.4.204 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:

**14.11.4.205 DeviceModelName**

`GenApi::IString& DeviceModelName`

Description: Model of the device.

Visibility:

#### 14.11.4.206 DevicePowerSupplySelector

`GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector`

Description:

Selects the power supply source to control or read.

Visibility:

#### 14.11.4.207 DeviceRegistersCheck

`GenApi:: ICommand& DeviceRegistersCheck`

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

#### 14.11.4.208 DeviceRegistersEndianness

`GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness`

Description: Endianess of the registers of the device.

Visibility:

#### 14.11.4.209 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

#### 14.11.4.210 DeviceRegistersStreamingStart

`GenApi:: ICommand& DeviceRegistersStreamingStart`

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

**14.11.4.211 DeviceRegistersValid**

```
GenApi::IBoolean& DeviceRegistersValid
```

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

**14.11.4.212 DeviceReset**

```
GenApi:: ICommand& DeviceReset
```

Description: This is a command that immediately resets and reboots the device.

Visibility:

**14.11.4.213 DeviceScanType**

```
GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType
```

Description: Scan type of the sensor of the device.

Visibility:

**14.11.4.214 DeviceSerialNumber**

```
GenApi::IString& DeviceSerialNumber
```

Description:

Device's serial number. This string is a unique identifier of the device.

Visibility:

**14.11.4.215 DeviceSerialPortBaudRate**

```
GenApi::IEnumerationT<DeviceSerialPortBaudRateEnum>& DeviceSerialPortBaudRate
```

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

Visibility: Expert

#### 14.11.4.216 DeviceSerialPortSelector

```
GenApi::IEnumeration<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility: Expert

#### 14.11.4.217 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

#### 14.11.4.218 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

#### 14.11.4.219 DeviceSFNCVersionSubMinor

```
GenApi::IInteger& DeviceSFNCVersionSubMinor
```

Description: Sub minor version of Standard Features Naming Convention that was used to create the device's GenICam XML.

Visibility: Beginner

#### 14.11.4.220 DeviceStreamChannelCount

```
GenApi::IInteger& DeviceStreamChannelCount
```

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

**14.11.4.221 DeviceStreamChannelEndianness**

```
GenApi::IEnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness
```

Description: Endianess of multi-byte pixel data for this stream.

Visibility: Guru

**14.11.4.222 DeviceStreamChannelLink**

```
GenApi::IInteger& DeviceStreamChannelLink
```

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

**14.11.4.223 DeviceStreamChannelPacketSize**

```
GenApi::IInteger& 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

**14.11.4.224 DeviceStreamChannelSelector**

```
GenApi::IInteger& DeviceStreamChannelSelector
```

Description: Selects the stream channel to control.

Visibility: Expert

**14.11.4.225 DeviceStreamChannelType**

```
GenApi::IEnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType
```

Description: Reports the type of the stream channel.

Visibility: Guru

**14.11.4.226 DeviceTapGeometry**

```
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.

Visibility: Expert

#### 14.11.4.227 DeviceTemperature

`GenApi::IFloat& DeviceTemperature`

Description: Device temperature in degrees Celsius (C).

Visibility:

#### 14.11.4.228 DeviceTemperatureSelector

`GenApi::IEnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector`

Description:

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

Visibility:

#### 14.11.4.229 DeviceTLType

`GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType`

Description: Transport Layer type of the device.

Visibility:

#### 14.11.4.230 DeviceTLVersionMajor

`GenApi::IInteger& DeviceTLVersionMajor`

Description:

Major version of the Transport Layer of the device.

Visibility:

#### 14.11.4.231 DeviceTLVersionMinor

`GenApi::IInteger& DeviceTLVersionMinor`

Description:

Minor version of the Transport Layer of the device.

Visibility:

**14.11.4.232 DeviceTLVersionSubMinor**

```
GenApi::IInteger& DeviceTLVersionSubMinor
```

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

**14.11.4.233 DeviceType**

```
GenApi::IEnumerationT<DeviceTypeEnums>& DeviceType
```

Description: Returns the device type.

Visibility: Guru

**14.11.4.234 DeviceUptime**

```
GenApi::IInteger& DeviceUptime
```

Description: Total time since the device was powered up in seconds.

Visibility:

**14.11.4.235 DeviceUserID**

```
GenApi::IString& DeviceUserID
```

Description: User-programmable device identifier.

Visibility:

**14.11.4.236 DeviceVendorName**

```
GenApi::IString& DeviceVendorName
```

Description: Name of the manufacturer of the device.

Visibility:

**14.11.4.237 DeviceVersion**

```
GenApi::IString& DeviceVersion
```

Description: Version of the device.

Visibility:

#### 14.11.4.238 EncoderDivider

```
GenApi::IInteger& EncoderDivider
```

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

#### 14.11.4.239 EncoderMode

```
GenApi::IEnumerationT<EncoderModeEnums>& EncoderMode
```

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

Visibility: Expert

#### 14.11.4.240 EncoderOutputMode

```
GenApi::IEnumerationT<EncoderOutputModeEnums>& EncoderOutputMode
```

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

Visibility: Expert

#### 14.11.4.241 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

#### 14.11.4.242 EncoderResetActivation

```
GenApi::IEnumerationT<EncoderResetActivationEnums>& EncoderResetActivation
```

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

#### 14.11.4.243 EncoderResetSource

```
GenApi::IEnumerationT<EncoderResetSourceEnums>& EncoderResetSource
```

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

**14.11.4.244 EncoderSelector**

```
GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector
```

Description: Selects which Encoder to configure.

Visibility: Expert

**14.11.4.245 EncoderSourceA**

```
GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA
```

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

Visibility: Expert

**14.11.4.246 EncoderSourceB**

```
GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB
```

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

Visibility: Expert

**14.11.4.247 EncoderStatus**

```
GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus
```

Description: Returns the motion status of the encoder.

Visibility: Expert

**14.11.4.248 EncoderTimeout**

```
GenApi::IFloat& EncoderTimeout
```

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

**14.11.4.249 EncoderValue**

```
GenApi::IInteger& EncoderValue
```

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

**14.11.4.250 EncoderValueAtReset**

```
GenApi::IInteger& EncoderValueAtReset
```

Description: Reads the value of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

**14.11.4.251 EnumerationCount**

```
GenApi::IInteger& EnumerationCount
```

Description: Number of enumerations since uptime.

Visibility:

**14.11.4.252 EventAcquisitionEnd**

```
GenApi::IInteger& EventAcquisitionEnd
```

Description: Returns the unique Identifier of the Acquisition End type of Event.

Visibility: Expert

**14.11.4.253 EventAcquisitionEndFrameID**

```
GenApi::IInteger& EventAcquisitionEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End Event.

Visibility: Expert

**14.11.4.254 EventAcquisitionEndTimestamp**

```
GenApi::IInteger& EventAcquisitionEndTimestamp
```

Description: Returns the Timestamp of the Acquisition End Event.

Visibility: Expert

**14.11.4.255 EventAcquisitionError**

```
GenApi::IInteger& EventAcquisitionError
```

Description: Returns the unique Identifier of the Acquisition Error type of Event.

Visibility: Expert

**14.11.4.256 EventAcquisitionErrorFrameID**

```
GenApi::IInteger& EventAcquisitionErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error Event.

Visibility: Expert

**14.11.4.257 EventAcquisitionErrorTimestamp**

```
GenApi::IInteger& EventAcquisitionErrorTimestamp
```

Description: Returns the Timestamp of the Acquisition Error Event.

Visibility: Expert

**14.11.4.258 EventAcquisitionStart**

```
GenApi::IInteger& EventAcquisitionStart
```

Description: Returns the unique Identifier of the Acquisition Start type of Event.

Visibility: Expert

**14.11.4.259 EventAcquisitionStartFrameID**

```
GenApi::IInteger& EventAcquisitionStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start Event.

Visibility: Expert

**14.11.4.260 EventAcquisitionStartTimestamp**

```
GenApi::IInteger& EventAcquisitionStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Start Event.

Visibility: Expert

**14.11.4.261 EventAcquisitionTransferEnd**

```
GenApi::IInteger& EventAcquisitionTransferEnd
```

Description: Returns the unique Identifier of the Acquisition Transfer End type of Event.

Visibility: Expert

**14.11.4.262 EventAcquisitionTransferEndFrameID**

```
GenApi::IInteger& EventAcquisitionTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End Event.

Visibility: Expert

**14.11.4.263 EventAcquisitionTransferEndTimestamp**

```
GenApi::IInteger& EventAcquisitionTransferEndTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer End Event.

Visibility: Expert

**14.11.4.264 EventAcquisitionTransferStart**

```
GenApi::IInteger& EventAcquisitionTransferStart
```

Description: Returns the unique Identifier of the Acquisition Transfer Start type of Event.

Visibility: Expert

**14.11.4.265 EventAcquisitionTransferStartFrameID**

```
GenApi::IInteger& EventAcquisitionTransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start Event.

Visibility: Expert

**14.11.4.266 EventAcquisitionTransferStartTimestamp**

```
GenApi::IInteger& EventAcquisitionTransferStartTimestamp
```

Description: Returns the Timestamp of the Acquisition Transfer Start Event.

Visibility: Expert

**14.11.4.267 EventAcquisitionTrigger**

```
GenApi::IInteger& EventAcquisitionTrigger
```

Description: Returns the unique Identifier of the Acquisition Trigger type of Event.

Visibility: Expert

**14.11.4.268 EventAcquisitionTriggerFrameID**

```
GenApi::IInteger& EventAcquisitionTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger Event.

Visibility: Expert

**14.11.4.269 EventAcquisitionTriggerTimestamp**

```
GenApi::IInteger& EventAcquisitionTriggerTimestamp
```

Description: Returns the Timestamp of the Acquisition Trigger Event.

Visibility: Expert

**14.11.4.270 EventActionLate**

```
GenApi::IInteger& EventActionLate
```

Description: Returns the unique Identifier of the Action Late type of Event.

Visibility: Expert

**14.11.4.271 EventActionLateFrameID**

```
GenApi::IInteger& EventActionLateFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late Event.

Visibility: Expert

**14.11.4.272 EventActionLateTimestamp**

```
GenApi::IInteger& EventActionLateTimestamp
```

Description: Returns the Timestamp of the Action Late Event.

Visibility: Expert

**14.11.4.273 EventCounter0End**

```
GenApi::IInteger& EventCounter0End
```

Description: Returns the unique Identifier of the Counter 0 End type of Event.

Visibility: Expert

**14.11.4.274 EventCounter0EndFrameID**

```
GenApi::IInteger& EventCounter0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End Event.

Visibility: Expert

**14.11.4.275 EventCounter0EndTimestamp**

```
GenApi::IInteger& EventCounter0EndTimestamp
```

Description: Returns the Timestamp of the Counter 0 End Event.

Visibility: Expert

**14.11.4.276 EventCounter0Start**

```
GenApi::IInteger& EventCounter0Start
```

Description: Returns the unique Identifier of the Counter 0 Start type of Event.

Visibility: Expert

**14.11.4.277 EventCounter0StartFrameID**

```
GenApi::IInteger& EventCounter0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.

Visibility: Expert

**14.11.4.278 EventCounter0StartTimestamp**

```
GenApi::IInteger& EventCounter0StartTimestamp
```

Description: Returns the Timestamp of the Counter 0 Start Event.

Visibility: Expert

**14.11.4.279 EventCounter1End**

```
GenApi::IInteger& EventCounter1End
```

Description: Returns the unique Identifier of the Counter 1 End type of Event.

Visibility: Expert

**14.11.4.280 EventCounter1EndFrameID**

```
GenApi::IInteger& EventCounter1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End Event.

Visibility: Expert

**14.11.4.281 EventCounter1EndTimestamp**

```
GenApi::IInteger& EventCounter1EndTimestamp
```

Description: Returns the Timestamp of the Counter 1 End Event.

Visibility: Expert

**14.11.4.282 EventCounter1Start**

```
GenApi::IInteger& EventCounter1Start
```

Description: Returns the unique Identifier of the Counter 1 Start type of Event.

Visibility: Expert

**14.11.4.283 EventCounter1StartFrameID**

```
GenApi::IInteger& EventCounter1StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start Event.

Visibility: Expert

**14.11.4.284 EventCounter1StartTimestamp**

```
GenApi::IInteger& EventCounter1StartTimestamp
```

Description: Returns the Timestamp of the Counter 1 Start Event.

Visibility: Expert

**14.11.4.285 EventEncoder0Restarted**

```
GenApi::IInteger& EventEncoder0Restarted
```

Description: Returns the unique Identifier of the Encoder 0 Restarted type of Event.

Visibility: Expert

**14.11.4.286 EventEncoder0RestartedFrameID**

```
GenApi::IInteger& EventEncoder0RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted Event.

Visibility: Expert

**14.11.4.287 EventEncoder0RestartedTimestamp**

```
GenApi::IInteger& EventEncoder0RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Restarted Event.

Visibility: Expert

**14.11.4.288 EventEncoder0Stopped**

```
GenApi::IInteger& EventEncoder0Stopped
```

Description: Returns the unique Identifier of the Encoder 0 Stopped type of Event.

Visibility: Expert

**14.11.4.289 EventEncoder0StoppedFrameID**

```
GenApi::IInteger& EventEncoder0StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped Event.

Visibility: Expert

**14.11.4.290 EventEncoder0StoppedTimestamp**

```
GenApi::IInteger& EventEncoder0StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 0 Stopped Event.

Visibility: Expert

**14.11.4.291 EventEncoder1Restarted**

```
GenApi::IInteger& EventEncoder1Restarted
```

Description: Returns the unique Identifier of the Encoder 1 Restarted type of Event.

Visibility: Expert

**14.11.4.292 EventEncoder1RestartedFrameID**

```
GenApi::IInteger& EventEncoder1RestartedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted Event.

Visibility: Expert

**14.11.4.293 EventEncoder1RestartedTimestamp**

```
GenApi::IInteger& EventEncoder1RestartedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Restarted Event.

Visibility: Expert

**14.11.4.294 EventEncoder1Stopped**

```
GenApi::IInteger& EventEncoder1Stopped
```

Description: Returns the unique Identifier of the Encoder 1 Stopped type of Event.

Visibility: Expert

**14.11.4.295 EventEncoder1StoppedFrameID**

```
GenApi::IInteger& EventEncoder1StoppedFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped Event.

Visibility: Expert

**14.11.4.296 EventEncoder1StoppedTimestamp**

```
GenApi::IInteger& EventEncoder1StoppedTimestamp
```

Description: Returns the Timestamp of the Encoder 1 Stopped Event.

Visibility: Expert

**14.11.4.297 EventError**

```
GenApi::IInteger& EventError
```

Description: Returns the unique identifier of the Error type of Event.

Visibility:

**14.11.4.298 EventErrorCode**

```
GenApi::IInteger& EventErrorCode
```

Description: Returns the error code for the error that happened Visibility:

**14.11.4.299 EventErrorFrameID**

```
GenApi::IInteger& EventErrorFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Error Event.

Visibility:

**14.11.4.300 EventErrorTimestamp**

```
GenApi::IInteger& EventErrorTimestamp
```

Description: Returns the Timestamp of the Error Event.

Visibility:

**14.11.4.301 EventExposureEnd**

```
GenApi::IInteger& EventExposureEnd
```

Description: Returns the unique identifier of the Exposure End type of Event.

Visibility:

**14.11.4.302 EventExposureEndFrameID**

```
GenApi::IInteger& EventExposureEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure End Event.

Visibility:

**14.11.4.303 EventExposureEndTimestamp**

```
GenApi::IInteger& EventExposureEndTimestamp
```

Description: Returns the Timestamp of the Exposure End Event.

Visibility:

**14.11.4.304 EventExposureStart**

```
GenApi::IInteger& EventExposureStart
```

Description: Returns the unique Identifier of the Exposure Start type of Event.

Visibility: Expert

**14.11.4.305 EventExposureStartFrameID**

```
GenApi::IInteger& EventExposureStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start Event.

Visibility: Expert

**14.11.4.306 EventExposureStartTimestamp**

```
GenApi::IInteger& EventExposureStartTimestamp
```

Description: Returns the Timestamp of the Exposure Start Event.

Visibility: Expert

**14.11.4.307 EventFrameBurstEnd**

```
GenApi::IInteger& EventFrameBurstEnd
```

Description: Returns the unique Identifier of the Frame Burst End type of Event.

Visibility: Expert

**14.11.4.308 EventFrameBurstEndFrameID**

```
GenApi::IInteger& EventFrameBurstEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End Event.

Visibility: Expert

**14.11.4.309 EventFrameBurstEndTimestamp**

```
GenApi::IInteger& EventFrameBurstEndTimestamp
```

Description: Returns the Timestamp of the Frame Burst End Event.

Visibility: Expert

#### 14.11.4.310 EventFrameBurstStart

`GenApi::IInteger& EventFrameBurstStart`

Description: Returns the unique Identifier of the Frame Burst Start type of Event.

Visibility: Expert

#### 14.11.4.311 EventFrameBurstStartFrameID

`GenApi::IInteger& EventFrameBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start Event.

Visibility: Expert

#### 14.11.4.312 EventFrameBurstStartTimestamp

`GenApi::IInteger& EventFrameBurstStartTimestamp`

Description: Returns the Timestamp of the Frame Burst Start Event.

Visibility: Expert

#### 14.11.4.313 EventFrameEnd

`GenApi::IInteger& EventFrameEnd`

Description: Returns the unique Identifier of the Frame End type of Event.

Visibility: Expert

#### 14.11.4.314 EventFrameEndFrameID

`GenApi::IInteger& EventFrameEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End Event.

Visibility: Expert

#### 14.11.4.315 EventFrameEndTimestamp

`GenApi::IInteger& EventFrameEndTimestamp`

Description: Returns the Timestamp of the Frame End Event.

Visibility: Expert

**14.11.4.316 EventFrameStart**

```
GenApi::IInteger& EventFrameStart
```

Description: Returns the unique Identifier of the Frame Start type of Event.

Visibility: Expert

**14.11.4.317 EventFrameStartFrameID**

```
GenApi::IInteger& EventFrameStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start Event.

Visibility: Expert

**14.11.4.318 EventFrameStartTimestamp**

```
GenApi::IInteger& EventFrameStartTimestamp
```

Description: Returns the Timestamp of the Frame Start Event.

Visibility: Expert

**14.11.4.319 EventFrameTransferEnd**

```
GenApi::IInteger& EventFrameTransferEnd
```

Description: Returns the unique Identifier of the Frame Transfer End type of Event.

Visibility: Expert

**14.11.4.320 EventFrameTransferEndFrameID**

```
GenApi::IInteger& EventFrameTransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End Event.

Visibility: Expert

**14.11.4.321 EventFrameTransferEndTimestamp**

```
GenApi::IInteger& EventFrameTransferEndTimestamp
```

Description: Returns the Timestamp of the Frame Transfer End Event.

Visibility: Expert

#### 14.11.4.322 EventFrameTransferStart

`GenApi::IInteger& EventFrameTransferStart`

Description: Returns the unique Identifier of the Frame Transfer Start type of Event.

Visibility: Expert

#### 14.11.4.323 EventFrameTransferStartFrameID

`GenApi::IInteger& EventFrameTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start Event.

Visibility: Expert

#### 14.11.4.324 EventFrameTransferStartTimestamp

`GenApi::IInteger& EventFrameTransferStartTimestamp`

Description: Returns the Timestamp of the Frame Transfer Start Event.

Visibility: Expert

#### 14.11.4.325 EventFrameTrigger

`GenApi::IInteger& 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

#### 14.11.4.326 EventFrameTriggerFrameID

`GenApi::IInteger& EventFrameTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger Event.

Visibility: Expert

#### 14.11.4.327 EventFrameTriggerTimestamp

`GenApi::IInteger& EventFrameTriggerTimestamp`

Description: Returns the Timestamp of the FrameTrigger Event.

It can be used to determine precisely when the event occurred. Visibility: Expert

**14.11.4.328 EventLine0AnyEdge**

`GenApi::IInteger& EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of Event.

Visibility: Expert

**14.11.4.329 EventLine0AnyEdgeFrameID**

`GenApi::IInteger& EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge Event.

Visibility: Expert

**14.11.4.330 EventLine0AnyEdgeTimestamp**

`GenApi::IInteger& EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge Event.

Visibility: Expert

**14.11.4.331 EventLine0FallingEdge**

`GenApi::IInteger& EventLine0FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of Event.

Visibility: Expert

**14.11.4.332 EventLine0FallingEdgeFrameID**

`GenApi::IInteger& EventLine0FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge Event.

Visibility: Expert

**14.11.4.333 EventLine0FallingEdgeTimestamp**

`GenApi::IInteger& EventLine0FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Falling Edge Event.

Visibility: Expert

**14.11.4.334 EventLine0RisingEdge**

```
GenApi::IInteger& EventLine0RisingEdge
```

Description: Returns the unique Identifier of the Line 0 Rising Edge type of Event.

Visibility: Expert

**14.11.4.335 EventLine0RisingEdgeFrameID**

```
GenApi::IInteger& EventLine0RisingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge Event.

Visibility: Expert

**14.11.4.336 EventLine0RisingEdgeTimestamp**

```
GenApi::IInteger& EventLine0RisingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 0 Rising Edge Event.

Visibility: Expert

**14.11.4.337 EventLine1AnyEdge**

```
GenApi::IInteger& EventLine1AnyEdge
```

Description: Returns the unique Identifier of the Line 1 Any Edge type of Event.

Visibility: Expert

**14.11.4.338 EventLine1AnyEdgeFrameID**

```
GenApi::IInteger& EventLine1AnyEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge Event.

Visibility: Expert

**14.11.4.339 EventLine1AnyEdgeTimestamp**

```
GenApi::IInteger& EventLine1AnyEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Any Edge Event.

Visibility: Expert

**14.11.4.340 EventLine1FallingEdge**

```
GenApi::IInteger& EventLine1FallingEdge
```

Description: Returns the unique Identifier of the Line 1 Falling Edge type of Event.

Visibility: Expert

**14.11.4.341 EventLine1FallingEdgeFrameID**

```
GenApi::IInteger& EventLine1FallingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge Event.

Visibility: Expert

**14.11.4.342 EventLine1FallingEdgeTimestamp**

```
GenApi::IInteger& EventLine1FallingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Falling Edge Event.

Visibility: Expert

**14.11.4.343 EventLine1RisingEdge**

```
GenApi::IInteger& EventLine1RisingEdge
```

Description: Returns the unique Identifier of the Line 1 Rising Edge type of Event.

Visibility: Expert

**14.11.4.344 EventLine1RisingEdgeFrameID**

```
GenApi::IInteger& EventLine1RisingEdgeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge Event.

Visibility: Expert

**14.11.4.345 EventLine1RisingEdgeTimestamp**

```
GenApi::IInteger& EventLine1RisingEdgeTimestamp
```

Description: Returns the Timestamp of the Line 1 Rising Edge Event.

Visibility: Expert

**14.11.4.346 EventLinkSpeedChange**

```
GenApi::IInteger& EventLinkSpeedChange
```

Description: Returns the unique Identifier of the Link Speed Change type of Event.

Visibility: Expert

**14.11.4.347 EventLinkSpeedChangeFrameID**

```
GenApi::IInteger& EventLinkSpeedChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change Event.

Visibility: Expert

**14.11.4.348 EventLinkSpeedChangeTimestamp**

```
GenApi::IInteger& EventLinkSpeedChangeTimestamp
```

Description: Returns the Timestamp of the Link Speed Change Event.

Visibility: Expert

**14.11.4.349 EventLinkTrigger0**

```
GenApi::IInteger& EventLinkTrigger0
```

Description: Returns the unique Identifier of the Link Trigger 0 type of Event.

Visibility: Expert

**14.11.4.350 EventLinkTrigger0FrameID**

```
GenApi::IInteger& EventLinkTrigger0FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 Event.

Visibility: Expert

**14.11.4.351 EventLinkTrigger0Timestamp**

```
GenApi::IInteger& EventLinkTrigger0Timestamp
```

Description: Returns the Timestamp of the Link Trigger 0 Event.

Visibility: Expert

**14.11.4.352 EventLinkTrigger1**

```
GenApi::IInteger& EventLinkTrigger1
```

Description: Returns the unique Identifier of the Link Trigger 1 type of Event.

Visibility: Expert

**14.11.4.353 EventLinkTrigger1FrameID**

```
GenApi::IInteger& EventLinkTrigger1FrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 Event.

Visibility: Expert

**14.11.4.354 EventLinkTrigger1Timestamp**

```
GenApi::IInteger& EventLinkTrigger1Timestamp
```

Description: Returns the Timestamp of the Link Trigger 1 Event.

Visibility: Expert

**14.11.4.355 EventNotification**

```
GenApi::IEnumerationT<EventNotificationEnums>& EventNotification
```

Description: Enables/Disables the selected event.

Visibility:

**14.11.4.356 EventSelector**

```
GenApi::IEnumerationT<EventSelectorEnums>& EventSelector
```

Description: Selects which Event to enable or disable.

Visibility:

**14.11.4.357 EventSequencerSetChange**

```
GenApi::IInteger& EventSequencerSetChange
```

Description: Returns the unique Identifier of the Sequencer Set Change type of Event.

Visibility: Expert

**14.11.4.358 EventSequencerSetChangeFrameID**

```
GenApi::IInteger& EventSequencerSetChangeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.

Visibility: Expert

**14.11.4.359 EventSequencerSetChangeTimestamp**

```
GenApi::IInteger& EventSequencerSetChangeTimestamp
```

Description: Returns the Timestamp of the Sequencer Set Change Event.

Visibility: Expert

**14.11.4.360 EventSerialData**

```
GenApi::IString& EventSerialData
```

Description: Returns the serial data that was received.

Visibility:

**14.11.4.361 EventSerialDataLength**

```
GenApi::IInteger& EventSerialDataLength
```

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

**14.11.4.362 EventSerialPortReceive**

```
GenApi::IInteger& EventSerialPortReceive
```

Description: Returns the unique identifier of the Serial Port Receive type of Event.

Visibility:

**14.11.4.363 EventSerialPortReceiveTimestamp**

```
GenApi::IInteger& EventSerialPortReceiveTimestamp
```

Description: Returns the Timestamp of the Serial Port Receive Event.

Visibility:

**14.11.4.364 EventSerialReceiveOverflow**

```
GenApi::IBoolean& EventSerialReceiveOverflow
```

Description: Returns the status of the event serial receive overflow.

Visibility:

**14.11.4.365 EventStream0TransferBlockEnd**

```
GenApi::IInteger& EventStream0TransferBlockEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.

Visibility: Expert

**14.11.4.366 EventStream0TransferBlockEndFrameID**

```
GenApi::IInteger& EventStream0TransferBlockEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.

Visibility: Expert

**14.11.4.367 EventStream0TransferBlockEndTimestamp**

```
GenApi::IInteger& EventStream0TransferBlockEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

Visibility: Expert

**14.11.4.368 EventStream0TransferBlockStart**

```
GenApi::IInteger& EventStream0TransferBlockStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of Event.

Visibility: Expert

**14.11.4.369 EventStream0TransferBlockStartFrameID**

```
GenApi::IInteger& EventStream0TransferBlockStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.

Visibility: Expert

**14.11.4.370 EventStream0TransferBlockStartTimestamp**

```
GenApi::IInteger& EventStream0TransferBlockStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Start Event.

Visibility: Expert

**14.11.4.371 EventStream0TransferBlockTrigger**

```
GenApi::IInteger& EventStream0TransferBlockTrigger
```

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.

Visibility: Expert

**14.11.4.372 EventStream0TransferBlockTriggerFrameID**

```
GenApi::IInteger& EventStream0TransferBlockTriggerFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

**14.11.4.373 EventStream0TransferBlockTriggerTimestamp**

```
GenApi::IInteger& EventStream0TransferBlockTriggerTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.

Visibility: Expert

**14.11.4.374 EventStream0TransferBurstEnd**

```
GenApi::IInteger& EventStream0TransferBurstEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.

Visibility: Expert

**14.11.4.375 EventStream0TransferBurstEndFrameID**

```
GenApi::IInteger& EventStream0TransferBurstEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.

Visibility: Expert

**14.11.4.376 EventStream0TransferBurstEndTimestamp**

```
GenApi::IInteger& EventStream0TransferBurstEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.

Visibility: Expert

**14.11.4.377 EventStream0TransferBurstStart**

```
GenApi::IInteger& EventStream0TransferBurstStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.

Visibility: Expert

**14.11.4.378 EventStream0TransferBurstStartFrameID**

```
GenApi::IInteger& EventStream0TransferBurstStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.

Visibility: Expert

**14.11.4.379 EventStream0TransferBurstStartTimestamp**

```
GenApi::IInteger& EventStream0TransferBurstStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.

Visibility: Expert

**14.11.4.380 EventStream0TransferEnd**

```
GenApi::IInteger& EventStream0TransferEnd
```

Description: Returns the unique Identifier of the Stream 0 Transfer End type of Event.

Visibility: Expert

**14.11.4.381 EventStream0TransferEndFrameID**

```
GenApi::IInteger& EventStream0TransferEndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End Event.

Visibility: Expert

**14.11.4.382 EventStream0TransferEndTimestamp**

```
GenApi::IInteger& EventStream0TransferEndTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer End Event.

Visibility: Expert

**14.11.4.383 EventStream0TransferOverflow**

```
GenApi::IInteger& EventStream0TransferOverflow
```

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.

Visibility: Expert

**14.11.4.384 EventStream0TransferOverflowFrameID**

```
GenApi::IInteger& EventStream0TransferOverflowFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.

Visibility: Expert

**14.11.4.385 EventStream0TransferOverflowTimestamp**

```
GenApi::IInteger& EventStream0TransferOverflowTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.

Visibility: Expert

**14.11.4.386 EventStream0TransferPause**

```
GenApi::IInteger& EventStream0TransferPause
```

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of Event.

Visibility: Expert

**14.11.4.387 EventStream0TransferPauseFrameID**

```
GenApi::IInteger& EventStream0TransferPauseFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause Event.

Visibility: Expert

**14.11.4.388 EventStream0TransferPauseTimestamp**

```
GenApi::IInteger& EventStream0TransferPauseTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Pause Event.

Visibility: Expert

**14.11.4.389 EventStream0TransferResume**

```
GenApi::IInteger& EventStream0TransferResume
```

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of Event.

Visibility: Expert

**14.11.4.390 EventStream0TransferResumeFrameID**

```
GenApi::IInteger& EventStream0TransferResumeFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume Event.

Visibility: Expert

**14.11.4.391 EventStream0TransferResumeTimestamp**

```
GenApi::IInteger& EventStream0TransferResumeTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Resume Event.

Visibility: Expert

**14.11.4.392 EventStream0TransferStart**

```
GenApi::IInteger& EventStream0TransferStart
```

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of Event.

Visibility: Expert

**14.11.4.393 EventStream0TransferStartFrameID**

```
GenApi::IInteger& EventStream0TransferStartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start Event.

Visibility: Expert

**14.11.4.394 EventStream0TransferStartTimestamp**

```
GenApi::IInteger& EventStream0TransferStartTimestamp
```

Description: Returns the Timestamp of the Stream 0 Transfer Start Event.

Visibility: Expert

**14.11.4.395 EventTest**

```
GenApi::IInteger& EventTest
```

Description: Returns the unique identifier of the Test type of Event.

Visibility:

**14.11.4.396 EventTestTimestamp**

```
GenApi::IInteger& EventTestTimestamp
```

Description: Returns the Timestamp of the Test Event.

Visibility:

**14.11.4.397 EventTimer0End**

```
GenApi::IInteger& EventTimer0End
```

Description: Returns the unique Identifier of the Timer 0 End type of Event.

Visibility: Expert

**14.11.4.398 EventTimer0EndFrameID**

```
GenApi::IInteger& EventTimer0EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End Event.

Visibility: Expert

**14.11.4.399 EventTimer0EndTimestamp**

```
GenApi::IInteger& EventTimer0EndTimestamp
```

Description: Returns the Timestamp of the Timer 0 End Event.

Visibility: Expert

**14.11.4.400 EventTimer0Start**

```
GenApi::IInteger& EventTimer0Start
```

Description: Returns the unique Identifier of the Timer 0 Start type of Event.

Visibility: Expert

**14.11.4.401 EventTimer0StartFrameID**

```
GenApi::IInteger& EventTimer0StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start Event.

Visibility: Expert

**14.11.4.402 EventTimer0StartTimestamp**

```
GenApi::IInteger& EventTimer0StartTimestamp
```

Description: Returns the Timestamp of the Timer 0 Start Event.

Visibility: Expert

**14.11.4.403 EventTimer1End**

```
GenApi::IInteger& EventTimer1End
```

Description: Returns the unique Identifier of the Timer 1 End type of Event.

Visibility: Expert

**14.11.4.404 EventTimer1EndFrameID**

```
GenApi::IInteger& EventTimer1EndFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End Event.

Visibility: Expert

**14.11.4.405 EventTimer1EndTimestamp**

```
GenApi::IInteger& EventTimer1EndTimestamp
```

Description: Returns the Timestamp of the Timer 1 End Event.

Visibility: Expert

#### 14.11.4.406 EventTimer1Start

```
GenApi::IInteger& EventTimer1Start
```

Description: Returns the unique Identifier of the Timer 1 Start type of Event.

Visibility: Expert

#### 14.11.4.407 EventTimer1StartFrameID

```
GenApi::IInteger& EventTimer1StartFrameID
```

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start Event.

Visibility: Expert

#### 14.11.4.408 EventTimer1StartTimestamp

```
GenApi::IInteger& EventTimer1StartTimestamp
```

Description: Returns the Timestamp of the Timer 1 Start Event.

Visibility: Expert

#### 14.11.4.409 ExposureActiveMode

```
GenApi::IEnumerationT<ExposureActiveModeEnums>& ExposureActiveMode
```

Description: Control sensor active exposure mode.

Visibility:

#### 14.11.4.410 ExposureAuto

```
GenApi::IEnumerationT<ExposureAutoEnums>& ExposureAuto
```

Description: Sets the automatic exposure mode Visibility:

#### 14.11.4.411 ExposureMode

```
GenApi::IEnumerationT<ExposureModeEnums>& ExposureMode
```

Description:

Sets the operation mode of the Exposure.

Visibility:

#### 14.11.4.412 ExposureTime

```
GenApi::IFloat& ExposureTime
```

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

#### 14.11.4.413 ExposureTimeMode

```
GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode
```

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

#### 14.11.4.414 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

#### 14.11.4.415 FactoryReset

```
GenApi:: ICommand& FactoryReset
```

Description: Returns all user tables to factory default Visibility:

#### 14.11.4.416 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

#### 14.11.4.417 FileAccessLength

`GenApi::IInteger& FileAccessLength`

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

#### 14.11.4.418 FileAccessOffset

`GenApi::IInteger& FileAccessOffset`

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

#### 14.11.4.419 FileMode

`GenApi::IEnumerationT<FileModeEnums>& FileMode`

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:

#### 14.11.4.420 FileOperationExecute

`GenApi:: ICommand& FileOperationExecute`

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

#### 14.11.4.421 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:

**14.11.4.422 FileOperationSelector**

```
GenApi::IEnumerationT<FileOperationSelectorEnums>& FileOperationSelector
```

Description:

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

Visibility:

**14.11.4.423 FileOperationStatus**

```
GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus
```

Description: Represents the file operation execution status.

Visibility:

**14.11.4.424 FileSelector**

```
GenApi::IEnumerationT<FileSelectorEnums>& FileSelector
```

Description:

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

Visibility:

**14.11.4.425 FileSize**

```
GenApi::IInteger& FileSize
```

Description: Represents the size of the selected file in bytes.

Visibility:

**14.11.4.426 Gain**

```
GenApi::IFloat& Gain
```

Description:

Controls the amplification of the video signal in dB.

Visibility:

#### 14.11.4.427 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:

#### 14.11.4.428 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

#### 14.11.4.429 GainSelector

`GenApi::IEnumerationT<GainSelectorEnums>& GainSelector`

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

#### 14.11.4.430 Gamma

`GenApi::IFloat& Gamma`

Description: Controls the gamma correction of pixel intensity.

Visibility:

#### 14.11.4.431 GammaEnable

`GenApi::IBoolean& GammaEnable`

Description: Enables/disables gamma correction.

Visibility:

**14.11.4.432 GevActiveLinkCount**

```
GenApi::IInteger& GevActiveLinkCount
```

Description: Indicates the current number of active logical links.

Visibility: Expert

**14.11.4.433 GevCCP**

```
GenApi::IEnumerationT<GevCCPEnums>& GevCCP
```

Description: Controls the device access privilege of an application.

Visibility:

**14.11.4.434 GevCurrentDefaultGateway**

```
GenApi::IInteger& GevCurrentDefaultGateway
```

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

**14.11.4.435 GevCurrentIPAddress**

```
GenApi::IInteger& GevCurrentIPAddress
```

Description: Reports the IP address for the given logical link.

Visibility:

**14.11.4.436 GevCurrentIPConfigurationDHCP**

```
GenApi::IBoolean& GevCurrentIPConfigurationDHCP
```

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

**14.11.4.437 GevCurrentIPConfigurationLLA**

```
GenApi::IBoolean& GevCurrentIPConfigurationLLA
```

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

**14.11.4.438 GevCurrentIPConfigurationPersistentIP**

```
GenApi::IBoolean& GevCurrentIPConfigurationPersistentIP
```

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

**14.11.4.439 GevCurrentPhysicalLinkConfiguration**

```
GenApi::IEnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration
```

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

**14.11.4.440 GevCurrentSubnetMask**

```
GenApi::IInteger& GevCurrentSubnetMask
```

Description: Reports the subnet mask of the given logical link.

Visibility:

**14.11.4.441 GevDiscoveryAckDelay**

```
GenApi::IInteger& GevDiscoveryAckDelay
```

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

**14.11.4.442 GevFirstURL**

```
GenApi::IString& GevFirstURL
```

Description: The first choice of URL for the XML device description file.

Visibility:

**14.11.4.443 GevGVCPExtendedStatusCodes**

```
GenApi::IBoolean& GevGVCPExtendedStatusCodes
```

Description: Enables the generation of extended status codes.

Visibility: Guru

**14.11.4.444 GevGVCPExtendedStatusCodesSelector**

```
GenApi::IEnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodes←
Selector
```

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

Visibility: Guru

**14.11.4.445 GevGVCPHeartbeatDisable**

```
GenApi::IBoolean& GevGVCPHeartbeatDisable
```

Description: Disables the GVCP heartbeat.

Visibility:

**14.11.4.446 GevGVCPPendingAck**

```
GenApi::IBoolean& GevGVCPPendingAck
```

Description: Enables the generation of PENDING\_ACK.

Visibility:

**14.11.4.447 GevGVCPPendingTimeout**

```
GenApi::IInteger& GevGVCPPendingTimeout
```

Description: Indicates the longest GVCP command execution time before the device returns a PENDING\_ACK in milliseconds.

Visibility:

**14.11.4.448 GevGVSPExtendedIDMode**

```
GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode
```

Description: Enables the extended IDs mode.

Visibility: Expert

**14.11.4.449 GevHeartbeatTimeout**

```
GenApi::IInteger& GevHeartbeatTimeout
```

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

#### 14.11.4.450 GevIEEE1588

`GenApi::IBoolean& GevIEEE1588`

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

#### 14.11.4.451 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:

#### 14.11.4.452 GevIEEE1588Mode

`GenApi::IEnumerationT<GevIEEE1588ModeEnums>& GevIEEE1588Mode`

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

#### 14.11.4.453 GevIEEE1588Status

`GenApi::IEnumerationT<GevIEEE1588StatusEnums>& GevIEEE1588Status`

Description: Provides the status of the IEEE 1588 clock.

Visibility:

#### 14.11.4.454 GevInterfaceSelector

`GenApi::IInteger& GevInterfaceSelector`

Description: Selects which logical link to control.

Visibility:

#### 14.11.4.455 GevIPConfigurationStatus

`GenApi::IEnumerationT<GevIPConfigurationStatusEnums>& GevIPConfigurationStatus`

Description: Reports the current IP configuration status.

Visibility: Beginner

**14.11.4.456 GevMACAddress**

```
GenApi::IInteger& GevMACAddress
```

Description: MAC address of the logical link.

Visibility:

**14.11.4.457 GevMCDA**

```
GenApi::IInteger& GevMCDA
```

Description: Controls the destination IP address of the message channel Visibility:

**14.11.4.458 GevMCPHostPort**

```
GenApi::IInteger& GevMCPHostPort
```

Description: The port to which the device must send messages Visibility:

**14.11.4.459 GevMCRC**

```
GenApi::IInteger& GevMCRC
```

Description: Indicates the number of retries of the message channel.

Visibility:

**14.11.4.460 GevMCSP**

```
GenApi::IInteger& GevMCSP
```

Description: Indicates the source port of the message channel.

Visibility:

**14.11.4.461 GevMCTT**

```
GenApi::IInteger& GevMCTT
```

Description: Indicates the transmission timeout of the message channel.

Visibility:

**14.11.4.462 GevNumberOfInterfaces**

```
GenApi::IInteger& GevNumberOfInterfaces
```

Description: Indicates the number of physical network interfaces supported by this device.

Visibility:

**14.11.4.463 GevPAUSEFrameReception**

```
GenApi::IBoolean& GevPAUSEFrameReception
```

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

**14.11.4.464 GevPAUSEFrameTransmission**

```
GenApi::IBoolean& GevPAUSEFrameTransmission
```

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

**14.11.4.465 GevPersistentDefaultGateway**

```
GenApi::IInteger& GevPersistentDefaultGateway
```

Description: Controls the persistent default gateway for this logical link.

Visibility:

**14.11.4.466 GevPersistentIPAddress**

```
GenApi::IInteger& GevPersistentIPAddress
```

Description: Controls the Persistent IP address for this logical link.

Visibility:

**14.11.4.467 GevPersistentSubnetMask**

```
GenApi::IInteger& GevPersistentSubnetMask
```

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

**14.11.4.468 GevPhysicalLinkConfiguration**

```
GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration
```

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

Visibility: Expert

**14.11.4.469 GevPrimaryApplicationIPAddress**

```
GenApi::IInteger& GevPrimaryApplicationIPAddress
```

Description: Returns the address of the primary application.

Visibility: Guru

**14.11.4.470 GevPrimaryApplicationSocket**

```
GenApi::IInteger& GevPrimaryApplicationSocket
```

Description: Returns the UDP source port of the primary application.

Visibility: Guru

**14.11.4.471 GevPrimaryApplicationSwitchoverKey**

```
GenApi::IInteger& GevPrimaryApplicationSwitchoverKey
```

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

**14.11.4.472 GevSCCFGAllInTransmission**

```
GenApi::IBoolean& GevSCCFGAllInTransmission
```

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

**14.11.4.473 GevSCCFGExtendedChunkData**

```
GenApi::IBoolean& GevSCCFGExtendedChunkData
```

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

#### 14.11.4.474 GevSCCFGPacketResendDestination

`GenApi::IBoolean& 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

#### 14.11.4.475 GevSCCFGUnconditionalStreaming

`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).

Visibility:

#### 14.11.4.476 GevSCDA

`GenApi::IInteger& 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:

#### 14.11.4.477 GevSCPD

`GenApi::IInteger& 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:

#### 14.11.4.478 GevSCPDirection

`GenApi::IInteger& GevSCPDirection`

Description: Transmit or Receive of the channel Visibility:

#### 14.11.4.479 GevSCPHostPort

`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.

Visibility:

**14.11.4.480 GevSCPIInterfaceIndex**

```
GenApi::IInteger& GevSCPIInterfaceIndex
```

Description: Index of the logical link to use.

Visibility:

**14.11.4.481 GevSCPSBigEndian**

```
GenApi::IBoolean& GevSCPSBigEndian
```

Description: Endianess of multi-byte pixel data for this stream.

Visibility:

**14.11.4.482 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:

**14.11.4.483 GevSCPSFireTestPacket**

```
GenApi::IBoolean& GevSCPSFireTestPacket
```

Description: Sends a test packet.

Visibility:

**14.11.4.484 GevSCSPacketSize**

```
GenApi::IInteger& GevSCSPacketSize
```

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

**14.11.4.485 GevSCSP**

```
GenApi::IInteger& GevSCSP
```

Description: Indicates the source port of the stream channel.

Visibility:

**14.11.4.486 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

**14.11.4.487 GevSCZoneCount**

`GenApi::IInteger& GevSCZoneCount`

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

**14.11.4.488 GevSCZoneDirectionAll**

`GenApi::IInteger& GevSCZoneDirectionAll`

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

**14.11.4.489 GevSecondURL**

`GenApi::IString& GevSecondURL`

Description: The second choice of URL to the XML device description file.

Visibility:

**14.11.4.490 GevStreamChannelSelector**

`GenApi::IInteger& GevStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility:

**14.11.4.491 GevSupportedOption**

`GenApi::IBoolean& GevSupportedOption`

Description: Returns if the selected GEV option is supported.

Visibility:

**14.11.4.492 GevSupportedOptionSelector**

```
GenApi::IEnumeration<GevSupportedOptionSelectorEnums>& GevSupportedOptionSelector
```

Description: Selects the GEV option to interrogate for existing support.

Visibility:

**14.11.4.493 GevTimestampTickFrequency**

```
GenApi::IInteger& GevTimestampTickFrequency
```

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

**14.11.4.494 GuiXmlManifestAddress**

```
GenApi::IInteger& GuiXmlManifestAddress
```

Description: Location of the GUI XML manifest table.

Visibility:

**14.11.4.495 Height**

```
GenApi::IInteger& Height
```

Description:

Height of the image provided by the device (in pixels).

Visibility:

**14.11.4.496 HeightMax**

```
GenApi::IInteger& 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:

#### 14.11.4.497 ImageComponentEnable

`GenApi::IBoolean& ImageComponentEnable`

Description: Controls if the selected component streaming is active.

Visibility: Beginner

#### 14.11.4.498 ImageComponentSelector

`GenApi::IEnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector`

Description: Selects a component to activate data streaming from.

Visibility: Beginner

#### 14.11.4.499 ImageCompressionBitrate

`GenApi::IFloat& ImageCompressionBitrate`

Description: Control the rate of the produced compressed stream.

Visibility: Expert

#### 14.11.4.500 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

#### 14.11.4.501 ImageCompressionMode

`GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode`

Description: Visibility:

#### 14.11.4.502 ImageCompressionQuality

`GenApi::IInteger& ImageCompressionQuality`

Description: Control the quality of the produced compressed stream.

Visibility: Expert

#### 14.11.4.503 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

#### 14.11.4.504 IspEnable

`GenApi::IBoolean& IspEnable`

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e. mono).

Visibility:

#### 14.11.4.505 LineFilterWidth

`GenApi::IFloat& LineFilterWidth`

Description: Filter width in microseconds for the selected line and filter combination Visibility:

#### 14.11.4.506 LineFormat

`GenApi::IEnumerationT<LineFormatEnums>& LineFormat`

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

Visibility:

#### 14.11.4.507 LineInputFilterSelector

`GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector`

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

Visibility:

#### 14.11.4.508 LineInverter

`GenApi::IBoolean& LineInverter`

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

#### 14.11.4.509 LineMode

`GenApi::IEnumerationT<LineModeEnums>& LineMode`

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

Visibility:

#### 14.11.4.510 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

#### 14.11.4.511 LineSelector

`GenApi::IEnumerationT<LineSelectorEnums>& LineSelector`

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

#### 14.11.4.512 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:

#### 14.11.4.513 LineStatus

`GenApi::IBoolean& LineStatus`

Description: Returns the current status of the selected input or output Line Visibility:

**14.11.4.514 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:

**14.11.4.515 LinkErrorCount**

```
GenApi::IInteger& LinkErrorCount
```

Description: Counts the number of error on the link.

Visibility:

**14.11.4.516 LinkUptime**

```
GenApi::IInteger& LinkUptime
```

Description: Time since the last phy negotiation (enumeration).

Visibility:

**14.11.4.517 LogicBlockLUTInputActivation**

```
GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation
```

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

**14.11.4.518 LogicBlockLUTInputSelector**

```
GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector
```

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

**14.11.4.519 LogicBlockLUTInputSource**

```
GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource
```

Description: Selects the source for the input into the Logic LUT.

Visibility:

#### 14.11.4.520 LogicBlockLUTOutputValue

`GenApi::IBoolean& LogicBlockLUTOutputValue`

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

#### 14.11.4.521 LogicBlockLUTOutputValueAll

`GenApi::IInteger& LogicBlockLUTOutputValueAll`

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

#### 14.11.4.522 LogicBlockLUTRowIndex

`GenApi::IInteger& LogicBlockLUTRowIndex`

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

#### 14.11.4.523 LogicBlockLUTSelector

`GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector`

Description: Selects which LogicBlock LUT to configure Visibility:

#### 14.11.4.524 LogicBlockSelector

`GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector`

Description: Selects which LogicBlock to configure Visibility:

#### 14.11.4.525 LUTEnable

`GenApi::IBoolean& LUTEnable`

Description:

Activates the selected LUT.

Visibility:

**14.11.4.526 LUTIndex**

```
GenApi::IInteger& LUTIndex
```

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

**14.11.4.527 LUTSelector**

```
GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector
```

Description:

Selects which LUT to control.

Visibility:

**14.11.4.528 LUTValue**

```
GenApi::IInteger& LUTValue
```

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

**14.11.4.529 LUTValueAll**

```
GenApi::IRegister& LUTValueAll
```

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

**14.11.4.530 MaxDeviceResetTime**

`GenApi::IInteger& MaxDeviceResetTime`

Description: Time to wait until device reset complete (ms).

Visibility:

**14.11.4.531 OffsetX**

`GenApi::IInteger& OffsetX`

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

**14.11.4.532 OffsetY**

`GenApi::IInteger& OffsetY`

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

**14.11.4.533 PacketResendRequestCount**

`GenApi::IInteger& PacketResendRequestCount`

Description: Counts the number of resend requests received from the host.

Visibility:

**14.11.4.534 PayloadSize**

`GenApi::IInteger& PayloadSize`

Description: Provides the number of bytes transferred for each image or chunk on the stream channel.

Visibility:

**14.11.4.535 PixelColorFilter**

```
GenApi::IEnumerationT<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:

**14.11.4.536 PixelDynamicRangeMax**

```
GenApi::IInteger& 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:

**14.11.4.537 PixelDynamicRangeMin**

```
GenApi::IInteger& 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:

**14.11.4.538 PixelFormat**

```
GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat
```

Description: Format of the pixel provided by the camera.

Visibility:

**14.11.4.539 PixelFormatInfoID**

```
GenApi::IInteger& PixelFormatInfoID
```

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

---

#### 14.11.4.540 PixelFormatInfoSelector

`GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector`

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

#### 14.11.4.541 PixelSize

`GenApi::IEnumerationT<PixelSizeEnums>& PixelSize`

Description: Total size in bits of a pixel of the image.

Visibility:

#### 14.11.4.542 PowerSupplyCurrent

`GenApi::IFloat& PowerSupplyCurrent`

Description:

Indicates the output current of the selected power supply (A).

Visibility:

#### 14.11.4.543 PowerSupplyVoltage

`GenApi::IFloat& PowerSupplyVoltage`

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

#### 14.11.4.544 RegionDestination

`GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination`

Description: Control the destination of the selected region.

Visibility: Expert

#### 14.11.4.545 RegionMode

```
GenApi::IEnumerationT<RegionModeEnums>& RegionMode
```

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

Visibility: Beginner

#### 14.11.4.546 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

#### 14.11.4.547 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:

#### 14.11.4.548 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:

#### 14.11.4.549 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:

#### 14.11.4.550 Saturation

`GenApi::IFloat& Saturation`

Description: Controls the color saturation.

Visibility:

#### 14.11.4.551 SaturationEnable

`GenApi::IBoolean& SaturationEnable`

Description: Enables/disables Saturation adjustment.

Visibility:

#### 14.11.4.552 Scan3dAxisMax

`GenApi::IFloat& Scan3dAxisMax`

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

#### 14.11.4.553 Scan3dAxisMin

`GenApi::IFloat& Scan3dAxisMin`

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

#### 14.11.4.554 Scan3dCoordinateOffset

`GenApi::IFloat& Scan3dCoordinateOffset`

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

#### 14.11.4.555 Scan3dCoordinateReferenceSelector

`GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReference<-- Selector`

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

**14.11.4.556 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

**14.11.4.557 Scan3dCoordinateScale**

```
GenApi::IFloat& Scan3dCoordinateScale
```

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

**14.11.4.558 Scan3dCoordinateSelector**

```
GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector
```

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

Visibility: Expert

**14.11.4.559 Scan3dCoordinateSystem**

```
GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem
```

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

**14.11.4.560 Scan3dCoordinateSystemReference**

```
GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference
```

Description: Defines coordinate system reference location.

Visibility: Expert

**14.11.4.561 Scan3dCoordinateTransformSelector**

```
GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransform<--> Selector
```

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

Visibility: Expert

#### 14.11.4.562 Scan3dDistanceUnit

`GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit`

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

#### 14.11.4.563 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

#### 14.11.4.564 Scan3dInvalidDataValue

`GenApi::IFloat& Scan3dInvalidDataValue`

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

#### 14.11.4.565 Scan3dOutputMode

`GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode`

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

Visibility: Expert

#### 14.11.4.566 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

#### 14.11.4.567 SensorDescription

`GenApi::IString& SensorDescription`

Description: Returns Sensor Description Visibility:

**14.11.4.568 SensorDigitizationTaps**

```
GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps
```

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

Visibility: Expert

**14.11.4.569 SensorHeight**

```
GenApi::IInteger& SensorHeight
```

Description: Effective height of the sensor in pixels.

Visibility:

**14.11.4.570 SensorShutterMode**

```
GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode
```

Description: Sets the shutter mode of the device.

Visibility:

**14.11.4.571 SensorTaps**

```
GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps
```

Description: Number of taps of the camera sensor.

Visibility: Expert

**14.11.4.572 SensorWidth**

```
GenApi::IInteger& SensorWidth
```

Description: Effective width of the sensor in pixels.

Visibility:

**14.11.4.573 SequencerConfigurationMode**

```
GenApi::IEnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode
```

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

#### 14.11.4.574 SequencerConfigurationValid

`GenApi::IEnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid`

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

#### 14.11.4.575 SequencerFeatureEnable

`GenApi::IBoolean& SequencerFeatureEnable`

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

#### 14.11.4.576 SequencerMode

`GenApi::IEnumerationT<SequencerModeEnums>& SequencerMode`

Description: Controls whether or not a sequencer is active.

Visibility:

#### 14.11.4.577 SequencerPathSelector

`GenApi::IInteger& SequencerPathSelector`

Description:

Selects branching path to be used for subsequent settings.

Visibility:

**14.11.4.578 SequencerSetActive**

`GenApi::IInteger& SequencerSetActive`

Description: Displays the currently active sequencer set.

Visibility:

**14.11.4.579 SequencerSetLoad**

`GenApi:: ICommand& SequencerSetLoad`

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

**14.11.4.580 SequencerSetNext**

`GenApi::IInteger& SequencerSetNext`

Description: Specifies the next sequencer set.

Visibility:

**14.11.4.581 SequencerSetSave**

`GenApi:: ICommand& SequencerSetSave`

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

**14.11.4.582 SequencerSetSelector**

`GenApi::IInteger& SequencerSetSelector`

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

**14.11.4.583 SequencerSetStart**

```
GenApi::IInteger& SequencerSetStart
```

Description: Sets the first sequencer set to be used.

Visibility:

**14.11.4.584 SequencerSetValid**

```
GenApi::IEnumerationT<SequencerSetValidEnums>& SequencerSetValid
```

Description:

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

Visibility:

**14.11.4.585 SequencerTriggerActivation**

```
GenApi::IEnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation
```

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

**14.11.4.586 SequencerTriggerSource**

```
GenApi::IEnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource
```

Description:

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

Visibility:

**14.11.4.587 SerialPortBaudRate**

```
GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate
```

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

Visibility:

**14.11.4.588 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:

**14.11.4.589 SerialPortParity**

```
GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity
```

Description: This feature controls the parity used by the selected serial port.

Visibility:

**14.11.4.590 SerialPortSelector**

```
GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector
```

Description: Selects which serial port of the device to control.

Visibility:

**14.11.4.591 SerialPortSource**

```
GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource
```

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

Visibility:

**14.11.4.592 SerialPortStopBits**

```
GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits
```

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

Visibility:

**14.11.4.593 SerialReceiveFramingErrorCount**

```
GenApi::IInteger& SerialReceiveFramingErrorCount
```

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

**14.11.4.594 SerialReceiveParityErrorCount**

```
GenApi::IInteger& SerialReceiveParityErrorCount
```

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

**14.11.4.595 SerialReceiveQueueClear**

```
GenApi:: ICommand& SerialReceiveQueueClear
```

Description: This is a command that clears the device serial port receive queue.

Visibility:

**14.11.4.596 SerialReceiveQueueCurrentCharacterCount**

```
GenApi::IInteger& SerialReceiveQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

**14.11.4.597 SerialReceiveQueueMaxCharacterCount**

```
GenApi::IInteger& SerialReceiveQueueMaxCharacterCount
```

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

**14.11.4.598 SerialTransmitQueueCurrentCharacterCount**

```
GenApi::IInteger& SerialTransmitQueueCurrentCharacterCount
```

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

**14.11.4.599 SerialTransmitQueueMaxCharacterCount**

```
GenApi::IInteger& SerialTransmitQueueMaxCharacterCount
```

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

**14.11.4.600 Sharpening**

```
GenApi::IFloat& 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:

**14.11.4.601 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:

**14.11.4.602 SharpeningEnable**

```
GenApi::IBoolean& SharpeningEnable
```

Description:

Enables/disables the sharpening feature. Sharpening is disabled by default.

Visibility:

#### 14.11.4.603 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:

#### 14.11.4.604 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

#### 14.11.4.605 SoftwareSignalSelector

`GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector`

Description: Selects which Software Signal features to control.

Visibility: Beginner

#### 14.11.4.606 SourceCount

`GenApi::IInteger& SourceCount`

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

#### 14.11.4.607 SourceSelector

`GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector`

Description: Selects the source to control.

Visibility: Beginner

**14.11.4.608 Test0001**

```
GenApi::IInteger& Test0001
```

Description: For testing only.

Visibility:

**14.11.4.609 TestEventGenerate**

```
GenApi:: ICommand& TestEventGenerate
```

Description: This command generates a test event and sends it to the host.

Visibility:

**14.11.4.610 TestPattern**

```
GenApi:: IEnumerationT<TestPatternEnums>& TestPattern
```

Description:

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

Visibility:

**14.11.4.611 TestPatternGeneratorSelector**

```
GenApi:: IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector
```

Description:

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

Visibility:

**14.11.4.612 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

#### 14.11.4.613 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

#### 14.11.4.614 TimerDuration

`GenApi::IFloat& TimerDuration`

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

#### 14.11.4.615 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

#### 14.11.4.616 TimerSelector

`GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector`

Description: Selects which Timer to configure.

Visibility: Expert

#### 14.11.4.617 TimerStatus

`GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus`

Description: Returns the current status of the Timer.

Visibility: Expert

#### 14.11.4.618 TimerTriggerActivation

`GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation`

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

**14.11.4.619 TimerTriggerSource**

```
GenApi::IEnumeration<TimerTriggerSourceEnums>& TimerTriggerSource
```

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

**14.11.4.620 TimerValue**

```
GenApi::IFloat& TimerValue
```

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

**14.11.4.621 Timestamp**

```
GenApi::IInteger& Timestamp
```

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

**14.11.4.622 TimestampLatch**

```
GenApi:: ICommand& TimestampLatch
```

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

**14.11.4.623 TimestampLatchValue**

```
GenApi::IInteger& TimestampLatchValue
```

Description: Returns the latched value of the timestamp counter.

Visibility:

**14.11.4.624 TimestampReset**

```
GenApi:: ICommand& TimestampReset
```

Description: Resets the current value of the device timestamp counter.

Visibility:

**14.11.4.625 TLParamsLocked**

```
GenApi::IInteger& TLParamsLocked
```

Description: Visibility:

**14.11.4.626 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

**14.11.4.627 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:

**14.11.4.628 TransferBurstCount**

```
GenApi::IInteger& TransferBurstCount
```

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

**14.11.4.629 TransferComponentSelector**

```
GenApi::IEnumeration<TransferComponentSelectorEnums>& TransferComponentSelector
```

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

**14.11.4.630 TransferControlMode**

```
GenApi::IEnumeration<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:

**14.11.4.631 TransferOperationMode**

```
GenApi::IEnumerationT<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:

**14.11.4.632 TransferPause**

```
GenApi:: ICommand& 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

**14.11.4.633 TransferQueueCurrentBlockCount**

```
GenApi::IInteger& TransferQueueCurrentBlockCount
```

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

**14.11.4.634 TransferQueueMaxBlockCount**

```
GenApi::IInteger& TransferQueueMaxBlockCount
```

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

**14.11.4.635 TransferQueueMode**

```
GenApi::IEnumerationT<TransferQueueModeEnums>& TransferQueueMode
```

Description: Specifies the operation mode of the transfer queue.

Visibility:

**14.11.4.636 TransferQueueOverflowCount**

```
GenApi::IInteger& TransferQueueOverflowCount
```

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

Visibility:

#### 14.11.4.637 TransferResume

`GenApi:: ICommand& TransferResume`

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

#### 14.11.4.638 TransferSelector

`GenApi:: IEnumerationT<TransferSelectorEnums>& TransferSelector`

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

Visibility: Expert

#### 14.11.4.639 TransferStart

`GenApi:: ICommand& 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:

#### 14.11.4.640 TransferStatus

`GenApi:: IBoolean& TransferStatus`

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

#### 14.11.4.641 TransferStatusSelector

`GenApi:: IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector`

Description: Selects which status of the transfer module to read.

Visibility: Guru

#### 14.11.4.642 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:

**14.11.4.643 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

**14.11.4.644 TransferTriggerActivation**

```
GenApi::IEnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation
```

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

**14.11.4.645 TransferTriggerMode**

```
GenApi::IEnumerationT<TransferTriggerModeEnums>& TransferTriggerMode
```

Description: Controls if the selected trigger is active.

Visibility: Guru

**14.11.4.646 TransferTriggerSelector**

```
GenApi::IEnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector
```

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

**14.11.4.647 TransferTriggerSource**

```
GenApi::IEnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource
```

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

**14.11.4.648 TriggerActivation**

```
GenApi::IEnumerationT<TriggerActivationEnums>& TriggerActivation
```

Description: Specifies the activation mode of the trigger.

Visibility:

#### 14.11.4.649 TriggerDelay

`GenApi::IFloat& TriggerDelay`

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

#### 14.11.4.650 TriggerDivider

`GenApi::IInteger& TriggerDivider`

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

#### 14.11.4.651 TriggerEventTest

`GenApi:: ICommand& TriggerEventTest`

Description: This command generates a test event and sends it to the host.

Visibility:

#### 14.11.4.652 TriggerMode

`GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode`

Description:

Controls whether or not trigger is active.

Visibility:

#### 14.11.4.653 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

**14.11.4.654 TriggerOverlap**

```
GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap
```

Description: Specifies the overlap mode of the trigger.

Visibility:

**14.11.4.655 TriggerSelector**

```
GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector
```

Description: Selects the type of trigger to configure.

Visibility:

**14.11.4.656 TriggerSoftware**

```
GenApi:: ICommand& TriggerSoftware
```

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

**14.11.4.657 TriggerSource**

```
GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource
```

Description:

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

Visibility:

**14.11.4.658 UserOutputSelector**

```
GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector
```

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

Visibility:

#### 14.11.4.659 UserOutputValue

```
GenApi::IBoolean& UserOutputValue
```

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

#### 14.11.4.660 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:

#### 14.11.4.661 UserOutputValueAllMask

```
GenApi::IInteger& 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

#### 14.11.4.662 UserSetDefault

```
GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault
```

Description:

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

Visibility:

#### 14.11.4.663 UserSetFeatureEnable

```
GenApi::IBoolean& UserSetFeatureEnable
```

Description: Whether or not the selected feature is saved to user sets.

Visibility:

**14.11.4.664 UserSetLoad**

```
GenApi:: ICommand& UserSetLoad
```

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

**14.11.4.665 UserSetSave**

```
GenApi:: ICommand& UserSetSave
```

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

**14.11.4.666 UserSetSelector**

```
GenApi:: IEnumerationT<UserSetSelectorEnums>& UserSetSelector
```

Description:

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

Visibility:

**14.11.4.667 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:

#### 14.11.4.668 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

#### 14.11.4.669 WhiteClipSelector

`GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector`

Description: Selects which White Clip to control.

Visibility: Expert

#### 14.11.4.670 Width

`GenApi::IInteger& Width`

Description:

Width of the image provided by the device (in pixels).

Visibility:

#### 14.11.4.671 WidthMax

`GenApi::IInteger& 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:

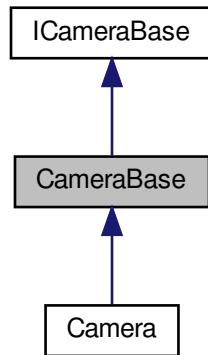
The documentation for this class was generated from the following file:

- [include/Camera.h](#)

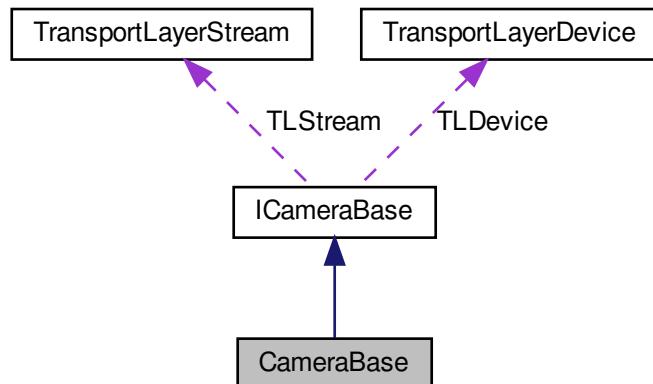
## 14.12 CameraBase Class Reference

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 Connect to camera, retrieve XML and generate node map.*

- void **DelInit** ()
 

*DelInit Disconnect camera port and free 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** () 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)
 

*ReadPort*
- void **WritePort** (uint64\_t iAddress, const void \*pBuffer, size\_t iSize)
 

*WritePort*
- 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 streamID=0)
 

*GetNextImage Gets the next image that was received by the transport layer.*
- **GenICam::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.*
- **GenICam::gcstring GetGuiXml** () const
 

*Returns the GUI XML that can be passed into the Spinnaker GUI framework.*
- void **RegisterEventHandler** (EventHandler &evtHandlerToRegister)
 

*RegisterEventHandler(EventHandler &) Registers a specific event handler for the camera.*
- void **RegisterEventHandler** (EventHandler &evtHandlerToRegister, const **GenICam::gcstring** &eventName)
 

*RegisterEventHandler(EventHandler &, const GenICam::gcstring&) Registers a specific event handler for the camera.*
- void **UnregisterEventHandler** (EventHandler &evtHandlerToUnregister)
 

*UnregisterEventHandler Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera DelInit().*

- `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

### 14.12.1 Detailed Description

The base class for the camera object.

### 14.12.2 Constructor & Destructor Documentation

#### 14.12.2.1 ~CameraBase()

```
virtual ~CameraBase (
    void ) [virtual]
```

Virtual Destructor.

#### 14.12.2.2 CameraBase() [1/2]

```
CameraBase (
    void ) [protected]
```

Default constructor.

#### 14.12.2.3 CameraBase() [2/2]

```
CameraBase (
    const CameraBase & ) [protected]
```

Copy constructor.

### 14.12.3 Member Function Documentation

#### 14.12.3.1 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](#).

#### 14.12.3.2 DeInit()

```
void DeInit ( ) [virtual]
```

DeInit Disconnect camera port and free [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](#).

#### 14.12.3.3 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](#).

#### 14.12.3.4 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\(\)](#) need to be released first using `image->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\( grabTimeout \)](#)
- [Image::Release\(\)](#)

Implements [ICameraBase](#).

#### 14.12.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements [ICameraBase](#).

#### 14.12.3.6 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](#).

#### 14.12.3.7 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](#).

#### 14.12.3.8 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](#).

### 14.12.3.9 GetNextImage()

```
ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE,
    uint64_t streamID = 0 ) [virtual]
```

GetNextImage Gets the next image that was received by the transport layer.

This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID 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>streamID</i>    | The stream to grab the image.                           |

#### Returns

pointer to an [Image](#) object

Implements [ICameraBase](#).

### 14.12.3.10 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](#).

#### 14.12.3.11 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

##### Returns

The number of data streams

Implements [ICameraBase](#).

#### 14.12.3.12 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](#).

#### 14.12.3.13 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](#).

#### 14.12.3.14 GetTLStreamNodeMap()

```
GenApi::INodeMap& GetTLStreamNodeMap ( ) 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.

##### Returns

A reference to the INodeMap.

Implements [ICameraBase](#).

#### 14.12.3.15 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](#).

#### 14.12.3.16 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](#).

#### 14.12.3.17 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](#).

#### 14.12.3.18 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](#).

#### 14.12.3.19 Init()

```
void Init ( ) [virtual]
```

Init Connect to camera, retrieve XML and generate node map.

This function needs to be called before any camera related API calls such as [BeginAcquisition\(\)](#), [EndAcquisition\(\)](#), [GetNodeMap\(\)](#), [GetNextImage\(\)](#).

See also

[BeginAcquisition\(\)](#)

[EndAcquisition\(\)](#)

[GetNodeMap\(\)](#)

[GetNextImage\(\)](#)

Implements [ICameraBase](#).

#### 14.12.3.20 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](#).

#### 14.12.3.21 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](#).

#### 14.12.3.22 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](#).

#### 14.12.3.23 operator=( )

```
CameraBase& operator= (
    const CameraBase & ) [protected]
```

Assignment operator.

#### 14.12.3.24 ReadPort()

```
void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

#### 14.12.3.25 RegisterEventHandler() [1/2]

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

[RegisterEventHandler\(EventHandler &\)](#) Registers a specific 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\(\)](#)

##### Parameters

|                                   |                                              |
|-----------------------------------|----------------------------------------------|
| <code>evtHandlerToRegister</code> | The event handler to register for the camera |
|-----------------------------------|----------------------------------------------|

Implements [ICameraBase](#).

#### 14.12.3.26 RegisterEventHandler() [2/2]

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [virtual]
```

[RegisterEventHandler\(EventHandler &, const GenICam::gcstring&\)](#) Registers a specific event handler for the camera.

See also

[Init\(\)](#)

Parameters

|                                   |                                              |
|-----------------------------------|----------------------------------------------|
| <code>evtHandlerToRegister</code> | The event handler to register for the camera |
| <code>eventName</code>            | The event name to register                   |

Implements [ICameraBase](#).

#### 14.12.3.27 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

|                   |                                                                    |
|-------------------|--------------------------------------------------------------------|
| <code>mode</code> | <a href="#">System</a> owned or <a href="#">User</a> owned buffers |
|-------------------|--------------------------------------------------------------------|

Implements [ICameraBase](#).

#### 14.12.3.28 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](#).

#### 14.12.3.29 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:  $((\text{unsigned int}) (\text{bufferSize} + 1024 - 1) / 1024) * 1024$  where 1024 is the USB3 packet size.

##### See also

[GetBufferOwnership\(\)](#)  
[SetBufferOwnership\(\)](#)  
 [GetUserBufferCount\(\)](#)  
 [GetUserBufferSize\(\)](#)  
 [GetUserBufferTotalSize\(\)](#)

**Parameters**

|                    |                                                                        |
|--------------------|------------------------------------------------------------------------|
| <i>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](#).

**14.12.3.30 UnregisterEventHandler()**

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

**UnregisterEventHandler** Unregisters an event handler for the camera Event handlers should be unregistered first before calling camera [DelInit\(\)](#).

Otherwise an exception will be thrown in the [DelInit\(\)](#) call and require the user to unregister event handlers before the camera can be re-initialized again.

**See also**

[DelInit\(\)](#)

**Parameters**

|                               |                                                 |
|-------------------------------|-------------------------------------------------|
| <i>evtHandlerToUnregister</i> | The event handler to unregister from the camera |
|-------------------------------|-------------------------------------------------|

Implements [ICameraBase](#).

**14.12.3.31 WritePort()**

```
void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

**14.12.4 Friends And Related Function Documentation**

#### 14.12.4.1 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

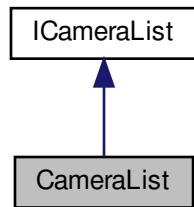
The documentation for this class was generated from the following file:

- [include/ CameraBase.h](#)

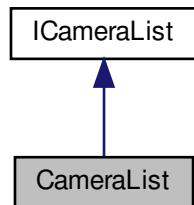
### 14.13 CameraList Class Reference

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 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 (CameraList &otherList)`  
`Appends a camera list to the current list.`

## Additional Inherited Members

### 14.13.1 Detailed Description

Used to hold a list of camera objects.

### 14.13.2 Constructor & Destructor Documentation

#### 14.13.2.1 CameraList() [1/2]

```
CameraList (
    void )
```

Default constructor.

#### 14.13.2.2 ~CameraList()

```
virtual ~CameraList (
    void ) [virtual]
```

Virtual destructor.

#### 14.13.2.3 CameraList() [2/2]

```
CameraList (
    const CameraList & iface )
```

Copy constructor.

### 14.13.3 Member Function Documentation

#### 14.13.3.1 Append()

```
void Append (
    CameraList & otherList ) [virtual]
```

Appends a camera list to the current list.

##### Parameters

|                  |                                       |
|------------------|---------------------------------------|
| <i>otherList</i> | The other list to append to this list |
|------------------|---------------------------------------|

Implements [ICameraList](#).

#### 14.13.3.2 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](#).

#### 14.13.3.3 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](#).

#### 14.13.3.4 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](#).

#### 14.13.3.5 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**

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| <code>serialNumber</code> | The serial number of the camera object to retrieve |
|---------------------------|----------------------------------------------------|

**Returns**

A pointer to a camera object.

Implements [ICameraList](#).

**14.13.3.6 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](#).

**14.13.3.7 operator=( )**

```
CameraList& operator= (
    const CameraList & iface )
```

Assignment operator.

**14.13.3.8 operator[]( )**

```
CameraPtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [ICameraList](#).

**14.13.3.9 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](#).

**14.13.3.10 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 <a href="#">Camera</a> object |
|--------------|----------------------------------------------------------------|

Implements [ICameraList](#).

**14.13.3.11 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 <a href="#">Camera</a> object to remove |
|---------------------|------------------------------------------------------------------|

Implements [ICameraList](#).

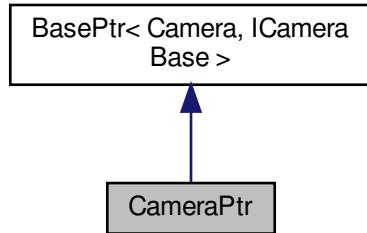
The documentation for this class was generated from the following file:

- [include/CameraList.h](#)

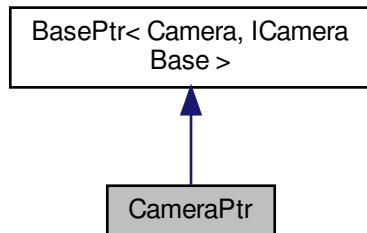
## 14.14 CameraPtr Class Reference

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

### 14.14.1 Detailed Description

A reference tracked pointer to a camera object.

## 14.14.2 Constructor & Destructor Documentation

### 14.14.2.1 CameraPtr() [1/4]

```
CameraPtr ( ) throw ( )    [inline]
```

Default constructor.

### 14.14.2.2 CameraPtr() [2/4]

```
CameraPtr ( const int ) throw ( )    [inline]
```

Default constructor.

### 14.14.2.3 CameraPtr() [3/4]

```
CameraPtr ( const long ) throw ( )    [inline]
```

Default constructor with argument.

### 14.14.2.4 CameraPtr() [4/4]

```
CameraPtr ( const std::nullptr_t ) throw ( )    [inline]
```

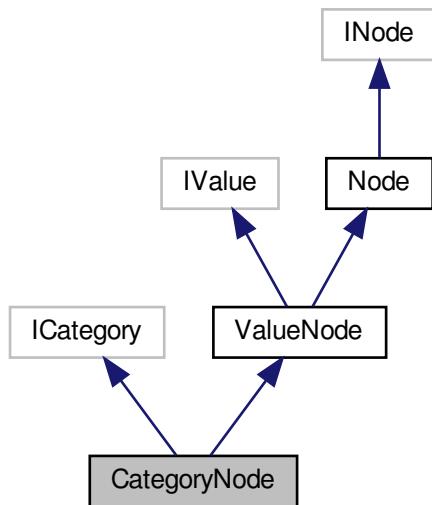
The documentation for this class was generated from the following file:

- [include/ CameraPtr.h](#)

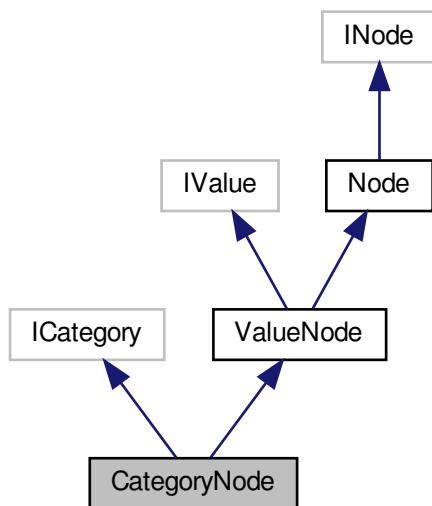
## 14.15 CategoryNode Class Reference

[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

### 14.15.1 Detailed Description

[Interface](#) for string properties.

### 14.15.2 Constructor & Destructor Documentation

#### 14.15.2.1 CategoryNode() [1/2]

```
CategoryNode ( )
```

#### 14.15.2.2 CategoryNode() [2/2]

```
CategoryNode ( std::shared_ptr< Node::NodeImpl > pCategory )
```

#### 14.15.2.3 ~CategoryNode()

```
virtual ~CategoryNode ( ) [virtual]
```

### 14.15.3 Member Function Documentation

### 14.15.3.1 GetFeatures()

```
virtual void GetFeatures (
    FeatureList_t & Features ) const [virtual]
```

Get all features of the category (including sub-categories)

### 14.15.3.2 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

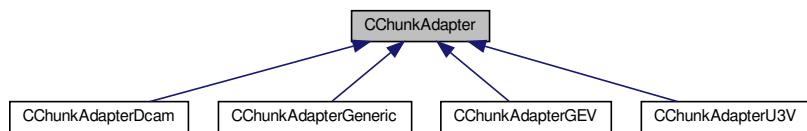
The documentation for this class was generated from the following file:

- include/SpinGenApi/[CategoryNode.h](#)

## 14.16 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](#)

### 14.16.1 Detailed Description

Connects a chunked buffer to a node map.

### 14.16.2 Constructor & Destructor Documentation

#### 14.16.2.1 ~CChunkAdapter()

```
virtual ~CChunkAdapter ( ) [virtual]
```

Destructor.

#### 14.16.2.2 CChunkAdapter()

```
CChunkAdapter (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 ) [protected]
```

Serves as default constructor.

### 14.16.3 Member Function Documentation

#### 14.16.3.1 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](#).

#### 14.16.3.2 AttachNodeMap()

```
void AttachNodeMap (
    INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

#### 14.16.3.3 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](#).

#### 14.16.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

#### 14.16.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

#### 14.16.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

#### 14.16.3.7 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

#### 14.16.4 Member Data Documentation

##### 14.16.4.1 m\_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

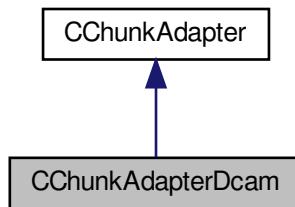
The documentation for this class was generated from the following file:

- include/SpinGenApi/ChunkAdapter.h

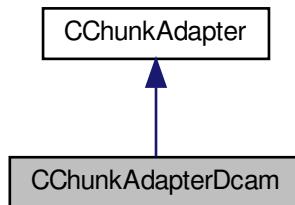
## 14.17 CChunkAdapterDcam Class Reference

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

### 14.17.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

### 14.17.2 Constructor & Destructor Documentation

#### 14.17.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

#### 14.17.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam () [virtual]
```

Destructor.

### 14.17.3 Member Function Documentation

#### 14.17.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

#### 14.17.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

#### 14.17.3.3 CheckCRC()

```
bool CheckCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

Checks CRC sum of buffer.

#### 14.17.3.4 HasCRC()

```
bool HasCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

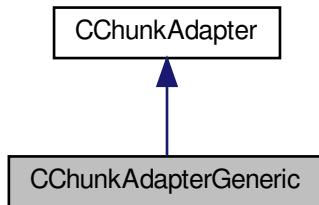
Checks if buffer has a CRC attached.

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

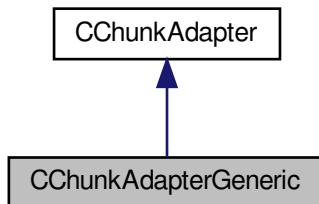
- include/SpinGenApi/[ChunkAdapterDcam.h](#)

## 14.18 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

#### 14.18.1 Constructor & Destructor Documentation

#### 14.18.1.1 CChunkAdapterGeneric()

```
CChunkAdapterGeneric (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

#### 14.18.1.2 ~CChunkAdapterGeneric()

```
virtual ~CChunkAdapterGeneric ( ) [virtual]
```

### 14.18.2 Member Function Documentation

#### 14.18.2.1 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](#).

#### 14.18.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkData_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

#### 14.18.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    SingleChunkDataStr_t * ChunkData,
    int64_t NumChunks,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

#### 14.18.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

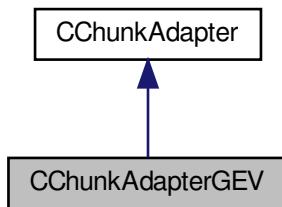
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

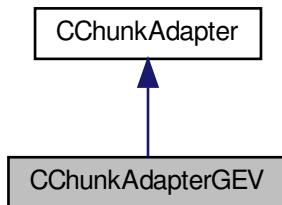
## 14.19 CChunkAdapterGEV Class Reference

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

### 14.19.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

### 14.19.2 Constructor & Destructor Documentation

#### 14.19.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

#### 14.19.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( ) [virtual]
```

Destructor.

### 14.19.3 Member Function Documentation

#### 14.19.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics\_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

#### 14.19.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

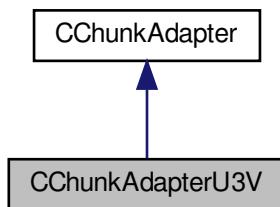
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGEV.h](#)

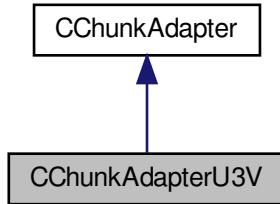
## 14.20 CChunkAdapterU3V Class Reference

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

### 14.20.1 Detailed Description

Connects a chunked U3V buffer to a node map.

### 14.20.2 Constructor & Destructor Documentation

#### 14.20.2.1 CChunkAdapterU3V()

```
CChunkAdapterU3V (
```

```
    INodeMap * pNodeMap = NULL,
```

```
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

#### 14.20.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

### 14.20.3 Member Function Documentation

#### 14.20.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

#### 14.20.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

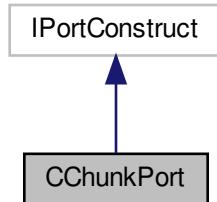
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

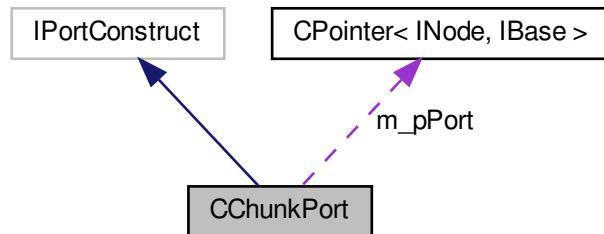
## 14.21 CChunkPort Class Reference

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](#)

### 14.21.1 Detailed Description

Port attachable to a chunk in a buffer.

### 14.21.2 Constructor & Destructor Documentation

#### 14.21.2.1 CChunkPort()

```
CChunkPort (
    IPort * pPort = NULL )
```

Constructor; can attach to a port.

### 14.21.2.2 ~CChunkPort()

```
~CChunkPort ( )
```

Destructor; detaches from the port.

## 14.21.3 Member Function Documentation

### 14.21.3.1 AttachChunk()

```
void AttachChunk (
    uint8_t * pBaseAddress,
    int64_t ChunkOffset,
    int64_t Length,
    bool Cache )
```

Attaches the Chunk to the ChunkPort.

### 14.21.3.2 AttachPort()

```
bool AttachPort (
    ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

### 14.21.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
    uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64\_t ID representation.

### 14.21.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
    uint8_t * pChunkIDBuffer,
    int ChunkIDLength )
```

Checks if a ChunkID matches.

**14.21.3.5 ClearCache()**

```
void ClearCache ( )
```

Clears the chunk cache.

**14.21.3.6 DetachChunk()**

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

**14.21.3.7 DetachPort()**

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

**14.21.3.8 GetAccessMode()**

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

**14.21.3.9 GetChunkIDLength()**

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

**14.21.3.10 GetPrincipalInterfaceType()**

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

**14.21.3.11 GetSwapEndianess()**

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

**14.21.3.12 InvalidateNode()**

```
void InvalidateNode ( )
```

**14.21.3.13 Read()**

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

**14.21.3.14 SetPortImpl()**

```
virtual void SetPortImpl (
    IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

**14.21.3.15 UpdateBuffer()**

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

**14.21.3.16 Write()**

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

#### 14.21.4 Member Data Documentation

##### 14.21.4.1 m\_pChunkPort

```
void* m_pChunkPort [protected]
```

##### 14.21.4.2 m\_pPort

```
CNodePtr m_pPort [protected]
```

##### 14.21.4.3 m\_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

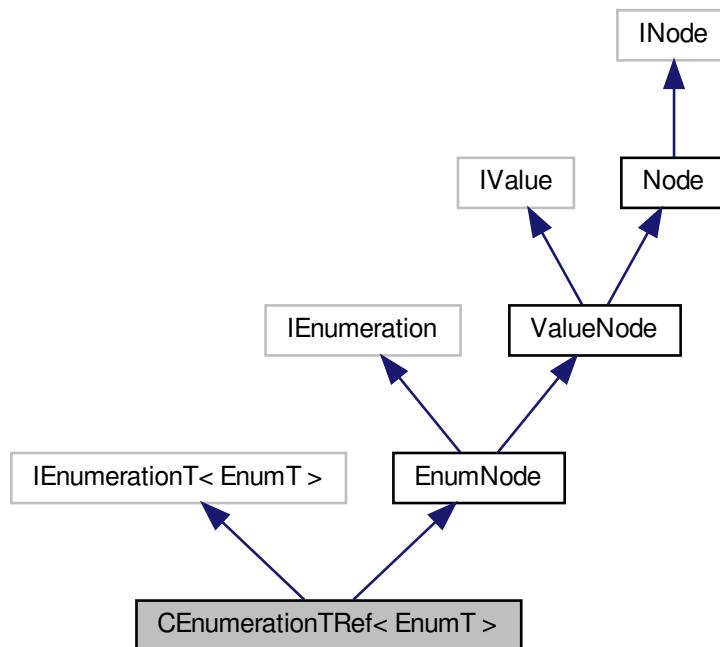
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkPort.h](#)

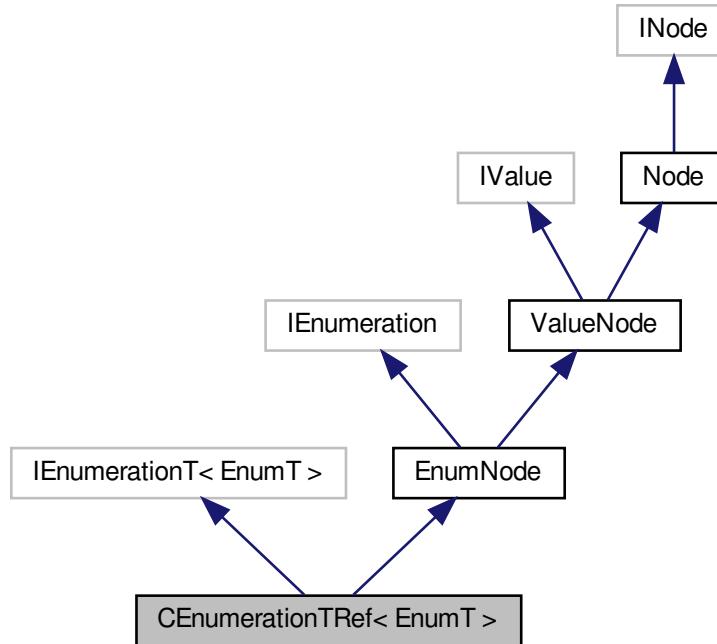
## 14.22 CEnumerationTRef< EnumT > Class Template Reference

[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 GenICam::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

### 14.22.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

### 14.22.2 Constructor & Destructor Documentation

#### 14.22.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

#### 14.22.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

#### 14.22.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( ) [virtual]
```

### 14.22.3 Member Function Documentation

### 14.22.3.1 GetCurrentEntry()

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

Get the current entry.

Reimplemented from [EnumNode](#).

### 14.22.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
    const EnumT Value ) [virtual]
```

returns the EnumEntry object belonging to the Value

### 14.22.3.3 GetEntry() [2/2]

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

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

### 14.22.3.4 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

**14.22.3.5 operator()()**

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

**14.22.3.6 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](#).

**14.22.3.7 operator=( ) [2/2]**

```
virtual IEnumeration& operator= (
    EnumT Value ) [virtual]
```

Set node value.

**14.22.3.8 SetEnumReference()**

```
virtual void SetEnumReference (
    int Index,
    GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

**14.22.3.9 SetNumEnums()**

```
virtual void SetNumEnums (
    int NumEnums ) [virtual]
```

sets the number of enum values

**14.22.3.10 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

**14.22.3.11 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) |

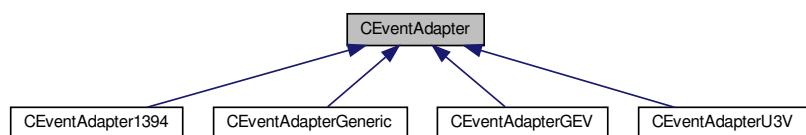
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumNodeT.h](#)

## 14.23 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](#)

### 14.23.1 Detailed Description

Delivers Events to ports.

### 14.23.2 Constructor & Destructor Documentation

#### 14.23.2.1 CEventAdapter()

```
CEventAdapter (
    INodeMap * pNodeMap = NULL )
```

Constructor.

#### 14.23.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( ) [virtual]
```

Destructor.

### 14.23.3 Member Function Documentation

#### 14.23.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
    INodeMap * pNodeMap ) [virtual]
```

Attaches to a node map and retrieves the chunk ports.

### 14.23.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [pure virtual]
```

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

### 14.23.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( ) [virtual]
```

Detaches from the node emap.

## 14.23.4 Member Data Documentation

### 14.23.4.1 m\_pEventAdapter

```
void* m_pEventAdapter [protected]
```

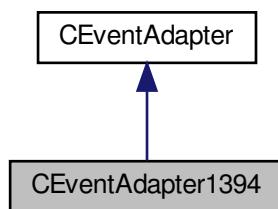
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter.h](#)

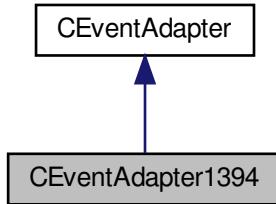
## 14.24 CEventAdapter1394 Class Reference

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

### 14.24.1 Detailed Description

Distribute the events to the node map.

### 14.24.2 Constructor & Destructor Documentation

#### 14.24.2.1 CEventAdapter1394()

```
CEventAdapter1394 (
    INodeMap * pNodeMap = NULL ) [explicit]
```

constructor

#### 14.24.2.2 ~CEventAdapter1394()

```
virtual ~CEventAdapter1394 ( ) [virtual]
```

### 14.24.3 Member Function Documentation

#### 14.24.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    EventData1394 & Event,
    uint32_t numBytes )
```

distributes events to node map

#### 14.24.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

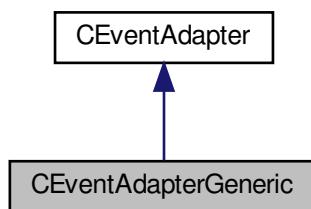
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

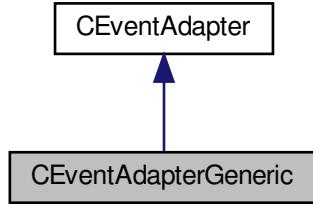
## 14.25 CEventAdapterGeneric Class Reference

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

### 14.25.1 Detailed Description

Connects a generic event to a node map.

### 14.25.2 Constructor & Destructor Documentation

#### 14.25.2.1 CEventAdapterGeneric()

```
CEventAdapterGeneric (
    INodeMap * pNodeMap = NULL )
```

Constructor.

### 14.25.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

## 14.25.3 Member Function Documentation

### 14.25.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

### 14.25.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes,
    const GenICam::gcstring & EventID ) [virtual]
```

### 14.25.3.3 DeliverMessage() [3/3]

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes,
    uint64_t EventID ) [virtual]
```

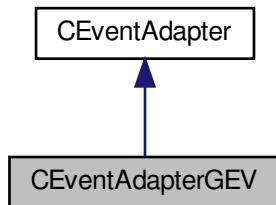
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGeneric.h](#)

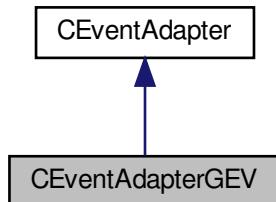
## 14.26 CEventAdapterGEV Class Reference

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 DeliverEventDataMessage (const GVCP_EVENTDATA_REQUEST *pEventData)`  
*Delivers the Event + Data listed in the EventData packet.*

## Additional Inherited Members

### 14.26.1 Detailed Description

Connects a GigE Event to a node map.

### 14.26.2 Constructor & Destructor Documentation

#### 14.26.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (   
    INodeMap * pNodeMap = NULL )
```

Constructor.

#### 14.26.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( ) [virtual]
```

Destructor.

### 14.26.3 Member Function Documentation

#### 14.26.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (   
    const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the Event packet.

#### 14.26.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (   
    const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the Event + Data listed in the EventData packet.

#### 14.26.3.3 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[ ],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

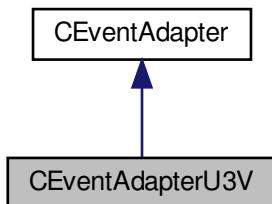
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

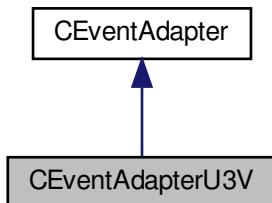
## 14.27 CEventAdapterU3V Class Reference

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

### 14.27.1 Detailed Description

Connects a U3V Event to a node map.

### 14.27.2 Constructor & Destructor Documentation

#### 14.27.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
    INodeMap * pNodeMap = NULL )
```

Constructor.

#### 14.27.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( ) [virtual]
```

Destructor.

### 14.27.3 Member Function Documentation

#### 14.27.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    const U3V_EVENT_MESSAGE * pEventMessage )
```

Delivers the Event + Data listed in the packet.

### 14.27.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

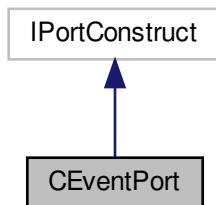
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

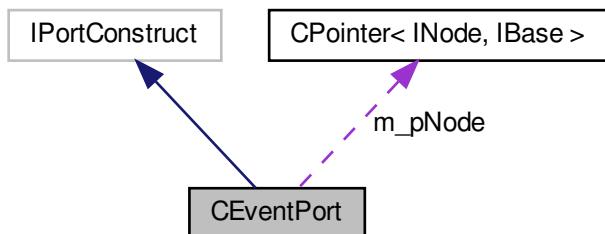
## 14.28 CEventPort Class Reference

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 `GetSwapEndianess ()`  
*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`

### 14.28.1 Detailed Description

Port attachable to an event.

### 14.28.2 Constructor & Destructor Documentation

#### 14.28.2.1 CEventPort()

```
CEventPort (   
    INode * pNode = NULL )
```

Constructor; can attach to a node.

#### 14.28.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

### 14.28.3 Member Function Documentation

#### 14.28.3.1 AttachEvent()

```
void AttachEvent (   
    uint8_t * pBaseAddress,   
    int64_t Length )
```

Attaches the an Event to the EventPort.

#### 14.28.3.2 AttachNode()

```
bool AttachNode (   
    ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the [Node](#).

#### 14.28.3.3 CheckEventID() [1/2]

```
bool CheckEventID (   
    uint64_t EventID )
```

Checks if a EventID matches, version using uint64\_t ID representation.

**14.28.3.4 CheckEventID() [2/2]**

```
bool CheckEventID (
    uint8_t * pEventIDBuffer,
    int EventIDLength )
```

Checks if a EventID matches.

**14.28.3.5 DetachEvent()**

```
void DetachEvent ( )
```

Detaches the Event from the EventPort.

**14.28.3.6 DetachNode()**

```
void DetachNode ( )
```

Detaches from the [Node](#).

**14.28.3.7 GetAccessMode()**

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

**14.28.3.8 GetEventIDLength()**

```
int GetEventIDLength ( )
```

Gets the EventID length.

**14.28.3.9 GetPrincipalInterfaceType()**

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

#### 14.28.3.10 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

#### 14.28.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

#### 14.28.3.12 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

#### 14.28.3.13 SetPortImpl()

```
virtual void SetPortImpl (
    ::Spinnaker::GenApi::IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

#### 14.28.3.14 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

### 14.28.4 Member Data Documentation

#### 14.28.4.1 m\_pEventPort

```
void* m_pEventPort [protected]
```

#### 14.28.4.2 m\_pNode

```
CNodePtr m_pNode [protected]
```

#### 14.28.4.3 m\_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

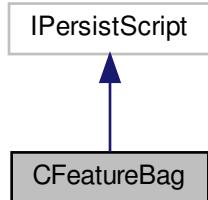
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventPort.h](#)

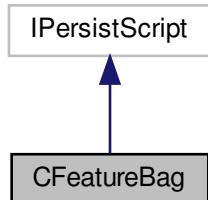
## 14.29 CFeatureBag Class Reference

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 ()`

### 14.29.1 Detailed Description

Bag holding streamable features of a nodetree.

### 14.29.2 Constructor & Destructor Documentation

#### 14.29.2.1 CFeatureBag()

```
CFeatureBag ( )
```

#### 14.29.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( ) [virtual]
```

### 14.29.3 Member Function Documentation

#### 14.29.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

#### 14.29.3.2 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.

**14.29.3.3 operator==( )**

```
bool operator== (
    const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

**14.29.3.4 PersistFeature()**

```
virtual void PersistFeature (
    IValue & item ) [virtual]
```

Stores a feature.

**14.29.3.5 SetInfo()**

```
virtual void SetInfo (
    GenICam::gestring & Info ) [virtual]
```

sets information about the node map

**14.29.3.6 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

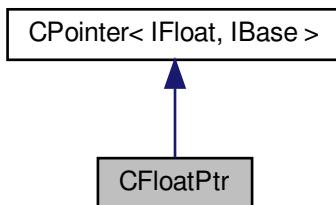
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Persistence.h](#)

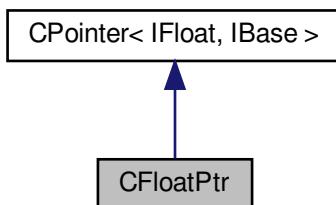
## 14.30 CFloatPtr Class Reference

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.*
- [IInteger \\* GetIntAlias \(\)](#)  
*gets the interface of an integer alias node.*
- [IEnumerator \\* GetEnumAlias \(\)](#)  
*gets the interface of an enum alias node.*

## Additional Inherited Members

### 14.30.1 Detailed Description

SmartPointer for IFloat interface pointer.

### 14.30.2 Constructor & Destructor Documentation

#### 14.30.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw ( )    [inline]
```

Default constructor.

#### 14.30.2.2 CFloatPtr() [2/2]

```
CFloatPtr ( 
    IBase * pB )    [inline]
```

Constructor from IBase pointer type.

### 14.30.3 Member Function Documentation

#### 14.30.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )    [inline]
```

gets the interface of an enum alias node.

#### 14.30.3.2 GetIntAlias()

```
IIInteger* GetIntAlias ( )    [inline]
```

gets the interface of an integer alias node.

#### 14.30.3.3 operator=( )

```
void operator= (
    IBase * pB ) [inline]
```

Assign IBase Pointer.

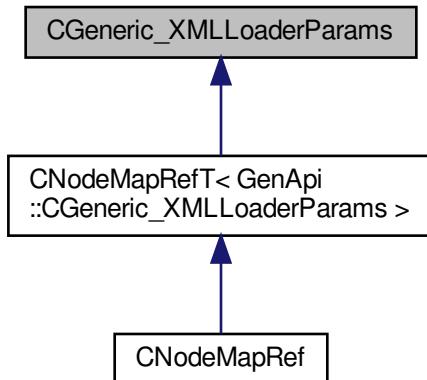
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

### 14.31 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 \(GenApi::INodeMap \\*\)](#)

#### 14.31.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

#### 14.31.2 Member Function Documentation

### 14.31.2.1 \_Initialize()

```
virtual void _Initialize (
    GenApi::INodeMap * ) [inline], [protected], [virtual]
```

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

- include/SpinGenApi/NodeMapRef.h

## 14.32 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](#)

### 14.32.1 Detailed Description

Named global lock which can be used over process boundaries.

### 14.32.2 Constructor & Destructor Documentation

#### 14.32.2.1 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.

#### 14.32.2.2 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.

#### 14.32.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

### 14.32.3 Member Function Documentation

#### 14.32.3.1 IsValid()

```
bool IsValid (
    void ) const
```

tests whether the lock is valid

#### 14.32.3.2 Lock()

```
bool Lock (
    unsigned int timeout_ms )
```

enters the lock (may block)

#### 14.32.3.3 TryLock()

```
bool TryLock (
    void )
```

tries to enter the lock and returns immediately when not possible

#### 14.32.3.4 Unlock()

```
void Unlock (
    void )
```

leaves the lock

### 14.32.4 Member Data Documentation

#### 14.32.4.1 m\_DebugCount

```
long m_DebugCount [mutable], [protected]
```

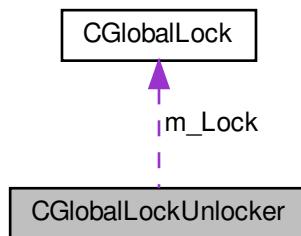
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

## 14.33 CGlobalLockUnlocker Class Reference

Unlocks the global lock object on destruction.

Collaboration diagram for CGlobalLockUnlocker:



## Public Member Functions

- `CGlobalLockUnlocker (CGlobalLock &lock)`
- `~CGlobalLockUnlocker ()`
- `void UnlockEarly (void)`

*This function allows to unlock the object early before the object is destroyed.*

## Protected Attributes

- `CGlobalLock & m_Lock`
- `bool m_enabled`

### 14.33.1 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 contructors

### 14.33.2 Constructor & Destructor Documentation

#### 14.33.2.1 CGlobalLockUnlocker()

```
CGlobalLockUnlocker (
    CGlobalLock & lock ) [inline]
```

#### 14.33.2.2 ~CGlobalLockUnlocker()

```
~CGlobalLockUnlocker ( ) [inline]
```

### 14.33.3 Member Function Documentation

#### 14.33.3.1 UnlockEarly()

```
void UnlockEarly (
    void ) [inline]
```

This function allows to unlock the object early before the object is destroyed.

#### 14.33.4 Member Data Documentation

##### 14.33.4.1 m\_enabled

```
bool m_enabled [protected]
```

##### 14.33.4.2 m\_Lock

```
CGlobalLock& m_Lock [protected]
```

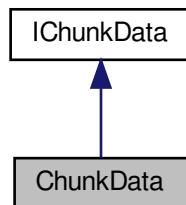
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

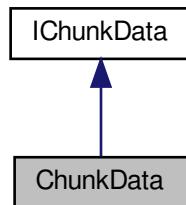
## 14.34 ChunkData Class Reference

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 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.

## Additional Inherited Members

### 14.34.1 Detailed Description

The chunk data which contains additional information about an image.

### 14.34.2 Constructor & Destructor Documentation

#### 14.34.2.1 ChunkData() [1/2]

`ChunkData ()`

#### 14.34.2.2 ChunkData() [2/2]

```
ChunkData (
    const ChunkData & src )
```

#### 14.34.2.3 ~ChunkData()

```
virtual ~ChunkData (
    void ) [virtual]
```

### 14.34.3 Member Function Documentation

#### 14.34.3.1 GetBlackLevel()

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.2 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](#).

#### 14.34.3.3 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.4 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](#).

#### 14.34.3.5 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](#).

#### 14.34.3.6 GetExposureTime()

```
float64_t GetExposureTime ( ) const [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.7 GetFrameID()

```
int64_t GetFrameID ( ) const [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.8 GetGain()

```
float64_t GetGain ( ) const [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.9 GetHeight()

```
int64_t GetHeight ( ) const [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.10 GetImage()

```
int64_t GetImage ( ) const [virtual]
```

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

#### 14.34.3.11 GetInferenceBoundingBoxResult()

```
InferenceBoundingBoxResult GetInferenceBoundingBoxResult ( ) const [virtual]
```

Description: Returns the chunk inference bounding box result data.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.12 GetInferenceConfidence()**

```
float64_t GetInferenceConfidence ( ) const [virtual]
```

Description: Returns the chunk data inference confidence percentage.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.13 GetInferenceFrameId()**

```
int64_t GetInferenceFrameId ( ) const [virtual]
```

Description: Returns the frame ID associated with the most recent inference result.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.14 GetInferenceResult()**

```
int64_t GetInferenceResult ( ) const [virtual]
```

Description: Returns the chunk data inference result.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.15 GetLinePitch()**

```
int64_t GetLinePitch ( ) const [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.16 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](#).

**14.34.3.17 GetOffsetX()**

```
int64_t GetOffsetX ( ) const [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

**14.34.3.18 GetOffsetY()**

```
int64_t GetOffsetY ( ) const [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

**14.34.3.19 GetPartSelector()**

```
int64_t GetPartSelector ( ) const [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.20 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](#).

**14.34.3.21 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](#).

**14.34.3.22 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](#).

**14.34.3.23 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](#).

#### 14.34.3.24 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](#).

#### 14.34.3.25 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](#).

#### 14.34.3.26 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](#).

#### 14.34.3.27 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.28 GetScan3dTransformValue()**

```
float64_t GetScan3dTransformValue ( ) const [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.29 GetScanLineSelector()**

```
int64_t GetScanLineSelector ( ) const [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.30 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](#).

**14.34.3.31 GetSerialDataLength()**

```
int64_t GetSerialDataLength ( ) const [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

**14.34.3.32 GetStreamChannelID()**

```
int64_t GetStreamChannelID ( ) const [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.33 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](#).

**14.34.3.34 GetTimestamp()**

```
int64_t GetTimestamp ( ) const [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

**14.34.3.35 GetTimestampLatchValue()**

```
int64_t GetTimestampLatchValue ( ) const [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.36 GetTransferBlockID()**

```
int64_t GetTransferBlockID ( ) const [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.37 GetTransferQueueCurrentBlockCount()**

```
int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

**14.34.3.38 GetWidth()**

```
int64_t GetWidth ( ) const [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

**14.34.3.39 SetChunks()**

```
void SetChunks (
    GenApi::INodeMap & pNodeMap ) [virtual]
```

Implements [IChunkData](#).

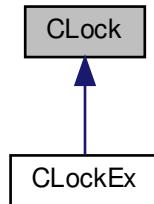
The documentation for this class was generated from the following file:

- [include/ChunkData.h](#)

## 14.35 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*

#### 14.35.1 Detailed Description

A lock class.

#### 14.35.2 Constructor & Destructor Documentation

##### 14.35.2.1 CLock()

[CLock \(\)](#)

Constructor.

#### 14.35.2.2 ~CLock()

```
~CLock ( )
```

Destructor.

### 14.35.3 Member Function Documentation

#### 14.35.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

#### 14.35.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

#### 14.35.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

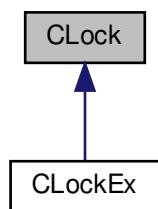
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

## 14.36 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`

### 14.36.1 Detailed Description

A lock class.

### 14.36.2 Constructor & Destructor Documentation

#### 14.36.2.1 `CLock()` [1/2]

`CLock ()`

Constructor.

#### 14.36.2.2 `CLock()` [2/2]

```
CLock (
    void * pLock )
```

Constructor.

#### 14.36.2.3 ~CLock()

```
~CLock ( )
```

Destructor.

### 14.36.3 Member Function Documentation

#### 14.36.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

#### 14.36.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

#### 14.36.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

### 14.36.4 Friends And Related Function Documentation

#### 14.36.4.1 NodeMap

```
friend class NodeMap [friend]
```

### 14.36.5 Member Data Documentation

#### 14.36.5.1 m\_bOwnLock

```
bool m_bOwnLock [protected]
```

#### 14.36.5.2 m\_lock

```
void* m_lock [protected]
```

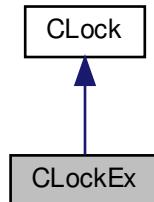
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

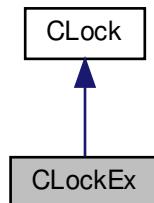
### 14.37 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



## Additional Inherited Members

### 14.37.1 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

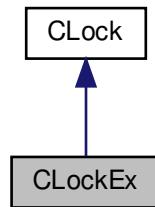
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

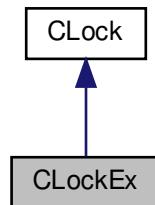
## 14.38 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



## Protected Attributes

- void \* [m\\_lockEx](#)

## Additional Inherited Members

### 14.38.1 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

### 14.38.2 Member Data Documentation

#### 14.38.2.1 [m\\_lockEx](#)

```
void* m_lockEx [protected]
```

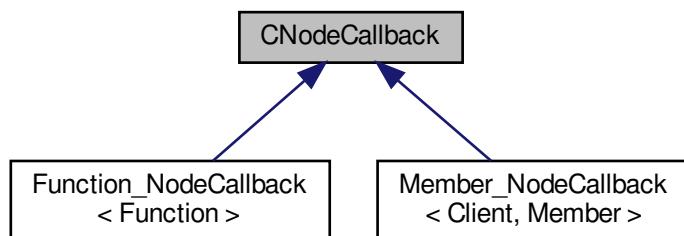
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Synch.h](#)

## 14.39 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 the 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*

### 14.39.1 Detailed Description

callback body instance for INode pointers

### 14.39.2 Constructor & Destructor Documentation

#### 14.39.2.1 CNodeCallback()

```
CNodeCallback (
    INode * pNode,
    ECallbackType CallbackType ) [inline]
```

#### 14.39.2.2 ~CNodeCallback()

```
virtual ~CNodeCallback ( ) [inline], [virtual]
```

virtual destructor

### 14.39.3 Member Function Documentation

#### 14.39.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in [Member\\_NodeCallback< Client, Member >](#), and [Function\\_NodeCallback< Function >](#).

#### 14.39.3.2 GetCallbackType()

```
ECallbackType GetCallbackType ( ) [inline]
```

#### 14.39.3.3 GetNode()

```
INode* GetNode ( ) [inline]
```

returns the node the callback is registered to

#### 14.39.3.4 operator()()

```
virtual void operator() ( ECallbackType CallbackType ) const [pure virtual]
```

fires the callback if the type is right

Implemented in [Member\\_NodeCallback< Client, Member >](#), and [Function\\_NodeCallback< Function >](#).

### 14.39.4 Member Data Documentation

#### 14.39.4.1 m\_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

#### 14.39.4.2 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

## 14.40 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 GenlCam::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 GenlCam::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 GenlCam::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 CNodeDataMap (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 \(const GenICam::gcstring &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true\)](#)  
*Creates a node map from the preprocessed memory internal representation of the camera description file(s).*
- [INodeMap \\* CreateNodeMap \(CLOCK &UserProvidedLock, const GenICam::gcstring &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true\)](#)  
*Creates a node map from the preprocessed memory internal representation of the camera description file(s).*
- void [GetSupportedSchemaVersions \(GenICam::gcstring\\_vector &SchemaVersions\) const](#)
- [GenICam::gcstring ToString \(\) const](#)  
*Outputs the pre-processed node map in string form (for debug purpose)*
- [GenICam::gcstring ToXml \(\) const](#)  
*Outputs the pre-processed node map in XML form (mainly for debug purpose)*
- void [GetNodeStatistics \(NodeStatistics\\_t &NodeStatistics\)](#)
- const [GenICam::gcstring ApplyStyleSheet \(const GenICam::gcstring &StyleSheetFileName\)](#)  
*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 \(INodeMap \\*pNodeMap\)](#)

### 14.40.1 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.

## 14.40.2 Constructor & Destructor Documentation

### 14.40.2.1 CNodeMapFactory() [1/5]

```
CNodeMapFactory( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

### 14.40.2.2 ~CNodeMapFactory()

```
virtual ~CNodeMapFactory( ) [virtual]
```

Destroys the node map factory data if all references to the data have been released.

### 14.40.2.3 CNodeMapFactory() [2/5]

```
CNodeMapFactory(
    const CNodeMapFactory& )
```

Creates another reference to the node map factory data.

No data is copied.

### 14.40.2.4 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.

**14.40.2.5 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.

#### 14.40.2.6 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.<br><code>gcstring cfdData; //... fill cfdData ...<br/> CNodeMapFactory factory(ContentType_Xml, cfdData.c_str(),<br/> cfmData.size());<br/> // Create the node map. The node map can be destroyed using the<br/> IDestroy interface<br/> later. INodeMap* pNodeMap = factory.CreateNodeMap();<br/> // The next step is attaching the device port (not shown).</code> |
| 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.

### 14.40.3 Member Function Documentation

#### 14.40.3.1 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.

#### 14.40.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
    const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

#### 14.40.3.3 ClearCache()

```
static bool ClearCache ( ) [static]
```

Deletes all preprocessed camera description files from the cache.

#### 14.40.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( ) [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

#### 14.40.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
    INodeMap * pNodeMap ) [static]
```

#### 14.40.3.6 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.

#### 14.40.3.7 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.

#### 14.40.3.8 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. <a href="#">Preprocess()</a> 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. |

#### 14.40.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
    NodeStatistics_t & NodeStatistics )
```

#### 14.40.3.10 GetSupportedSchemaVersions()

```
void GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) const
```

Each list entry is a string with the format "{Major}.{Minor}" were {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

#### 14.40.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

#### 14.40.3.12 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.

#### 14.40.3.13 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).

#### 14.40.3.14 IsPreprocessed()

```
bool IsPreprocessed ( ) const
```

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

#### 14.40.3.15 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.

#### 14.40.3.16 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.

#### 14.40.3.17 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.

**14.40.3.18 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.

**14.40.3.19 ToString()**

```
GenICam::gcstring ToString( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

**14.40.3.20 ToXml()**

```
GenICam::gcstring ToXml( ) const
```

Outputs the pre-processed node map in XML form (mainly for debug purpose)

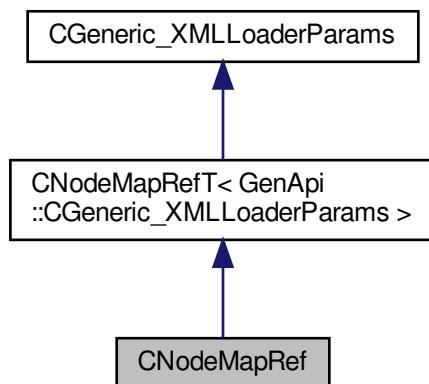
The documentation for this class was generated from the following file:

- include/SpinGenApi/NodeMapFactory.h

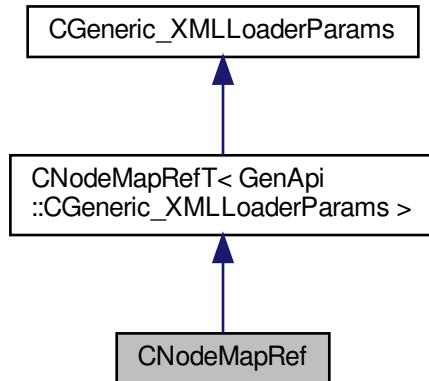
## 14.41 CNodeMapRef Class Reference

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

### 14.41.1 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.

### 14.41.2 Constructor & Destructor Documentation

**14.41.2.1 CNodeMapRef() [1/3]**

```
CNodeMapRef (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

**14.41.2.2 CNodeMapRef() [2/3]**

```
CNodeMapRef (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

**14.41.2.3 CNodeMapRef() [3/3]**

```
CNodeMapRef (
    const CNodeMapRef & Them ) [inline]
```

Copy constructor.

**14.41.3 Member Function Documentation****14.41.3.1 operator=() [1/2]**

```
CNodeMapRef& operator= (
    const CNodeMapRef & Them ) [inline]
```

Assignment.

**14.41.3.2 operator=() [2/2]**

```
CNodeMapRef& operator= (
    INodeMap * pNodeMap ) [inline]
```

Assignment of an INodeMap\*.

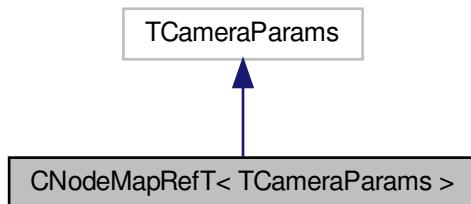
The documentation for this class was generated from the following file:

- include/SpinGenApi/NodeMapRef.h

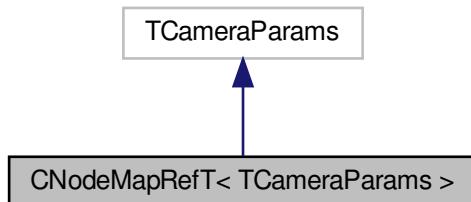
## 14.42 CNodeMapRefT< TCameraParams > Class Template Reference

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 &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)  
*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](#).*

### 14.42.1 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) |
|----------------------|---------------------------------------------------------------------------|

**14.42.2 Constructor & Destructor Documentation****14.42.2.1 CNodeMapRefT() [1/3]**

```
CNodeMapRefT (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

**14.42.2.2 CNodeMapRefT() [2/3]**

```
CNodeMapRefT (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

**14.42.2.3 CNodeMapRefT() [3/3]**

```
CNodeMapRefT (
    const CNodeMapRefT< TCameraParams > & Them )
```

Copy constructor.

**14.42.2.4 ~CNodeMapRefT()**

```
~CNodeMapRefT [inline], [virtual]
```

Destructor.

**14.42.3 Member Function Documentation**

#### 14.42.3.1 `_ClearXMLCache()`

```
static bool _ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

#### 14.42.3.2 `_Connect() [1/2]`

```
virtual bool _Connect (
    IPort * pPort ) [virtual]
```

Connects a port to the standard port "Device".

#### 14.42.3.3 `_Connect() [2/2]`

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

Connects a port to a port node with given name.

#### 14.42.3.4 `_Destroy()`

```
void _Destroy [inline]
```

Destroys the node map.

#### 14.42.3.5 `_GetDeviceName()`

```
virtual GenICam::gcstring _GetDeviceName ( ) [virtual]
```

Get device name.

#### 14.42.3.6 `_GetNode()`

```
virtual INode* _GetNode (
    const GenICam::gcstring & key ) [virtual]
```

Retrieves the node from the central map by name.

#### 14.42.3.7 `_GetNodes()`

```
virtual void _GetNodes (
    NodeList_t & Nodes ) [virtual]
```

Retrieves all nodes in the node map.

#### 14.42.3.8 `_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}" were {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

#### 14.42.3.9 `_InvalidateNodes()`

```
virtual void _InvalidateNodes ( ) [virtual]
```

Invalidates all nodes.

#### 14.42.3.10 `_LoadXMLFromFile()`

```
void _LoadXMLFromFile (
    const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

#### 14.42.3.11 `_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.

**14.42.3.12 \_LoadXMLFromString()**

```
void _LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

**14.42.3.13 \_LoadXMLFromStringInject()**

```
void _LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData const,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

**14.42.3.14 \_LoadXMLFromZIPData()**

```
void _LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

**14.42.3.15 \_LoadXMLFromZIPFile()**

```
void _LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

**14.42.3.16 \_Poll()**

```
virtual void _Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

#### 14.42.3.17 operator=() [1/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    const CNodeMapRefT< TCameraParams > & Them )
```

Assignment.

#### 14.42.3.18 operator=() [2/2]

```
CNodeMapRefT< TCameraParams > & operator= (
    INodeMap * pNodeMap )
```

Assignment of an INodeMap\*.

### 14.42.4 Member Data Documentation

#### 14.42.4.1 \_Ptr

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

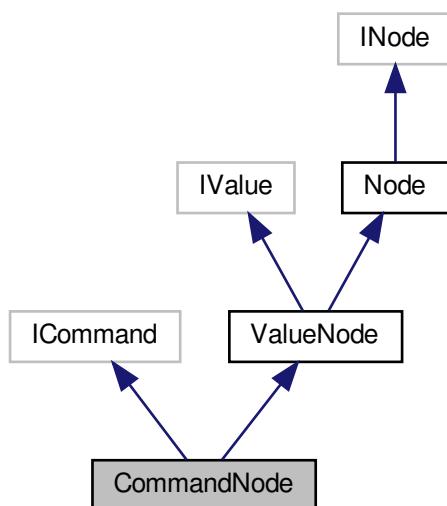
The documentation for this class was generated from the following file:

- include/SpinGenApi/[NodeMapRef.h](#)

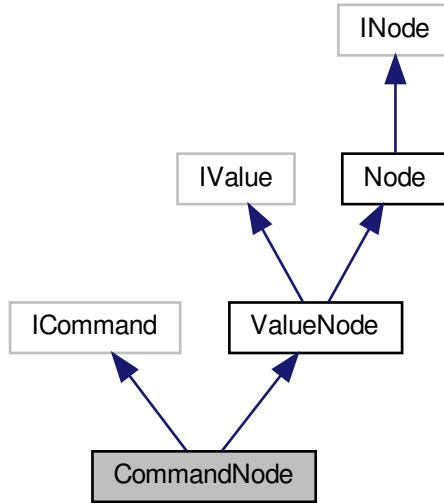
## 14.43 CommandNode Class Reference

[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

### 14.43.1 Detailed Description

[Interface](#) for string properties.

### 14.43.2 Constructor & Destructor Documentation

**14.43.2.1 CommandNode() [1/2]**

```
CommandNode ( )
```

**14.43.2.2 CommandNode() [2/2]**

```
CommandNode (
    std::shared_ptr< Node::NodeImpl > pCommand )
```

**14.43.2.3 ~CommandNode()**

```
virtual ~CommandNode ( ) [virtual]
```

**14.43.3 Member Function Documentation****14.43.3.1 Execute()**

```
virtual void Execute (
    bool Verify = true ) [virtual]
```

Execute the command.

**Parameters**

|               |                                                            |
|---------------|------------------------------------------------------------|
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |
|---------------|------------------------------------------------------------|

**14.43.3.2 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

### 14.43.3.3 operator()()

```
virtual void operator() () [virtual]
```

Execute the command.

### 14.43.3.4 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](#)

## 14.44 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 \(\)](#)

### 14.44.1 Detailed Description

Definition of a simple [Counter](#) class.

## 14.44.2 Constructor & Destructor Documentation

### 14.44.2.1 Counter()

```
Counter ( ) [inline]
```

## 14.44.3 Member Function Documentation

### 14.44.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

### 14.44.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

### 14.44.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

### 14.44.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

### 14.44.3.5 operator++() [2/2]

```
unsigned int operator++ (
    int ) [inline]
```

**14.44.3.6 operator--() [1/2]**

```
unsigned int operator-- ( ) [inline]
```

**14.44.3.7 operator--() [2/2]**

```
unsigned int operator-- (
    int ) [inline]
```

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

- include/SpinGenApi/Counter.h

## 14.45 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.*

### 14.45.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a `GenApi` pointer dealing with the `dynamic_cast` automatically.

### 14.45.2 Constructor & Destructor Documentation

#### 14.45.2.1 CPointer() [1/2]

```
CPointer (
    void ) [inline]
```

Default constructor.

#### 14.45.2.2 CPointer() [2/2]

```
CPointer (
    B * pB ) [inline]
```

Constructor from `INode` pointer type.

#### 14.45.2.3 ~CPointer()

```
virtual ~CPointer (
    void ) [inline], [virtual]
```

### 14.45.3 Member Function Documentation

#### 14.45.3.1 IsValid()

```
bool IsValid ( ) const throw ( ) [inline]
```

true if the pointer is valid

#### 14.45.3.2 operator bool()

```
operator bool (
    void ) const throw ( ) [inline]
```

true if the pointer is valid

#### 14.45.3.3 operator T\*()

```
operator T* (
    void ) const [inline]
```

Dereferencing.

#### 14.45.3.4 operator"!=() [1/5]

```
bool operator!= (
    const CPointer< T, B > & rT ) const [inline]
```

pointer unequal

#### 14.45.3.5 operator"!=() [2/5]

```
bool operator!= (
    const int nMustBeNull ) const [inline]
```

pointer unequal

**14.45.3.6 operator"!=() [3/5]**

```
bool operator!= (
    const long int nMustBeNull ) const [inline]  
pointer unequal
```

**14.45.3.7 operator"!=() [4/5]**

```
bool operator!= (
    const std::nullptr_t nullPtr ) const [inline]  
pointer unequal
```

**14.45.3.8 operator"!=() [5/5]**

```
bool operator!= (
    T * pT ) const [inline]  
pointer unequal
```

**14.45.3.9 operator()()**

```
T& operator() (
    void ) const [inline]
```

Dereferencing.

**14.45.3.10 operator\*()**

```
T& operator* (
    void ) const [inline]
```

Dereferencing.

**14.45.3.11 operator->()**

```
T* operator-> (
    void ) const [inline]
```

Dereferencing.

**14.45.3.12 operator=( )**

```
void operator= (
    B * pB )  [inline]
```

Assign INode Pointer.

**14.45.3.13 operator==( ) [1/3]**

```
bool operator== (
    const CPointer< T, B > & rT ) const [inline]
```

pointer equal

**14.45.3.14 operator==( ) [2/3]**

```
bool operator== (
    int nMustBeNull ) const [inline]
```

pointer equal

**14.45.3.15 operator==( ) [3/3]**

```
bool operator== (
    T * pT ) const [inline]
```

pointer equal

**14.45.4 Member Data Documentation****14.45.4.1 m\_pT**

```
T* m_pT [protected]
```

Underlying raw pointer.

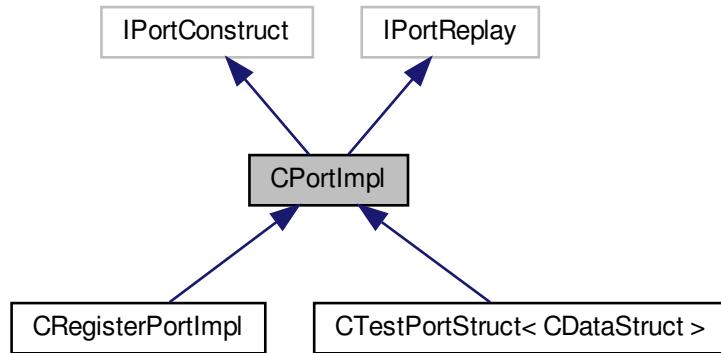
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

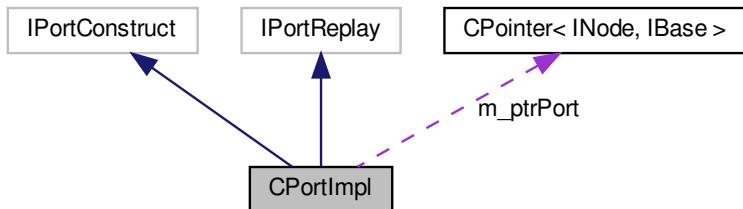
## 14.46 CPortImpl Class Reference

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 [GetSwapEndianess \(\)](#)  
*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.*

### 14.46.1 Detailed Description

Standard implementation for a port.

### 14.46.2 Constructor & Destructor Documentation

#### 14.46.2.1 CPortImpl()

[CPortImpl \( \) \[inline\]](#)

Constructor.

#### 14.46.2.2 ~CPortImpl()

[virtual ~CPortImpl \( \) \[inline\], \[virtual\]](#)

Destructor.

### 14.46.3 Member Function Documentation

#### 14.46.3.1 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 >](#).

#### 14.46.3.2 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

#### 14.46.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```

#### 14.46.3.4 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 >](#).

#### 14.46.3.5 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.

#### 14.46.3.6 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

#### 14.46.3.7 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 >](#).

#### 14.46.4 Member Data Documentation

##### 14.46.4.1 m\_ptrPort

```
CNodePtr m_ptrPort [protected]
```

Pointer to the node holding a reference to this implementation.

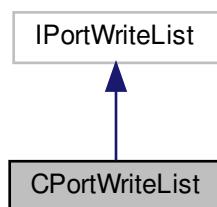
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortImpl.h](#)

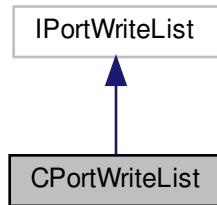
## 14.47 CPortWriteList Class Reference

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`

### 14.47.1 Detailed Description

Container holding a list of port write commands.

### 14.47.2 Constructor & Destructor Documentation

#### 14.47.2.1 CPortWriteList()

`CPortWriteList ( )`

Constructor.

#### 14.47.2.2 ~CPortWriteList()

```
~CPortWriteList ( )
```

Destructor.

### 14.47.3 Member Function Documentation

#### 14.47.3.1 GetCookie()

```
virtual int64_t GetCookie ( ) [virtual]
```

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

#### 14.47.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ( )
```

#### 14.47.3.3 Replay()

```
virtual void Replay (
    IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

#### 14.47.3.4 SetCookie()

```
virtual void SetCookie (
    const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

#### 14.47.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

#### 14.47.4 Member Data Documentation

##### 14.47.4.1 m\_pWriteList

```
void* m_pWriteList [protected]
```

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

- include/SpinGenApi/[PortWriteList.h](#)

### 14.48 CpuUsageInfo Struct Reference

#### Public Attributes

- bool [dummy](#)

##### 14.48.1 Member Data Documentation

###### 14.48.1.1 dummy

```
bool dummy
```

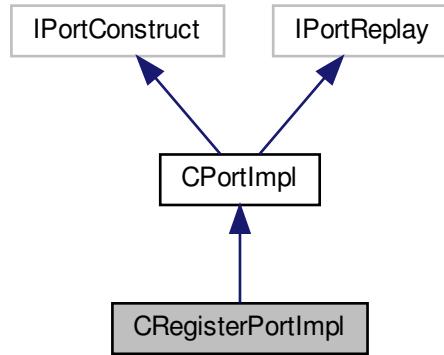
The documentation for this struct was generated from the following file:

- src/GigEVisionPerformance/[CpuUtil.h](#)

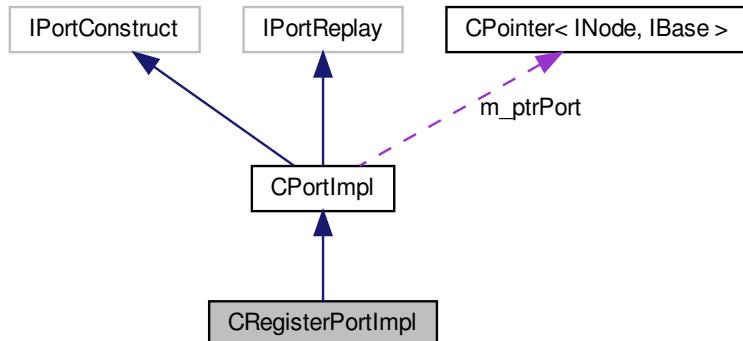
## 14.49 CRegisterPortImpl Class Reference

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 TransportLayerSwapsEndianess=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

### 14.49.1 Detailed Description

Standard implementation for a port using a register based transport layer.

### 14.49.2 Constructor & Destructor Documentation

#### 14.49.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
    int MaxNumQuadlets = 1,
    bool TransportLayerSwapsEndianess = false ) [inline]
```

Constructor.

#### 14.49.2.2 ~CRegisterPortImpl()

```
virtual ~CRegisterPortImpl ( ) [inline], [virtual]
```

Destructor.

### 14.49.3 Member Function Documentation

#### 14.49.3.1 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](#).

#### 14.49.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

#### 14.49.3.3 ReadRegister()

```
virtual void ReadRegister (
    uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads an array of quadlets from the port.

#### 14.49.3.4 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

#### 14.49.3.5 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](#).

#### 14.49.3.6 WriteRegister()

```
virtual void WriteRegister (
    const uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes an array of quadlets to the port.

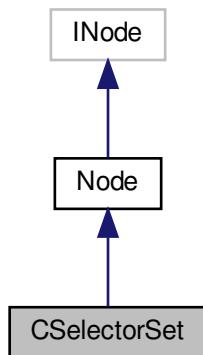
The documentation for this class was generated from the following file:

- include/SpinGenApi/[RegisterPortImpl.h](#)

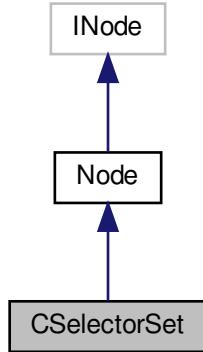
## 14.50 CSelectorSet Class Reference

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 GenlCam::gcstring ToString \(\)](#)
- [virtual void GetSelectorList \(FeatureList\\_t &SelectorList, bool Incremental=false\)](#)

## Additional Inherited Members

### 14.50.1 Detailed Description

The set of selectors selecting a given node.

### 14.50.2 Constructor & Destructor Documentation

#### 14.50.2.1 CSelectorSet()

```
CSelectorSet (
    IBase * pBase )
```

Constructor.

**Parameters**

|                    |                                        |
|--------------------|----------------------------------------|
| <code>pBase</code> | > Feature selected by the selector set |
|--------------------|----------------------------------------|

**14.50.2.2 ~CSelectorSet()**

```
~CSelectorSet ( )
```

Destructor.

**14.50.3 Member Function Documentation****14.50.3.1 GetSelectorList()**

```
virtual void GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [virtual]
```

**14.50.3.2 IsEmpty()**

```
bool IsEmpty ( )
```

returns true if no selectors are present

**14.50.3.3 Restore()**

```
virtual void Restore ( ) [virtual]
```

**14.50.3.4 SetFirst()**

```
virtual bool SetFirst ( ) [virtual]
```

#### 14.50.3.5 SetNext()

```
virtual bool SetNext (
    bool Tick = true ) [virtual]
```

#### 14.50.3.6 ToString()

```
virtual GenICam::gcstring ToString () [virtual]
```

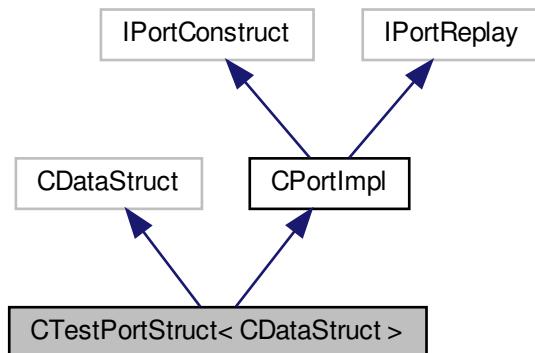
The documentation for this class was generated from the following file:

- include/SpinGenApi/SelectorSet.h

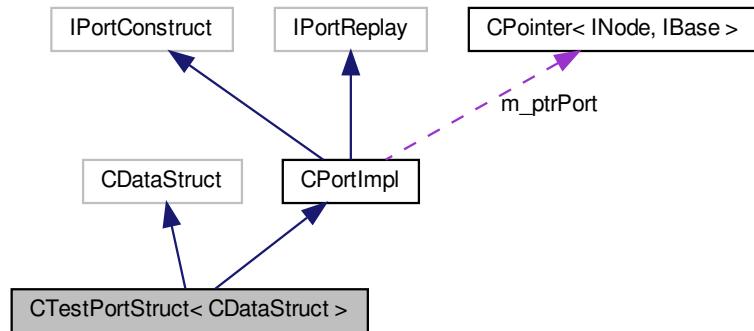
## 14.51 CTestPortStruct< CDataStruct > Class Template Reference

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*

### 14.51.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

## 14.51.2 Constructor & Destructor Documentation

### 14.51.2.1 CTestPortStruct()

```
CTestPortStruct (
    int64_t BaseAddress = 0 ) [inline]
```

## 14.51.3 Member Function Documentation

### 14.51.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Get the access mode of the node.

Implements [CPortImpl](#).

### 14.51.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

### 14.51.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

### 14.51.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

#### 14.51.3.5 MemSet()

```
void MemSet (
    const char FillValue ) [inline]
```

#### 14.51.3.6 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

#### 14.51.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

#### 14.51.3.8 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](#).

### 14.51.4 Member Data Documentation

#### 14.51.4.1 m\_BaseAddress

```
int64_t m_BaseAddress [protected]
```

the base address used for the struct

#### 14.51.4.2 m\_NumReads

```
int64_t m_NumReads [protected]
```

Number of reads since last reset.

#### 14.51.4.3 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](#)

## 14.52 DCAM\_CHECKSUM Struct Reference

### Public Attributes

- uint32\_t [CRCChecksum](#)

#### 14.52.1 Member Data Documentation

##### 14.52.1.1 CRCChecksum

```
uint32_t CRCChecksum
```

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

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

## 14.53 DCAM\_CHUNK\_TRAILER Struct Reference

### Public Attributes

- SPIN\_GUID [ChunkID](#)
- uint32\_t [ChunkLength](#)
- uint32\_t [InverseChunkLength](#)

### 14.53.1 Member Data Documentation

#### 14.53.1.1 ChunkID

```
SPIN_GUID ChunkID
```

#### 14.53.1.2 ChunkLength

```
uint32_t ChunkLength
```

#### 14.53.1.3 InverseChunkLength

```
uint32_t InverseChunkLength
```

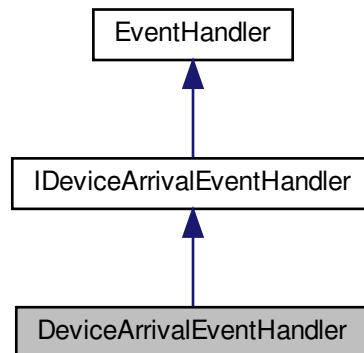
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterDcam.h](#)

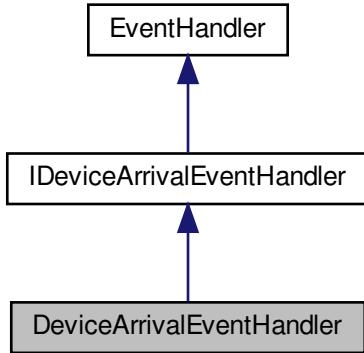
## 14.54 DeviceArrivalEventHandler Class Reference

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 \(uint64\\_t serialNumber\)=0](#)  
*Callback to the device arrival event.*

## Protected Member Functions

- [DeviceArrivalEventHandler & operator= \(const DeviceArrivalEventHandler &\)](#)  
*Assignment operator.*

## Additional Inherited Members

### 14.54.1 Detailed Description

An event handler for capturing the device arrival event.

### 14.54.2 Constructor & Destructor Documentation

#### 14.54.2.1 DeviceArrivalEventHandler()

```
DeviceArrivalEventHandler ( )
```

Default constructor.

#### 14.54.2.2 ~DeviceArrivalEventHandler()

```
virtual ~DeviceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

### 14.54.3 Member Function Documentation

#### 14.54.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device arrival event.

Implements [IDeviceArrivalEventHandler](#).

#### 14.54.3.2 operator=()

```
DeviceArrivalEventHandler& operator= (
    const DeviceArrivalEventHandler & ) [protected]
```

Assignment operator.

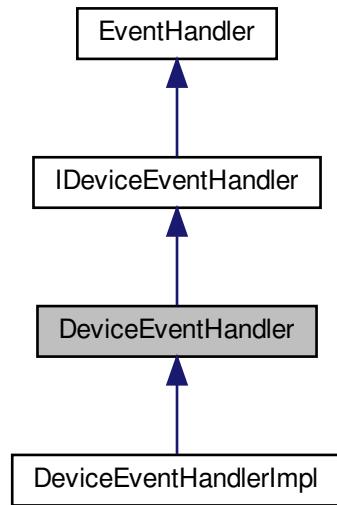
The documentation for this class was generated from the following file:

- [include/DeviceArrivalEventHandler.h](#)

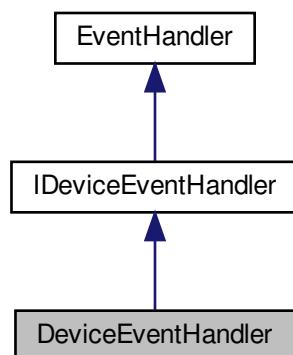
## 14.55 DeviceEventHandler Class Reference

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

### 14.55.1 Detailed Description

A handler to device events.

### 14.55.2 Constructor & Destructor Documentation

#### 14.55.2.1 `DeviceEventHandler()`

`DeviceEventHandler ()`

Default constructor.

#### 14.55.2.2 `~DeviceEventHandler()`

`virtual ~DeviceEventHandler () [virtual]`

Virtual destructor.

### 14.55.3 Member Function Documentation

#### 14.55.3.1 GetDeviceEventId()

```
uint64_t GetDeviceEventId ( ) const [virtual]
```

Get the ID of the device event.

##### Returns

The device event ID

Implements [IDeviceEventHandler](#).

#### 14.55.3.2 GetDeviceEventName()

```
GenICam::gcstring GetDeviceEventName ( ) const [virtual]
```

Get the name of the device event.

##### Returns

The device event name

Implements [IDeviceEventHandler](#).

#### 14.55.3.3 OnDeviceEvent()

```
virtual void OnDeviceEvent ( Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Device event callback.

##### Parameters

|                        |                       |
|------------------------|-----------------------|
| <code>eventName</code> | The name of the event |
|------------------------|-----------------------|

Implements [IDeviceEventHandler](#).

Implemented in [DeviceEventHandlerImpl](#).

#### 14.55.3.4 operator=( )

```
DeviceEventHandler& operator= ( const DeviceEventHandler & ) [protected]
```

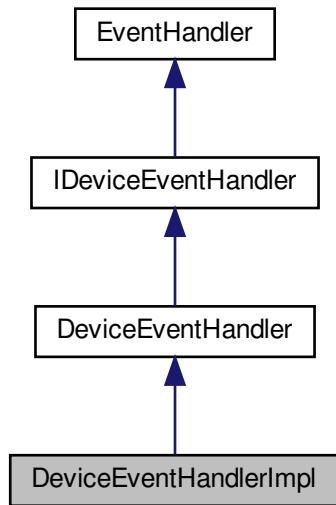
Assignment operator.

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

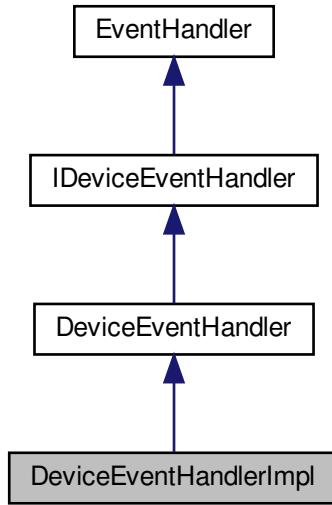
- [include/DeviceEventHandler.h](#)

## 14.56 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

### 14.56.1 Constructor & Destructor Documentation

#### 14.56.1.1 DeviceEventHandlerImpl()

```
DeviceEventHandlerImpl (gcstring eventName) [inline]
```

#### 14.56.1.2 ~DeviceEventHandlerImpl()

```
~DeviceEventHandlerImpl () [inline]
```

## 14.56.2 Member Function Documentation

### 14.56.2.1 OnDeviceEvent()

```
void OnDeviceEvent (
    gcstring eventName ) [inline], [virtual]
```

Device event callback.

#### Parameters

|                  |                       |
|------------------|-----------------------|
| <i>eventName</i> | The name of the event |
|------------------|-----------------------|

Implements [DeviceEventHandler](#).

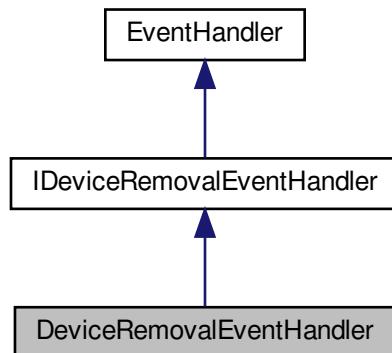
The documentation for this class was generated from the following file:

- [src/DeviceEvents/DeviceEvents.cpp](#)

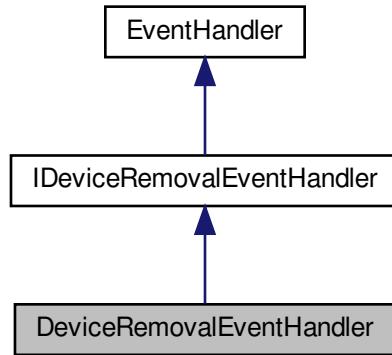
## 14.57 DeviceRemovalEventHandler Class Reference

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 \(uint64\\_t serialNumber\)=0](#)  
*Device removal event callback.*

## Protected Member Functions

- [DeviceRemovalEventHandler & operator= \(const DeviceRemovalEventHandler &\)](#)  
*Assignment operator.*

## Additional Inherited Members

### 14.57.1 Detailed Description

An event handler for capturing the device removal event.

### 14.57.2 Constructor & Destructor Documentation

#### 14.57.2.1 DeviceRemovalEventHandler()

```
DeviceRemovalEventHandler ( )
```

Default Constructor.

#### 14.57.2.2 ~DeviceRemovalEventHandler()

```
virtual ~DeviceRemovalEventHandler ( ) [virtual]
```

Virtual Destructor.

### 14.57.3 Member Function Documentation

#### 14.57.3.1 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Device removal event callback.

##### Parameters

|                     |                                         |
|---------------------|-----------------------------------------|
| <i>serialNumber</i> | The serial number of the device removed |
|---------------------|-----------------------------------------|

Implements [IDeviceRemovalEventHandler](#).

#### 14.57.3.2 operator=( )

```
DeviceRemovalEventHandler& operator= (
    const DeviceRemovalEventHandler & ) [protected]
```

Assignment operator.

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

- [include/DeviceRemovalEventHandler.h](#)

## 14.58 double\_automvector\_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`

### 14.58.1 Detailed Description

Vector of doubles with reference counting.

### 14.58.2 Constructor & Destructor Documentation

#### 14.58.2.1 double\_autovector\_t() [1/3]

```
double_autovector_t( )
```

#### 14.58.2.2 double\_autovector\_t() [2/3]

```
double_autovector_t(
    const double_autovector_t & obj )
```

#### 14.58.2.3 double\_autovector\_t() [3/3]

```
double_autovector_t(
    size_t n ) [explicit]
```

**14.58.2.4 ~double\_autovector\_t()**

```
virtual ~double_autovector_t (
    void ) [virtual]
```

**14.58.3 Member Function Documentation****14.58.3.1 operator delete()**

```
void operator delete (
    void * pWhere )
```

**14.58.3.2 operator new()**

```
void* operator new (
    size_t uiSize )
```

**14.58.3.3 operator=()**

```
double_autovector_t& operator= (
    const double_autovector_t & obj )
```

**14.58.3.4 operator[](1/2)**

```
double& operator[ ] (
    size_t uiIndex )
```

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

```
const double& operator[ ] (
    size_t uiIndex ) const
```

#### 14.58.3.6 size()

```
size_t size ( ) const
```

### 14.58.4 Member Data Documentation

#### 14.58.4.1 \_pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

#### 14.58.4.2 \_pv

```
std::vector<double>* _pv [protected]
```

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

- include/SpinGenApi/Autovector.h

## 14.59 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.*

### 14.59.1 Detailed Description

Holds conversion methods for the access mode enumeration.

### 14.59.2 Member Function Documentation

#### 14.59.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EAccessMode * pValue ) [static]
```

Converts a string to enum value.

#### 14.59.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EAccessMode Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.59.2.3 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

## 14.60 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.*

#### 14.60.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

## 14.60.2 Member Function Documentation

### 14.60.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

Converts a string to enum value.

### 14.60.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ECachingMode Value ) [static]
```

Converts a string to an int32\_t property.

### 14.60.2.3 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

## 14.61 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.*

### 14.61.1 Detailed Description

Holds conversion methods for the notation type of floats.

### 14.61.2 Member Function Documentation

#### 14.61.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to enum value.

#### 14.61.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EDisplayNotation Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.61.2.3 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

## 14.62 EEndianessClass Class Reference

Holds conversion methods for the endianess enumeration.

## Static Public Member Functions

- static bool [FromString](#) (const [GenICam::gcstring](#) &ValueStr, EEndianess \*pValue)  
*Converts a string to enum value.*
- static void [ToString](#) ([GenICam::gcstring](#) &ValueStr, EEndianess \*pValue)  
*Converts a string to an int32\_t property.*
- static [GenICam::gcstring](#) [ToString](#) (EEndianess Value)  
*Converts a string to an int32\_t property.*

### 14.62.1 Detailed Description

Holds conversion methods for the endianess enumeration.

### 14.62.2 Member Function Documentation

#### 14.62.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to enum value.

#### 14.62.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EEndianess Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.62.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EEndianess * 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](#)

## 14.63 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.*

### 14.63.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

### 14.63.2 Member Function Documentation

#### 14.63.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to enum value.

#### 14.63.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EGenApiSchemaVersion Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.63.2.3 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](#)

## 14.64 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.*

### 14.64.1 Detailed Description

Holds conversion methods for the notation type of floats.

### 14.64.2 Member Function Documentation

#### 14.64.2.1 [FromString\(\)](#)

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to enum value.

#### 14.64.2.2 [ToString\(\) \[1/2\]](#)

```
static GenICam::gcstring ToString (
    EInputDirection Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.64.2.3 [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](#)

## 14.65 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.*

### 14.65.1 Detailed Description

Holds conversion methods for the namespace enumeration.

### 14.65.2 Member Function Documentation

#### 14.65.2.1 [FromString\(\)](#)

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to enum value.

#### 14.65.2.2 [ToString\(\) \[1/2\]](#)

```
static GenICam::gcstring ToString (
    ENameSpace Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.65.2.3 [ToString\(\) \[2/2\]](#)

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ENameSpace * pValue ) [static]
```

Converts a string to an int32\_t property.

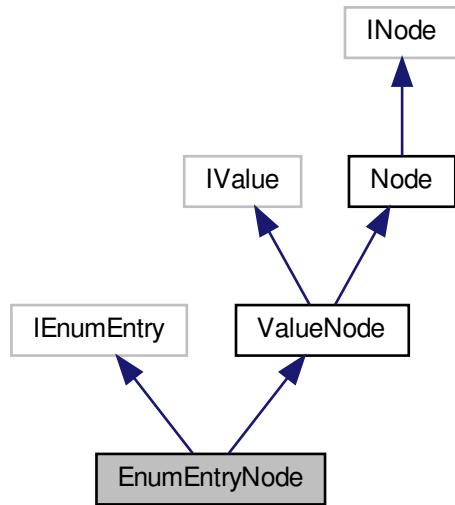
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

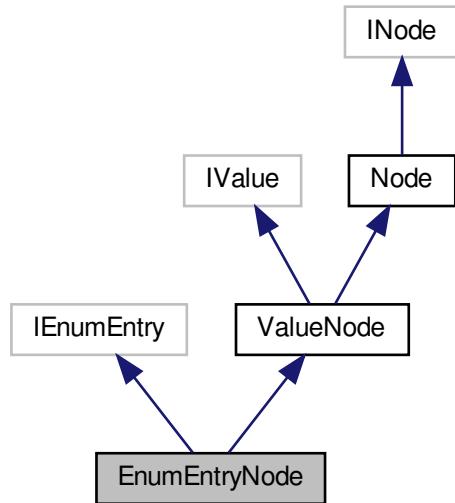
## 14.66 EnumEntryNode Class Reference

[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 GenICam::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

### 14.66.1 Detailed Description

[Interface](#) for string properties.

### 14.66.2 Constructor & Destructor Documentation

#### 14.66.2.1 `EnumEntryNode()` [1/2]

```
EnumEntryNode ( )
```

#### 14.66.2.2 `EnumEntryNode()` [2/2]

```
EnumEntryNode ( std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

#### 14.66.2.3 `~EnumEntryNode()`

```
virtual ~EnumEntryNode ( ) [virtual]
```

### 14.66.3 Member Function Documentation

#### 14.66.3.1 GetNumericValue()

```
virtual double GetNumericValue ( ) [virtual]
```

Get double number associated with the entry.

#### 14.66.3.2 GetSymbolic()

```
virtual GenICam::gcstring GetSymbolic ( ) const [virtual]
```

Get symbolic enum value.

#### 14.66.3.3 GetValue()

```
virtual int64_t GetValue ( ) [virtual]
```

Get numeric enum value.

#### 14.66.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( ) [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

#### 14.66.3.5 SetReference()

```
virtual void SetReference (  
    INode * pBase ) [virtual]
```

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

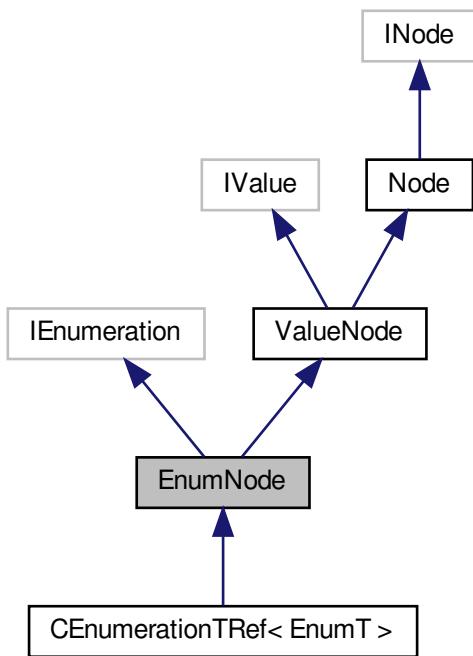
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumEntryNode.h](#)

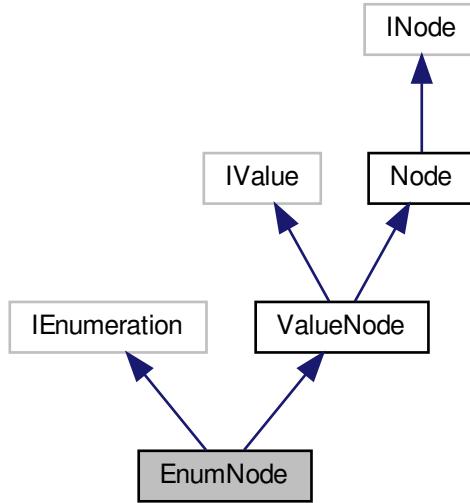
## 14.67 EnumNode Class Reference

[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](#)

### 14.67.1 Detailed Description

[Interface](#) for string properties.

### 14.67.2 Constructor & Destructor Documentation

#### 14.67.2.1 [EnumNode\(\)](#) [1/2]

```
EnumNode ( )
```

#### 14.67.2.2 [EnumNode\(\)](#) [2/2]

```
EnumNode (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

#### 14.67.2.3 [~EnumNode\(\)](#)

```
virtual ~EnumNode ( ) [virtual]
```

### 14.67.3 Member Function Documentation

#### 14.67.3.1 [GetCurrentEntry\(\)](#)

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

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

### 14.67.3.2 GetEntries()

```
virtual void GetEntries (
    NodeList_t & Entries ) [virtual]
```

Get list of entry nodes.

### 14.67.3.3 GetEntry()

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

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

### 14.67.3.4 GetEntryByName()

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

Get an entry node by name.

### 14.67.3.5 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

#### 14.67.3.6 GetSymbolics()

```
virtual void GetSymbolics (
    StringList_t & Symbolics ) [virtual]
```

Get list of symbolic Values.

#### 14.67.3.7 operator\*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get string node value.

#### 14.67.3.8 operator=()

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set string node value.

Reimplemented in [CEnumerationTRef<EnumT>](#).

#### 14.67.3.9 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) |

#### 14.67.3.10 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

#### 14.67.4 Member Data Documentation

##### 14.67.4.1 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](#)

## 14.68 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.*

### 14.68.1 Detailed Description

Holds conversion methods for the representation enumeration.

### 14.68.2 Member Function Documentation

#### 14.68.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to enum value.

#### 14.68.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ERepresentation Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.68.2.3 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

## 14.69 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.*

#### 14.69.1 Detailed Description

Holds conversion methods for the sign enumeration.

## 14.69.2 Member Function Documentation

### 14.69.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to enum value.

### 14.69.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ESign Value ) [static]
```

Converts a string to an int32\_t property.

### 14.69.2.3 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

## 14.70 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.*

### 14.70.1 Detailed Description

Holds conversion methods for the converter formulas.

### 14.70.2 Member Function Documentation

#### 14.70.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to enum value.

#### 14.70.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    ESlope Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.70.2.3 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

## 14.71 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.*

### 14.71.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

### 14.71.2 Member Function Documentation

#### 14.71.2.1 [FromString\(\)](#)

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to enum value.

#### 14.71.2.2 [ToString\(\)](#) [1/2]

```
static GenICam::gcstring ToString (
    EStandardNameSpace Value ) [static]
```

Converts a string to an int32\_t property.

#### 14.71.2.3 [ToString\(\)](#) [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to an int32\_t property.

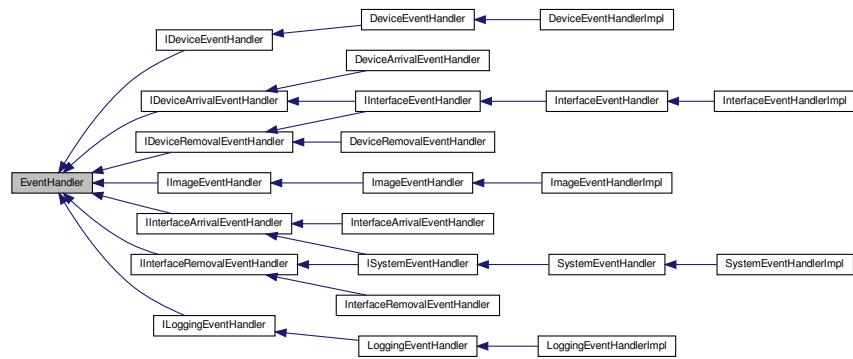
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EnumClasses.h](#)

## 14.72 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 `EventProcessor`
- class `IDataStream`
- class `Stream`

### 14.72.1 Detailed Description

The base class for all event handler types.

### 14.72.2 Constructor & Destructor Documentation

#### 14.72.2.1 ~EventHandler()

```
virtual ~EventHandler ( ) [virtual]
```

Virtual Destructor.

#### 14.72.2.2 EventHandler()

```
EventHandler ( ) [protected]
```

### 14.72.3 Member Function Documentation

#### 14.72.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData ( )
```

Gets the event payload data.

##### Returns

The event payload data

#### 14.72.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

##### Returns

The event payload data size

#### 14.72.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

##### Returns

The event type

#### 14.72.3.4 operator=( )

```
EventHandler& operator= (
    const EventHandler & ) [protected]
```

#### 14.72.3.5 SetEventPayload()

```
void SetEventPayload (
    uint8_t * offset,
    size_t length ) [protected]
```

#### 14.72.3.6 SetEventType()

```
void SetEventType (
    EventType eventType )
```

Sets the event type.

##### Parameters

|                  |                |
|------------------|----------------|
| <i>eventType</i> | The event type |
|------------------|----------------|

### 14.72.4 Friends And Related Function Documentation

#### 14.72.4.1 EventProcessor

```
friend class EventProcessor [friend]
```

#### 14.72.4.2 IDataStream

```
friend class IDataStream [friend]
```

#### 14.72.4.3 Stream

```
friend class Stream [friend]
```

### 14.72.5 Member Data Documentation

#### 14.72.5.1 m\_pEventData

```
EventData* m_pEventData [protected]
```

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

- [include/EventHandler.h](#)

## 14.73 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.*

#### 14.73.1 Detailed Description

Holds conversion methods for the visibility enumeration.

#### 14.73.2 Member Function Documentation

### 14.73.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to enum value.

### 14.73.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EVisibility Value ) [static]
```

Converts a string to an int32\_t property.

### 14.73.2.3 ToString() [2/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to an int32\_t property.

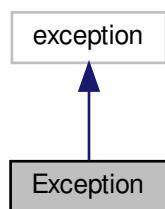
The documentation for this class was generated from the following file:

- include/SpinGenApi/EnumClasses.h

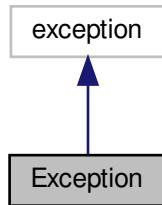
## 14.74 Exception Class Reference

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`

### 14.74.1 Detailed Description

The `Exception` object represents an error that is returned from the library.

Overloaded operators allow comparisons against other `Exception` objects.

## 14.74.2 Constructor & Destructor Documentation

### 14.74.2.1 Exception() [1/4]

```
Exception ( )
```

Default constructor.

### 14.74.2.2 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                                |

### 14.74.2.3 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                   |

**Parameters**

|                  |                                           |
|------------------|-------------------------------------------|
| <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                                |

**14.74.2.4 Exception() [4/4]**

```
Exception (  
    const Exception & except )
```

Copy constructor.

**14.74.2.5 ~Exception()**

```
virtual ~Exception ( ) throw ( ) [virtual]
```

Default destructor.

**14.74.3 Member Function Documentation****14.74.3.1 GetBuildDate()**

```
const char* GetBuildDate ( ) const
```

**14.74.3.2 GetBuildTime()**

```
const char* GetBuildTime ( ) const
```

**14.74.3.3 GetError()**

```
Error GetError ( ) const
```

#### 14.74.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

#### 14.74.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

#### 14.74.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

#### 14.74.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

#### 14.74.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

#### 14.74.3.9 operator"!=()

```
bool operator!= (
    const Error err ) const
```

Inequality operator.

#### 14.74.3.10 operator=( )

```
Exception& operator= (
    const Exception & except )
```

Assignment operator.

#### 14.74.3.11 operator==( )

```
bool operator== (
    const Error err ) const
```

Equality operator.

#### 14.74.3.12 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](#)

## 14.75 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.*

### 14.75.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

## 14.75.2 Member Function Documentation

### 14.75.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EYesNo * pValue ) [static]
```

Converts a string to enum value.

### 14.75.2.2 ToString() [1/2]

```
static GenICam::gcstring ToString (
    EYesNo Value ) [static]
```

Converts a string to an int32\_t property.

### 14.75.2.3 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

## 14.76 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.*

### 14.76.1 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

### 14.76.2 Constructor & Destructor Documentation

#### 14.76.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ()
```

Constructor.

#### 14.76.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter () [virtual]
```

### 14.76.3 Member Function Documentation

#### 14.76.3.1 attach()

```
bool attach (
    ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to NodeMap

**Parameters**

|                   |                                                                                    |
|-------------------|------------------------------------------------------------------------------------|
| <i>pInterface</i> | NodeMap of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
|-------------------|------------------------------------------------------------------------------------|

**Returns**

true if attach was successful, false if not

**14.76.3.2 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

**14.76.3.3 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

**14.76.3.4 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 FileMode                |

#### Returns

max length of FileAccessBuffer in the given mode on the given file

### 14.76.3.5 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 FileMode                |

#### Returns

true on success, false on error

### 14.76.3.6 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

**14.76.3.7 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

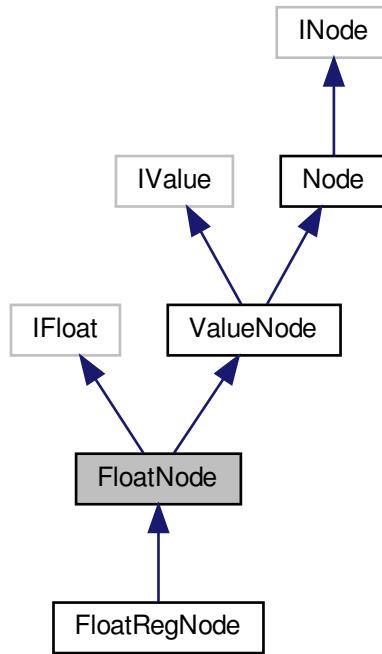
The documentation for this class was generated from the following file:

- include/SpinGenApi/Filestream.h

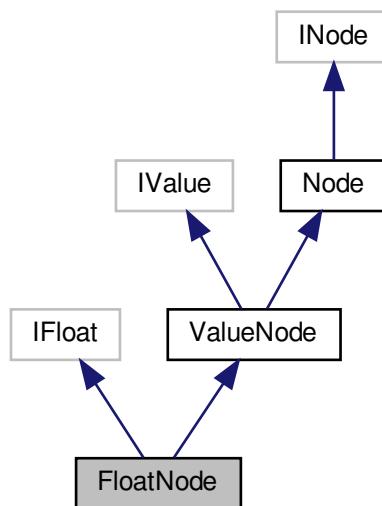
**14.77 FloatNode Class Reference**

[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_vector_t GetListOfValidValues (bool bounded=true)`  
`Get list of valid value.`
- `virtual ERepresentation GetRepresentation ()`  
`Get recommended representation.`
- `virtual GenICam::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.`
- `IInteger * 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

### 14.77.1 Detailed Description

Interface for string properties.

## 14.77.2 Constructor & Destructor Documentation

### 14.77.2.1 `FloatNode()` [1/2]

```
FloatNode ( )
```

### 14.77.2.2 `FloatNode()` [2/2]

```
FloatNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

### 14.77.2.3 `~FloatNode()`

```
virtual ~FloatNode ( ) [virtual]
```

## 14.77.3 Member Function Documentation

### 14.77.3.1 `GetDisplayNotation()`

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

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

### 14.77.3.2 `GetDisplayPrecision()`

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

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

### 14.77.3.3 `GetEnumAlias()`

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

**14.77.3.4 GetInc()**

```
virtual double GetInc ( ) [virtual]
```

Get the constant increment if there is any.

**14.77.3.5 GetIncMode()**

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

Get increment mode.

**14.77.3.6 GetIntAlias()**

```
IIInteger* GetIntAlias ( )
```

gets the interface of an alias node.

**14.77.3.7 GetListOfValidValues()**

```
virtual double_vector_t GetListOfValidValues ( bool bounded = true ) [virtual]
```

Get list of valid value.

**14.77.3.8 GetMax()**

```
virtual double GetMax ( ) [virtual]
```

Get maximum value allowed.

**14.77.3.9 GetMin()**

```
virtual double GetMin ( ) [virtual]
```

Get minimum value allowed.

#### 14.77.3.10 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

#### 14.77.3.11 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) const [virtual]
```

Get the physical unit name.

#### 14.77.3.12 GetValue()

```
virtual double GetValue (  
    bool Verify = false,  
    bool IgnoreCache = false ) [virtual]
```

Get node value.

##### Parameters

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

##### Returns

The value read

#### 14.77.3.13 HasInc()

```
virtual bool HasInc ( ) [virtual]
```

True if the float has a constant increment.

#### 14.77.3.14 ImposeMax()

```
virtual void ImposeMax (   
    double Value ) [virtual]
```

Restrict maximum value.

**14.77.3.15 ImposeMin()**

```
virtual void ImposeMin (
    double Value ) [virtual]
```

Restrict minimum value.

**14.77.3.16 operator()()**

```
virtual double operator() () [virtual]
```

Get node value.

**14.77.3.17 operator\*()**

```
virtual double operator* () [virtual]
```

Get node value.

**14.77.3.18 operator=( )**

```
virtual IFloat& operator= (
    double Value ) [virtual]
```

Set node value.

**14.77.3.19 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

**14.77.3.20 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) |

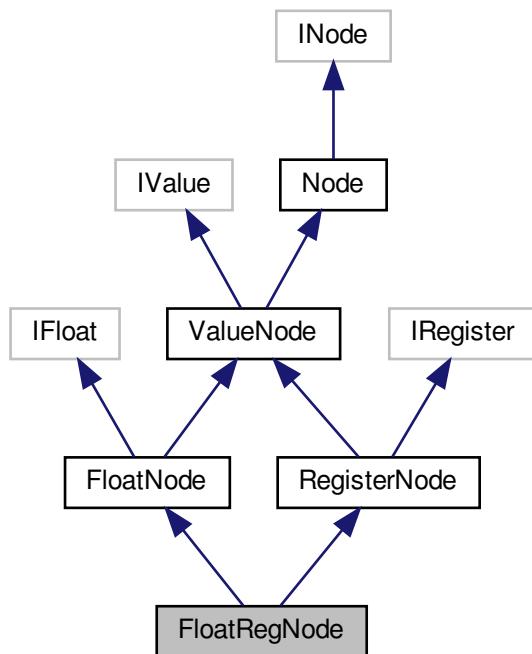
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatNode.h](#)

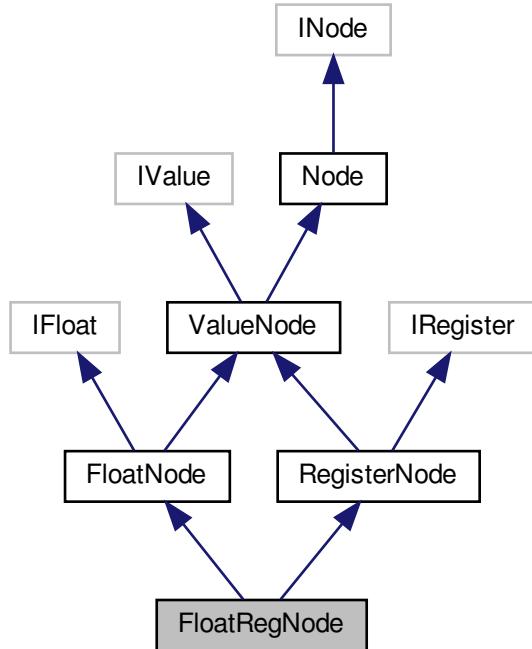
## 14.78 FloatRegNode Class Reference

[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

### 14.78.1 Detailed Description

[Interface](#) for string properties.

### 14.78.2 Constructor & Destructor Documentation

**14.78.2.1 [FloatRegNode\(\)](#) [1/2]**

```
FloatRegNode ( )
```

**14.78.2.2 [FloatRegNode\(\)](#) [2/2]**

```
FloatRegNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

**14.78.2.3 [~FloatRegNode\(\)](#)**

```
virtual ~FloatRegNode ( ) [virtual]
```

**14.78.3 Member Function Documentation****14.78.3.1 [SetReference\(\)](#)**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [FloatNode](#).

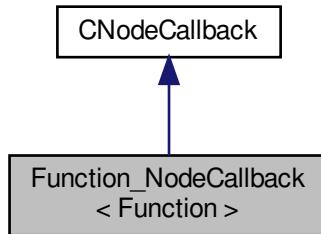
The documentation for this class was generated from the following file:

- include/SpinGenApi/[FloatRegNode.h](#)

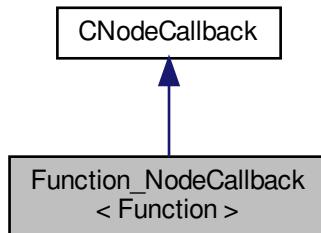
## 14.79 Function\_NodeCallback< Function > Class Template Reference

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

#### 14.79.1 Detailed Description

```
template<class Function>
class Spinnaker::GenApi::Function_NodeCallback< Function >
```

Container for a function pointer.

## 14.79.2 Constructor & Destructor Documentation

### 14.79.2.1 Function\_NodeCallback()

```
Function_NodeCallback (
    INode * pNode,
    const Function & function,
    ECallbackType CallbackType ) [inline]
```

Constructor.

## 14.79.3 Member Function Documentation

### 14.79.3.1 Destroy()

```
virtual void Destroy () [inline], [virtual]
```

destroys teh object

Implements [CNodeCallback](#).

### 14.79.3.2 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](#)

## 14.80 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)`

## 14.80.1 Constructor & Destructor Documentation

### 14.80.1.1 `gcstring()` [1/5]

```
gcstring ( )
```

### 14.80.1.2 `gcstring()` [2/5]

```
gcstring (
    const char * pc )
```

### 14.80.1.3 `gcstring()` [3/5]

```
gcstring (
    const char * pc,
    size_t n )
```

### 14.80.1.4 `gcstring()` [4/5]

```
gcstring (
    size_t count,
    char ch )
```

**14.80.1.5 `gcstring()` [5/5]**

```
gcstring (
    const gcstring & str )
```

**14.80.1.6 `~gcstring()`**

```
virtual ~gcstring (
    void ) [virtual]
```

**14.80.2 Member Function Documentation****14.80.2.1 `_npos()`**

```
static size_t _npos (
    void ) [static]
```

**14.80.2.2 `append()` [1/2]**

```
virtual gcstring& append (
    const gcstring & str ) [virtual]
```

**14.80.2.3 `append()` [2/2]**

```
virtual gcstring& append (
    size_t count,
    char ch ) [virtual]
```

**14.80.2.4 `assign()` [1/4]**

```
virtual gcstring& assign (
    const char * pc ) [virtual]
```

**14.80.2.5 assign() [2/4]**

```
virtual gcstring& assign (
    const char * pc,
    size_t n )  [virtual]
```

**14.80.2.6 assign() [3/4]**

```
virtual gcstring& assign (
    const gcstring & str )  [virtual]
```

**14.80.2.7 assign() [4/4]**

```
virtual gcstring& assign (
    size_t count,
    char ch )  [virtual]
```

**14.80.2.8 c\_str()**

```
virtual const char* c_str (
    void ) const  [virtual]
```

**14.80.2.9 compare()**

```
virtual int compare (
    const gcstring & str ) const  [virtual]
```

**14.80.2.10 empty()**

```
virtual bool empty (
    void ) const  [virtual]
```

**14.80.2.11 find() [1/5]**

```
virtual size_t find (
    char ch,
    size_t offset = 0 ) const [virtual]
```

**14.80.2.12 find() [2/5]**

```
virtual size_t find (
    const char * pc,
    size_t offset,
    size_t count ) const [virtual]
```

**14.80.2.13 find() [3/5]**

```
virtual size_t find (
    const char * pc,
    size_t offset = 0 ) const [virtual]
```

**14.80.2.14 find() [4/5]**

```
virtual size_t find (
    const gcstring & str,
    size_t offset,
    size_t count ) const [virtual]
```

**14.80.2.15 find() [5/5]**

```
virtual size_t find (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

**14.80.2.16 find\_first\_not\_of()**

```
virtual size_t find_first_not_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

**14.80.2.17 find\_first\_of()**

```
virtual size_t find_first_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

**14.80.2.18 length()**

```
virtual size_t length (
    void ) const [virtual]
```

**14.80.2.19 max\_size()**

```
virtual size_t max_size ( ) const [virtual]
```

**14.80.2.20 operator const char \*()**

```
operator const char * (
    void ) const
```

**14.80.2.21 operator delete() [1/2]**

```
void operator delete (
    void * pWhere )
```

**14.80.2.22 operator delete() [2/2]**

```
void operator delete (
    void * pWhere,
    void * pNewWhere )
```

**14.80.2.23 operator new() [1/2]**

```
void* operator new (
    size_t uiSize )
```

**14.80.2.24 operator new() [2/2]**

```
void* operator new (
    size_t uiSize,
    void * pWhere )
```

**14.80.2.25 operator"!=() [1/2]**

```
bool operator!= (
    const char * pc ) const
```

**14.80.2.26 operator"!=() [2/2]**

```
bool operator!= (
    const gcstring & str ) const
```

**14.80.2.27 operator+=(() [1/5]**

```
gcstring& operator+= (
    char ch )
```

**14.80.2.28 operator+=(() [2/5]**

```
gcstring operator+= (
    char ch ) const
```

**14.80.2.29 operator+=(() [3/5]**

```
gcstring& operator+= (
    const char * pc )
```

**14.80.2.30 operator+=(() [4/5]**

```
gcstring& operator+= (
    const gcstring & str )
```

**14.80.2.31 operator+=() [5/5]**

```
gcstring operator+= (
    const gcstring & str ) const
```

**14.80.2.32 operator<()**

```
bool operator< (
    const gcstring & str ) const
```

**14.80.2.33 operator=(())**

```
gcstring& operator= (
    const gcstring & str )
```

**14.80.2.34 operator==(()) [1/2]**

```
bool operator== (
    const char * pc ) const
```

**14.80.2.35 operator==(()) [2/2]**

```
bool operator== (
    const gcstring & str ) const
```

**14.80.2.36 operator>()**

```
bool operator> (
    const gcstring & str ) const
```

**14.80.2.37 resize()**

```
virtual void resize (
    size_t n ) [virtual]
```

**14.80.2.38 size()**

```
virtual size_t size (
    void ) const [virtual]
```

**14.80.2.39 substr()**

```
virtual gcstring substr (
    size_t offset = 0,
    size_t count = GCSTRING_NPOS ) const [virtual]
```

**14.80.2.40 swap()**

```
virtual void swap (
    gcstring & Right ) [virtual]
```

**14.80.3 Friends And Related Function Documentation****14.80.3.1 operator+ [1/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const char * left,
    const gcstring & right ) [friend]
```

**14.80.3.2 operator+ [2/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const char * right ) [friend]
```

**14.80.3.3 operator+ [3/3]**

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const gcstring & right ) [friend]
```

## 14.80.4 Member Data Documentation

### 14.80.4.1 npos

```
const size_t npos [static]
```

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

- include/SpinGenApi/GCString.h

## 14.81 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](#)

### 14.81.1 Constructor & Destructor Documentation

#### 14.81.1.1 GrabInfo()

```
GrabInfo (
    const string & deviceSerial ) [inline]
```

### 14.81.2 Member Data Documentation

#### 14.81.2.1 imageEventHandler

```
std::shared_ptr<ImageEventHandlerImpl> imageEventHandler
```

#### 14.81.2.2 numImagesGrabbed

```
unsigned int numImagesGrabbed
```

#### 14.81.2.3 numIncompleteImages

```
unsigned int numIncompleteImages
```

#### 14.81.2.4 numRemovals

```
unsigned int numRemovals
```

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

- src/AcquisitionMultipleCameraRecovery/[AcquisitionMultipleCameraRecovery.cpp](#)

## 14.82 GVCP\_CHUNK\_TRAILER Struct Reference

header of a GVCP request packet

### Public Attributes

- uint32\_t [ChunkID](#)
- uint32\_t [ChunkLength](#)

#### 14.82.1 Detailed Description

header of a GVCP request packet

#### 14.82.2 Member Data Documentation

##### 14.82.2.1 ChunkID

```
uint32_t ChunkID
```

### 14.82.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

- include/SpinGenApi/ChunkAdapterGEV.h

## 14.83 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](#)

### 14.83.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

### 14.83.2 Member Data Documentation

#### 14.83.2.1 BlockId

```
uint16_t BlockId
```

#### 14.83.2.2 EventId

```
uint16_t EventId
```

#### 14.83.2.3 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

#### 14.83.2.4 StreamChannelId

```
uint16_t StreamChannelId
```

#### 14.83.2.5 TimestampHigh

```
uint32_t TimestampHigh
```

#### 14.83.2.6 TimestampLow

```
uint32_t TimestampLow
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

### 14.84 GVCP\_EVENT\_ITEM\_BASIC Struct Reference

layout of a GVCP event item (common to all types)

#### Public Attributes

- `uint16_t ReservedOrEventSize`
- `uint16_t EventId`

#### 14.84.1 Detailed Description

layout of a GVCP event item (common to all types)

#### 14.84.2 Member Data Documentation

#### 14.84.2.1 EventId

```
uint16_t EventId
```

#### 14.84.2.2 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.85 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](#)

#### 14.85.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

#### 14.85.2 Member Data Documentation

##### 14.85.2.1 BlockId

```
uint16_t BlockId
```

**14.85.2.2 BlockId64High**

```
uint32_t BlockId64High
```

**14.85.2.3 BlockId64Low**

```
uint32_t BlockId64Low
```

**14.85.2.4 EventId**

```
uint16_t EventId
```

**14.85.2.5 ReservedOrEventSize**

```
uint16_t ReservedOrEventSize
```

**14.85.2.6 StreamChannelId**

```
uint16_t StreamChannelId
```

**14.85.2.7 TimestampHigh**

```
uint32_t TimestampHigh
```

**14.85.2.8 TimestampLow**

```
uint32_t TimestampLow
```

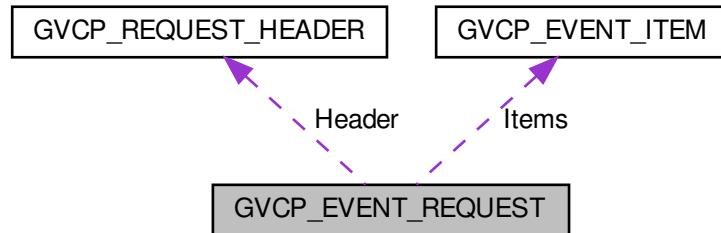
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.86 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]

#### 14.86.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

#### 14.86.2 Member Data Documentation

##### 14.86.2.1 Header

`GVCP_REQUEST_HEADER` Header

##### 14.86.2.2 Items

`GVCP_EVENT_ITEM` Items[1]

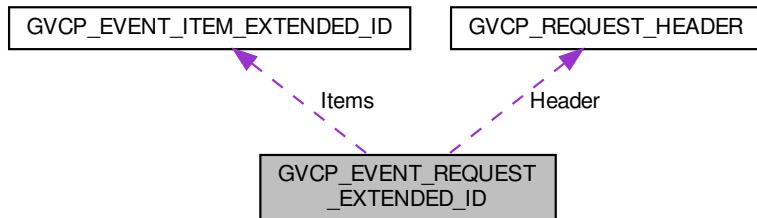
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.87 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]

#### 14.87.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

#### 14.87.2 Member Data Documentation

##### 14.87.2.1 Header

`GVCP_REQUEST_HEADER` Header

##### 14.87.2.2 Items

`GVCP_EVENT_ITEM_EXTENDED_ID` Items [1]

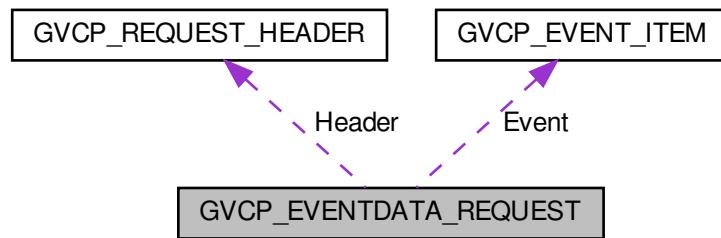
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.88 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]`

#### 14.88.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

#### 14.88.2 Member Data Documentation

##### 14.88.2.1 Data

```
uint32_t Data[1]
```

##### 14.88.2.2 Event

```
GVCP_EVENT_ITEM Event
```

### 14.88.2.3 Header

`GVCP_REQUEST_HEADER` Header

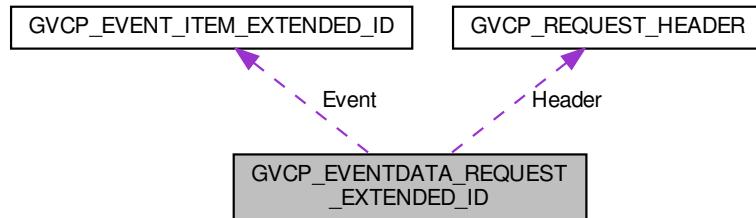
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.89 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]`

### 14.89.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

### 14.89.2 Member Data Documentation

#### 14.89.2.1 Data

`uint32_t Data [1]`

### 14.89.2.2 Event

`GVCP_EVENT_ITEM_EXTENDED_ID` Event

### 14.89.2.3 Header

`GVCP_REQUEST_HEADER` Header

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.90 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`

### 14.90.1 Detailed Description

header of a GVCP request packet

### 14.90.2 Member Data Documentation

#### 14.90.2.1 Command

`uint16_t` Command

#### 14.90.2.2 Flags

`uint8_t` Flags

#### 14.90.2.3 Length

```
uint16_t Length
```

#### 14.90.2.4 Magic

```
uint8_t Magic
```

#### 14.90.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/[EventAdapterGEV.h](#)

## 14.91 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.*

#### 14.91.1 Detailed Description

Options for saving H264 files.

## 14.91.2 Constructor & Destructor Documentation

### 14.91.2.1 H264Option()

```
H264Option ( ) [inline]
```

## 14.91.3 Member Data Documentation

### 14.91.3.1 bitrate

```
unsigned int bitrate
```

Bit-rate to encode at.

### 14.91.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

### 14.91.3.3 height

```
unsigned int height
```

Height of source image.

### 14.91.3.4 reserved

```
unsigned int reserved[256]
```

Reserved for future use.

#### 14.91.3.5 width

```
unsigned int width
```

Width of source image.

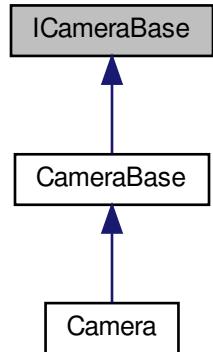
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

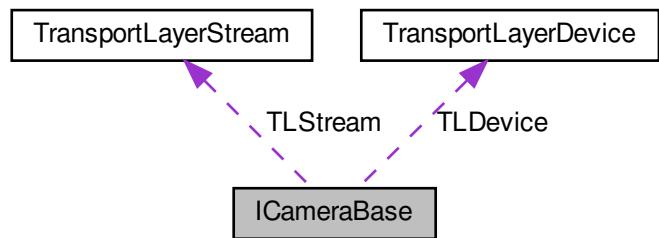
## 14.92 ICameraBase Class Reference

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 [DeInit \(\)=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 \(\) 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 \(uint64\\_t grabTimeout=EVENT\\_TIMEOUT\\_INFINITE, uint64\\_t streamID=0\)=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 \(EventHandler &evtHandlerToRegister, const GenICam::gcstring &eventName\)=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 [CameralInternal](#)
- class [InterfacelImpl](#)

### 14.92.1 Detailed Description

The interface file for base class for the camera object.

### 14.92.2 Constructor & Destructor Documentation

#### 14.92.2.1 ~ICameraBase()

```
virtual ~ICameraBase (
    void ) [inline], [virtual]
```

#### 14.92.2.2 ICameraBase() [1/2]

```
ICameraBase () [inline], [protected]
```

#### 14.92.2.3 ICameraBase() [2/2]

```
ICameraBase (
    const ICameraBase & ) [inline], [protected]
```

### 14.92.3 Member Function Documentation

#### 14.92.3.1 BeginAcquisition()

```
virtual void BeginAcquisition () [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.2 DeInit()

```
virtual void DeInit ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.4 EndAcquisition()

```
virtual void EndAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.5 ForceIP()

```
virtual void ForceIP ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.8 GetGuiXml()

```
virtual GenICam::gcstring GetGuiXml ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT\_TIMEOUT\_INFINITE,
    uint64_t streamID = 0 ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.10 GetNodeMap()

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) [pure virtual]
```

Implemented in [CameraBase](#).

#### 14.92.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.14 GetTLStreamNodeMap()**

```
virtual GenApi::INodeMap& GetTLStreamNodeMap() const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.15 GetUniqueId()**

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

Implemented in [CameraBase](#).

**14.92.3.16 GetUserBufferCount()**

```
virtual uint64_t GetUserBufferCount() const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.17 GetUserBufferSize()**

```
virtual uint64_t GetUserBufferSize() const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.18 GetUserBufferTotalSize()**

```
virtual uint64_t GetUserBufferTotalSize() const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.19 Init()**

```
virtual void Init() [pure virtual]
```

Implemented in [CameraBase](#), and [Camera](#).

**14.92.3.20 IsInitialized()**

```
virtual bool IsInitialized ( ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.21 IsStreaming()**

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.22 IsValid()**

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.23 operator=( )**

```
ICameraBase& operator= (
    const ICameraBase & ) [protected]
```

**14.92.3.24 ReadPort()**

```
virtual void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.25 RegisterEventHandler() [1/2]**

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.26 RegisterEventHandler() [2/2]**

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister,
    const GenICam::gcstring & eventName ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.27 SetBufferOwnership()**

```
virtual void SetBufferOwnership (
    const BufferOwnership mode ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.28 SetUserBuffers() [1/2]**

```
virtual void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.29 SetUserBuffers() [2/2]**

```
virtual void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [pure virtual]
```

Implemented in [CameraBase](#).

**14.92.3.30 UnregisterEventHandler()**

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [CameraBase](#).

### 14.92.3.31 WritePort()

```
virtual void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

## 14.92.4 Friends And Related Function Documentation

### 14.92.4.1 CameraInternal

```
friend class CameraInternal [friend]
```

### 14.92.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

## 14.92.5 Member Data Documentation

### 14.92.5.1 m\_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData [protected]
```

### 14.92.5.2 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.

#### 14.92.5.3 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.

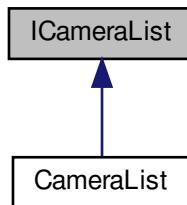
The documentation for this class was generated from the following file:

- `include/Interface/ICameraBase.h`

## 14.93 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 `RemoveBySerial (std::string serialNumber)=0`
- virtual void `RemoveByIndex (unsigned int index)=0`
- virtual void `RemoveByDeviceID (std::string deviceID)=0`
- virtual void `Append (CameraList &otherList)=0`

### Protected Member Functions

- `ICameraList ()`
- `ICameraList (const ICameraList &)`
- `ICameraList & operator= (const ICameraList &)`

## Protected Attributes

- CameraListData \* `m_pCameraListData`

## Friends

- class `InterfacImpl`
- class `CameraListImpl`

### 14.93.1 Detailed Description

Used to hold a list of camera objects.

### 14.93.2 Constructor & Destructor Documentation

#### 14.93.2.1 ~ICameraList()

```
virtual ~ICameraList ( ) [inline], [virtual]
```

#### 14.93.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

#### 14.93.2.3 ICameraList() [2/2]

```
ICameraList (
    const ICameraList & ) [inline], [protected]
```

### 14.93.3 Member Function Documentation

#### 14.93.3.1 Append()

```
virtual void Append (
    CameraList & otherList ) [pure virtual]
```

Implemented in `CameraList`.

#### 14.93.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.3 GetByDeviceID()

```
virtual CameraPtr GetByDeviceID (  
    std::string deviceID ) const [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.4 GetByIndex()

```
virtual CameraPtr GetByIndex (  
    unsigned int index ) const [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.5 GetBySerial()

```
virtual CameraPtr GetBySerial (  
    std::string serialNumber ) const [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.6 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.7 operator=()

```
ICameraList& operator= (const ICameraList & ) [protected]
```

#### 14.93.3.8 operator[]( )

```
virtual CameraPtr operator[ ] (  
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.9 RemoveByDeviceID()

```
virtual void RemoveByDeviceID (   
    std::string deviceID ) [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.10 RemoveByIndex()

```
virtual void RemoveByIndex (   
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

#### 14.93.3.11 RemoveBySerial()

```
virtual void RemoveBySerial (   
    std::string serialNumber ) [pure virtual]
```

Implemented in [CameraList](#).

### 14.93.4 Friends And Related Function Documentation

#### 14.93.4.1 CameraListImpl

```
friend class CameraListImpl [friend]
```

#### 14.93.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

### 14.93.5 Member Data Documentation

#### 14.93.5.1 m\_pCameraListData

```
CameraListData* m_pCameraListData [protected]
```

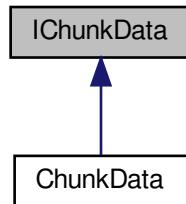
The documentation for this class was generated from the following file:

- include/Interface/ICameraList.h

## 14.94 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 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` ()

### 14.94.1 Detailed Description

The [Interface](#) file for `ChunkData`.

### 14.94.2 Constructor & Destructor Documentation

#### 14.94.2.1 ~IChunkData()

```
virtual ~IChunkData ( ) [inline], [virtual]
```

#### 14.94.2.2 IChunkData()

```
IChunkData ( ) [inline], [protected]
```

### 14.94.3 Member Function Documentation

#### 14.94.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

#### 14.94.3.2 GetCounterValue()

```
virtual int64_t GetCounterValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

#### 14.94.3.3 GetCRC()

```
virtual int64_t GetCRC ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

#### 14.94.3.4 GetEncoderValue()

```
virtual int64_t GetEncoderValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

#### 14.94.3.5 GetExposureEndLineStatusAll()

```
virtual int64_t GetExposureEndLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.6 GetExposureTime()**

```
virtual float64_t GetExposureTime () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.7 GetFrameID()**

```
virtual int64_t GetFrameID () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.8 GetGain()**

```
virtual float64_t GetGain () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.9 GetHeight()**

```
virtual int64_t GetHeight () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.10 GetImage()**

```
virtual int64_t GetImage () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.11 GetInferenceBoundingBoxResult()**

```
virtual InferenceBoundingBoxResult GetInferenceBoundingBoxResult () const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.12 GetInferenceConfidence()**

```
virtual float64_t GetInferenceConfidence ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.13 GetInferenceFrameId()**

```
virtual int64_t GetInferenceFrameId ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.14 GetInferenceResult()**

```
virtual int64_t GetInferenceResult ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.15 GetLinePitch()**

```
virtual int64_t GetLinePitch ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.16 GetLineStatusAll()**

```
virtual int64_t GetLineStatusAll ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.17 GetOffsetX()**

```
virtual int64_t GetOffsetX ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.18 GetOffsetY()**

```
virtual int64_t GetOffsetY ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.19 GetPartSelector()**

```
virtual int64_t GetPartSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.20 GetPixelDynamicRangeMax()**

```
virtual int64_t GetPixelDynamicRangeMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.21 GetPixelDynamicRangeMin()**

```
virtual int64_t GetPixelDynamicRangeMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.22 GetScan3dAxisMax()**

```
virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.23 GetScan3dAxisMin()**

```
virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.24 GetScan3dCoordinateOffset()**

```
virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.25 GetScan3dCoordinateReferenceValue()**

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.26 GetScan3dCoordinateScale()**

```
virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.27 GetScan3dInvalidDataValue()**

```
virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.28 GetScan3dTransformValue()**

```
virtual float64_t GetScan3dTransformValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.29 GetScanLineSelector()**

```
virtual int64_t GetScanLineSelector ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.30 GetSequencerSetActive()**

```
virtual int64_t GetSequencerSetActive ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.31 GetSerialDataLength()**

```
virtual int64_t GetSerialDataLength ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.32 GetStreamChannelID()**

```
virtual int64_t GetStreamChannelID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.33 GetTimerValue()**

```
virtual float64_t GetTimerValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.34 GetTimestamp()**

```
virtual int64_t GetTimestamp ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.35 GetTimestampLatchValue()**

```
virtual int64_t GetTimestampLatchValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.36 GetTransferBlockID()**

```
virtual int64_t GetTransferBlockID ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.37 GetTransferQueueCurrentBlockCount()**

```
virtual int64_t GetTransferQueueCurrentBlockCount ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.38 GetWidth()**

```
virtual int64_t GetWidth ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

**14.94.3.39 SetChunks()**

```
virtual void SetChunks (   
    GenApi::INodeMap & pNodeMap ) [pure virtual]
```

Implemented in [ChunkData](#).

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

- include/Interface/IChunkData.h

## 14.95 IDataStream Class Reference

### Public Member Functions

- virtual ~IDataStream ()
- virtual StreamTypeEnum GetStreamType () const =0
- virtual void AnnounceImage (size\_t size)=0
- virtual void AnnounceImage (size\_t size, void \*pPrivate)=0
- virtual void AnnounceImage (size\_t size, void \*pData, void \*pPrivate)=0
- virtual void RevokeImages ()=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 ReleaseImage (const uint64\_t imageID)=0
- virtual void FlushQueueAllDiscard ()=0
- virtual bool IsStreaming ()=0
- virtual void KillBufferEvent ()=0
- virtual bool IsImageInUse (const uint64\_t imageID)=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 i← InfoCmd)=0
- virtual size\_t GetBufferInfoSizeType (GenTL::BUFFER\_HANDLE hBuffer, GenTL::BUFFER\_INFO\_CMD i← InfoCmd)=0
- virtual uint64\_t GetBufferInfoUInt64Type (GenTL::BUFFER\_HANDLE hBuffer, GenTL::BUFFER\_INFO\_CMD i← InfoCmd)=0
- virtual bool GetBufferInfoBool8Type (GenTL::BUFFER\_HANDLE hBuffer, GenTL::BUFFER\_INFO\_CMD i← InfoCmd)=0
- virtual void RegisterImageEventHandler (IImageEventHandler &imageEventHandler, EventPollingOptions pollingOption)=0
- virtual void UnregisterImageEventHandler (IImageEventHandler &imageEventHandler)=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 ()

#### 14.95.1 Constructor & Destructor Documentation

**14.95.1.1 ~IDataStream()**

```
virtual ~IDataStream ( ) [inline], [virtual]
```

**14.95.1.2 IDataStream()**

```
IDataStream ( ) [inline], [protected]
```

**14.95.2 Member Function Documentation****14.95.2.1 AnnounceImage() [1/3]**

```
virtual void AnnounceImage (
    size_t size ) [pure virtual]
```

**14.95.2.2 AnnounceImage() [2/3]**

```
virtual void AnnounceImage (
    size_t size,
    void * pData,
    void * pPrivate ) [pure virtual]
```

**14.95.2.3 AnnounceImage() [3/3]**

```
virtual void AnnounceImage (
    size_t size,
    void * pPrivate ) [pure virtual]
```

**14.95.2.4 AttachBuffer()**

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    GenApi::SingleChunkData_t * ChunkData,
    int64_t NumChunks ) [pure virtual]
```

**14.95.2.5 CleanupChunkAdapter()**

```
virtual void CleanupChunkAdapter ( ) [pure virtual]
```

**14.95.2.6 FlushQueueAllDiscard()**

```
virtual void FlushQueueAllDiscard ( ) [pure virtual]
```

**14.95.2.7 GetBufferChunkData()**

```
virtual GenTL::GC_ERROR GetBufferChunkData (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::SINGLE_CHUNK_DATA * pChunkData,
    size_t * piNumChunks ) [pure virtual]
```

**14.95.2.8 GetBufferInfoBool8Type()**

```
virtual bool GetBufferInfoBool8Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

**14.95.2.9 GetBufferInfoPtrType()**

```
virtual void* GetBufferInfoPtrType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

**14.95.2.10 GetBufferInfoSizeType()**

```
virtual size_t GetBufferInfoSizeType (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

**14.95.2.11 GetBufferInfoUInt64Type()**

```
virtual uint64_t GetBufferInfoUInt64Type (
    GenTL::BUFFER_HANDLE hBuffer,
    GenTL::BUFFER_INFO_CMD iInfoCmd ) [pure virtual]
```

**14.95.2.12 GetDeviceNodeMap()**

```
virtual GenApi::INodeMap* GetDeviceNodeMap ( ) const [pure virtual]
```

**14.95.2.13 GetNextImage()**

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout ) [pure virtual]
```

**14.95.2.14 GetNextImageInternal()**

```
virtual ImagePtr GetNextImageInternal (
    void ** ppPrivate,
    uint64_t grabTimeout ) [pure virtual]
```

**14.95.2.15 GetNodeMap()**

```
virtual GenApi::INodeMap& GetNodeMap ( ) const [pure virtual]
```

**14.95.2.16 GetNumImagesInUse()**

```
virtual unsigned int GetNumImagesInUse ( ) const [pure virtual]
```

**14.95.2.17 GetPort()**

```
virtual Port& GetPort ( ) const [pure virtual]
```

**14.95.2.18 GetStreamInfoBool8Type()**

```
virtual bool GetStreamInfoBool8Type (
    GenTL::STREAM_INFO_CMD iInfoCmd) [pure virtual]
```

**14.95.2.19 GetStreamInfoSizeType()**

```
virtual size_t GetStreamInfoSizeType (
    GenTL::STREAM_INFO_CMD iInfoCmd) [pure virtual]
```

**14.95.2.20 GetStreamType()**

```
virtual StreamTypeEnum GetStreamType () const [pure virtual]
```

**14.95.2.21 InitChunkAdapter()**

```
virtual void InitChunkAdapter (
    GenApi::INodeMap & nodemap) [pure virtual]
```

**14.95.2.22 IsCRCCheckEnabled()**

```
virtual bool IsCRCCheckEnabled () const [pure virtual]
```

**14.95.2.23 IsImageInUse()**

```
virtual bool IsImageInUse (
    const uint64_t imageID) [pure virtual]
```

**14.95.2.24 IsStreaming()**

```
virtual bool IsStreaming () [pure virtual]
```

**14.95.2.25 KillBufferEvent()**

```
virtual void KillBufferEvent ( ) [pure virtual]
```

**14.95.2.26 RegisterImageEventHandler()**

```
virtual void RegisterImageEventHandler (
    IImageEventHandler & imageEventHandler,
    EventPollingOptions pollingOption ) [pure virtual]
```

**14.95.2.27 ReleaseImage()**

```
virtual void ReleaseImage (
    const uint64_t imageID ) [pure virtual]
```

**14.95.2.28 RevokeImages()**

```
virtual void RevokeImages ( ) [pure virtual]
```

**14.95.2.29 StartStream()**

```
virtual void StartStream (
    const unsigned int stream_index = 0 ) [pure virtual]
```

**14.95.2.30 StopStream()**

```
virtual void StopStream ( ) [pure virtual]
```

**14.95.2.31 TransportLayerStreamInfo()**

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const [pure virtual]
```

#### 14.95.2.32 UnregisterImageEventHandler()

```
virtual void UnregisterImageEventHandler (
    IImageEventHandler & imageEventHandler ) [pure virtual]
```

#### 14.95.2.33 WaitOnImageEvent()

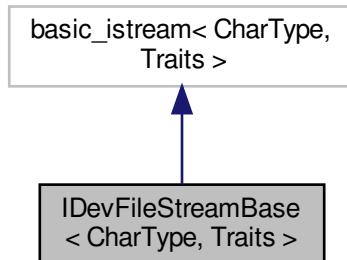
```
virtual void WaitOnImageEvent (
    uint64_t timeout ) [pure virtual]
```

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

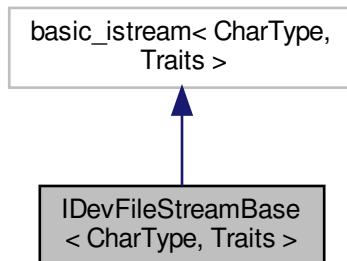
- include/Interface/IStream.h

## 14.96 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.*

### 14.96.1 Member Typedef Documentation

#### 14.96.1.1 filebuf\_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf_type
```

#### 14.96.1.2 ios\_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

#### 14.96.1.3 istream\_type

```
typedef std::basic_istream<CharType, Traits> istream_type
```

### 14.96.2 Member Function Documentation

#### 14.96.2.1 close()

```
void close ( ) [inline]
```

Close the file on the device.

### 14.96.2.2 is\_open()

```
bool is_open ( ) const [inline]
```

### 14.96.2.3 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> | <a href="#">NodeMap</a> of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
| <i>pFileName</i>  | Name of the file to open                                                                           |
| <i>mode</i>       | open mode                                                                                          |

### 14.96.2.4 rdbuf()

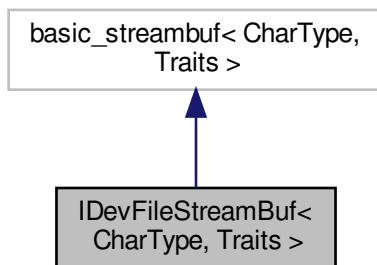
```
filebuf_type* rdbuf ( ) const [inline]
```

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

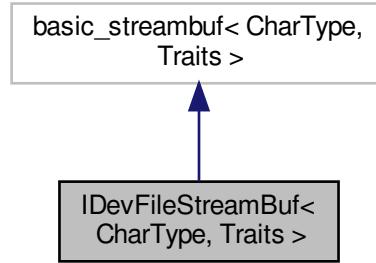
- include/SpinGenApi/[Filestream.h](#)

## 14.97 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 \(int\\_type c\)](#)

### 14.97.1 Constructor & Destructor Documentation

#### 14.97.1.1 IDevFileStreamBuf()

```
IDevFileStreamBuf ( ) [inline]
```

#### 14.97.1.2 ~IDevFileStreamBuf()

```
~IDevFileStreamBuf ( ) [inline]
```

## 14.97.2 Member Function Documentation

### 14.97.2.1 close()

```
filebuf_type* close ( ) [inline]
```

### 14.97.2.2 is\_open()

```
bool is_open ( ) const [inline]
```

### 14.97.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

### 14.97.2.4 pbackfail()

```
int_type pbackfail (
    int_type c ) [inline], [protected]
```

### 14.97.2.5 underflow()

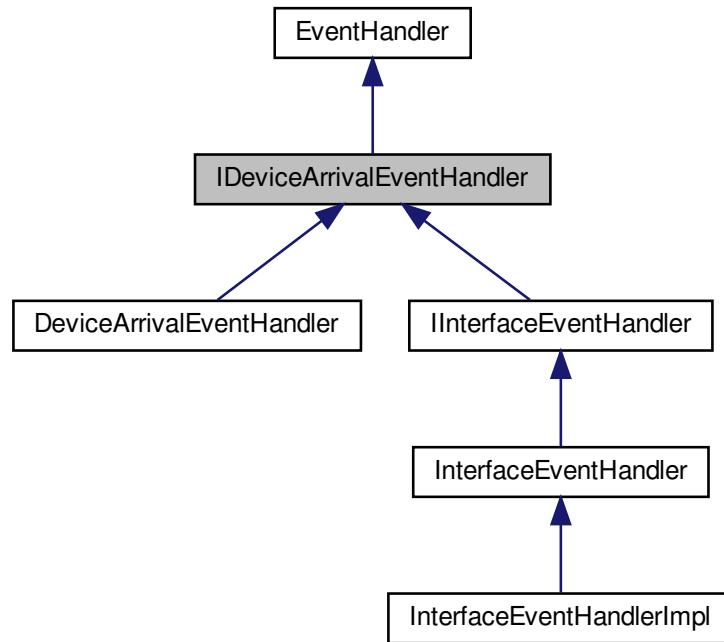
```
int_type underflow ( ) [inline], [protected]
```

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

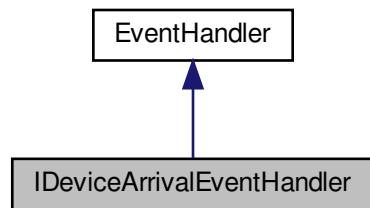
- include/SpinGenApi/Filestream.h

## 14.98 IDeviceArrivalEventHandler Class Reference

Inheritance diagram for IDeviceArrivalEventHandler:



Collaboration diagram for IDeviceArrivalEventHandler:



### Public Member Functions

- virtual [~IDeviceArrivalEventHandler \(\)](#)
- virtual void [OnDeviceArrival \(uint64\\_t serialNumber\)=0](#)

## Protected Member Functions

- `IDeviceArrivalEventHandler ()`
- `IDeviceArrivalEventHandler (const IDeviceArrivalEventHandler &)`
- `IDeviceArrivalEventHandler & operator= (const IDeviceArrivalEventHandler &)`

## Additional Inherited Members

### 14.98.1 Constructor & Destructor Documentation

#### 14.98.1.1 `~IDeviceArrivalEventHandler()`

```
virtual ~IDeviceArrivalEventHandler ( ) [inline], [virtual]
```

#### 14.98.1.2 `IDeviceArrivalEventHandler() [1/2]`

```
IDeviceArrivalEventHandler ( ) [inline], [protected]
```

#### 14.98.1.3 `IDeviceArrivalEventHandler() [2/2]`

```
IDeviceArrivalEventHandler (
    const IDeviceArrivalEventHandler & ) [inline], [protected]
```

### 14.98.2 Member Function Documentation

#### 14.98.2.1 `OnDeviceArrival()`

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in `DeviceArrivalEventHandler`, `InterfaceEventHandler`, `IInterfaceEventHandler`, `InterfaceEventHandlerImpl`, and `InterfaceEventHandlerImpl`.

### 14.98.2.2 operator=()

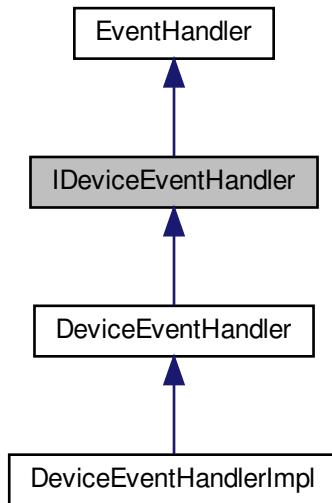
```
IDeviceArrivalEventHandler& operator= (
    const IDeviceArrivalEventHandler & ) [protected]
```

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

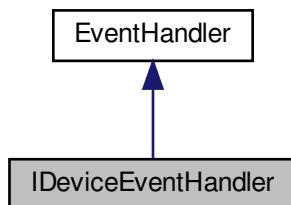
- include/Interface/IDeviceArrivalEventHandler.h

## 14.99 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

### 14.99.1 Constructor & Destructor Documentation

#### 14.99.1.1 ~IDeviceEventHandler()

```
virtual ~IDeviceEventHandler ( ) [inline], [virtual]
```

#### 14.99.1.2 IDeviceEventHandler() [1/2]

```
IDeviceEventHandler ( ) [inline], [protected]
```

#### 14.99.1.3 IDeviceEventHandler() [2/2]

```
IDeviceEventHandler (
    const IDeviceEventHandler & ) [inline], [protected]
```

### 14.99.2 Member Function Documentation

#### 14.99.2.1 GetDeviceEventId()

```
virtual uint64_t GetDeviceEventId ( ) const [pure virtual]
```

Implemented in [DeviceEventHandler](#).

#### 14.99.2.2 GetDeviceEventName()

```
virtual GenICam::gcstring GetDeviceEventName () const [pure virtual]
```

Implemented in [DeviceEventHandler](#).

#### 14.99.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Implemented in [DeviceEventHandler](#), and [DeviceEventHandlerImpl](#).

#### 14.99.2.4 operator=(\*)

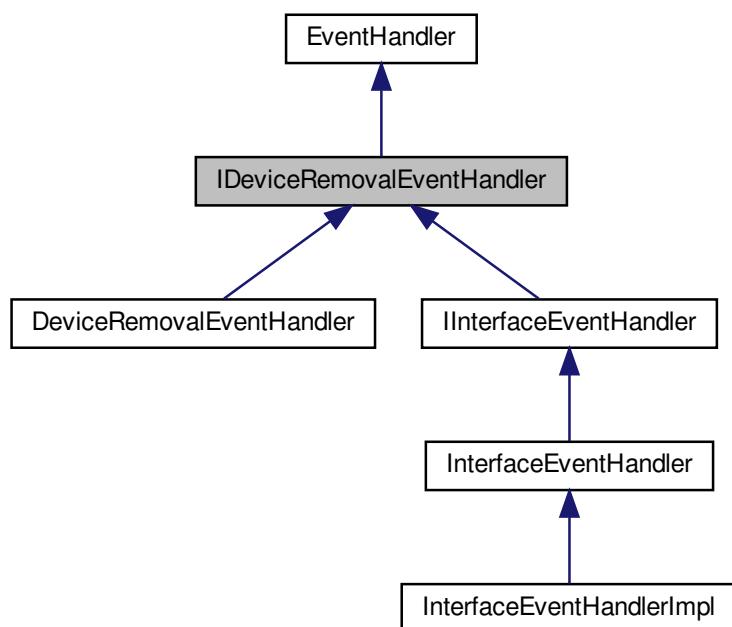
```
IDeviceEventHandler& operator= (
    const IDeviceEventHandler & ) [protected]
```

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

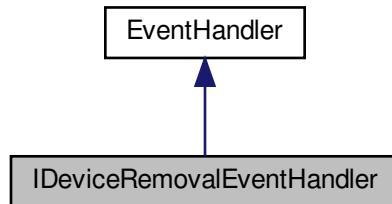
- [include/Interface/IDeviceEventHandler.h](#)

## 14.100 IDeviceRemovalEventHandler Class Reference

Inheritance diagram for IDeviceRemovalEventHandler:



Collaboration diagram for IDeviceRemovalEventHandler:



## Public Member Functions

- virtual `~IDeviceRemovalEventHandler ()`
- virtual void `OnDeviceRemoval (uint64_t serialNumber)=0`

## Protected Member Functions

- `IDeviceRemovalEventHandler ()`
- `IDeviceRemovalEventHandler (const IDeviceRemovalEventHandler &)`
- `IDeviceRemovalEventHandler & operator= (const IDeviceRemovalEventHandler &)`

## Additional Inherited Members

### 14.100.1 Constructor & Destructor Documentation

#### 14.100.1.1 `~IDeviceRemovalEventHandler()`

```
virtual ~IDeviceRemovalEventHandler ( ) [inline], [virtual]
```

#### 14.100.1.2 `IDeviceRemovalEventHandler() [1/2]`

```
IDeviceRemovalEventHandler ( ) [inline], [protected]
```

**14.100.1.3 IDeviceRemovalEventHandler() [2/2]**

```
IODeviceRemovalEventHandler (
    const IDeviceRemovalEventHandler & ) [inline], [protected]
```

**14.100.2 Member Function Documentation****14.100.2.1 OnDeviceRemoval()**

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in [InterfaceEventHandler](#), [DeviceRemovalEventHandler](#), [IIInterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

**14.100.2.2 operator=( )**

```
IODeviceRemovalEventHandler& operator= (
    const IDeviceRemovalEventHandler & ) [protected]
```

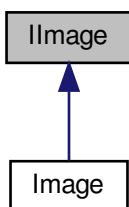
The documentation for this class was generated from the following file:

- include/Interface/IODeviceRemovalEventHandler.h

**14.101 IIImage Class Reference**

The interface file for [Image](#).

Inheritance diagram for IIImage:



## Public Member Functions

- virtual ~IlImage ()
- virtual ColorProcessingAlgorithm GetColorProcessing () const =0
- virtual ImagePtr Convert (PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const =0
- virtual void Convert (ImagePtr destinationImage, PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) 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 Release ()=0
- virtual uint64\_t GetID () 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 PayloadTypeIDs GetTLPayloadType () const =0
- virtual uint64\_t GetTLPixelFormat () const =0
- virtual PixelFormatNamespaceID GetTLPixelFormatNamespace () const =0
- virtual GenICam::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=FROM\_FILE\_EXT)=0
- virtual void Save (const char \*pFilename, PNGOption &pOption)=0
- virtual void Save (const char \*pFilename, PPMOption &pOption)=0
- virtual void Save (const char \*pFilename, PGMOpti&n>on &pOption)=0
- virtual void Save (const char \*pFilename, TIFFOption &pOption)=0
- virtual void Save (const char \*pFilename, JPEGOption &pOption)=0
- virtual void Save (const char \*pFilename, JPG2Option &pOption)=0
- virtual void Save (const char \*pFilename, BMPOption &pOption)=0
- virtual const ChunkData & GetChunkData () const =0
- virtual void CalculateStatistics (ImageStatistics &pStatistics)=0
- virtual bool HasCRC () const =0
- virtual bool CheckCRC () const =0
- virtual size\_t GetImageSize () const =0
- virtual bool IsInUse ()=0
- virtual ImageStatus GetImageStatus () const =0

## Protected Member Functions

- `IImage ()`
- virtual `ImageData * GetImageData () const =0`

## Friends

- class `Stream`

### 14.101.1 Detailed Description

The interface file for `Image`.

### 14.101.2 Constructor & Destructor Documentation

#### 14.101.2.1 `~IImage()`

```
virtual ~IImage ( ) [inline], [virtual]
```

#### 14.101.2.2 `IImage()`

```
IImage ( ) [inline], [protected]
```

### 14.101.3 Member Function Documentation

#### 14.101.3.1 `CalculateStatistics()`

```
virtual void CalculateStatistics (
    ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in `Image`.

#### 14.101.3.2 `CheckCRC()`

```
virtual bool CheckCRC ( ) const [pure virtual]
```

Implemented in `Image`.

**14.101.3.3 Convert() [1/2]**

```
virtual void Convert (
    ImagePtr destinationImage,
    PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.4 Convert() [2/2]**

```
virtual ImagePtr Convert (
    PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.5 DeepCopy()**

```
virtual void DeepCopy (
    const ImagePtr pSrcImage ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.6 GetBitsPerPixel()**

```
virtual size_t GetBitsPerPixel ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.7 GetBufferSize()**

```
virtual size_t GetBufferSize ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.8 GetChunkData()**

```
virtual const ChunkData& GetChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.9 GetChunkLayoutId()**

```
virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.10 GetColorProcessing()**

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.11 GetData()**

```
virtual void* GetData ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.12 GetDataAbsoluteMax()**

```
virtual float GetDataAbsoluteMax ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.13 GetDataAbsoluteMin()**

```
virtual float GetDataAbsoluteMin ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.14 GetFrameID()**

```
virtual uint64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.15 GetHeight()**

```
virtual size_t GetHeight ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.16 GetID()**

```
virtual uint64_t GetID ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.17 GetImageData()**

```
virtual ImageData* GetImageData ( ) const [protected], [pure virtual]
```

Implemented in [Image](#).

**14.101.3.18 GetImageSize()**

```
virtual size_t GetImageSize ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.19 GetImageStatus()**

```
virtual ImageStatus GetImageStatus ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.20 GetNumChannels()**

```
virtual size_t GetNumChannels ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.21 GetPayloadType()**

```
virtual size_t GetPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.22 GetPixelFormat()**

```
virtual PixelFormatEnums GetPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.23 GetPixelFormatIntType()**

```
virtual PixelFormatIntType GetPixelFormatIntType ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.24 GetPixelFormatName()**

```
virtual GenICam::gcstring GetPixelFormatName ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.25 GetPrivateData()**

```
virtual void* GetPrivateData ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.26 GetStride()**

```
virtual size_t GetStride ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.27 GetTimeStamp()**

```
virtual uint64_t GetTimeStamp ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.28 GetTLPayloadType()**

```
virtual PayloadTypeInfoIDs GetTLPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.29 GetTLPixelFormat()**

```
virtual uint64_t GetTLPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.30 GetTLPixelFormatNamespace()**

```
virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.31 GetValidPayloadSize()**

```
virtual size_t GetValidPayloadSize ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.32 GetWidth()**

```
virtual size_t GetWidth ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.33 GetXOffset()**

```
virtual size_t GetXOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.34 GetXPadding()**

```
virtual size_t GetXPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.35 GetYOffset()**

```
virtual size_t GetYOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.36 GetYPadding()**

```
virtual size_t GetYPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.37 HasCRC()**

```
virtual bool HasCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.38 IsIncomplete()**

```
virtual bool IsIncomplete( ) const [pure virtual]
```

Implemented in [Image](#).

**14.101.3.39 IsInUse()**

```
virtual bool IsInUse( ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.40 Release()**

```
virtual void Release( ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.41 ResetImage() [1/2]**

```
virtual void ResetImage( size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.42 ResetImage() [2/2]**

```
virtual void ResetImage( size_t width, size_t height, size_t offsetX, size_t offsetY, PixelFormatEnums pixelFormat, void * pData ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.43 Save() [1/8]**

```
virtual void Save (
    const char * pFilename,
    BMPOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.44 Save() [2/8]**

```
virtual void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.45 Save() [3/8]**

```
virtual void Save (
    const char * pFilename,
    JPEGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.46 Save() [4/8]**

```
virtual void Save (
    const char * pFilename,
    JPG2Option & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.47 Save() [5/8]**

```
virtual void Save (
    const char * pFilename,
    PGMOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.48 Save() [6/8]**

```
virtual void Save (
    const char * pFilename,
    PNGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.49 Save() [7/8]**

```
virtual void Save (
    const char * pFilename,
    PPMOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.3.50 Save() [8/8]**

```
virtual void Save (
    const char * pFilename,
    TIFFOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

**14.101.4 Friends And Related Function Documentation****14.101.4.1 Stream**

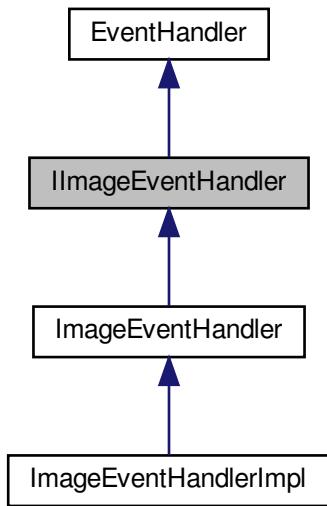
```
friend class Stream [friend]
```

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

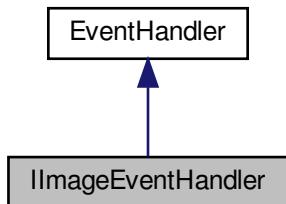
- include/Interface/[IImage.h](#)

## 14.102 IImageEventHandler Class Reference

Inheritance diagram for IImageEventHandler:



Collaboration diagram for IImageEventHandler:



### Public Member Functions

- virtual `~IImageEventHandler ()`
- virtual void `OnImageEvent (ImagePtr image)=0`

### Protected Member Functions

- `IImageEventHandler ()`
- `IImageEventHandler (const IImageEventHandler &)`
- `IImageEventHandler & operator= (const IImageEventHandler &)`

## Additional Inherited Members

### 14.102.1 Constructor & Destructor Documentation

#### 14.102.1.1 ~IImageEventHandler()

```
virtual ~IImageEventHandler ( ) [inline], [virtual]
```

#### 14.102.1.2 IImageEventHandler() [1/2]

```
IImageEventHandler ( ) [inline], [protected]
```

#### 14.102.1.3 IImageEventHandler() [2/2]

```
IImageEventHandler (
    const IImageEventHandler & ) [inline], [protected]
```

### 14.102.2 Member Function Documentation

#### 14.102.2.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

Implemented in [ImageEventHandler](#), [ImageEventHandlerImpl](#), and [ImageEventHandlerImpl](#).

#### 14.102.2.2 operator=( )

```
IImageEventHandler& operator= (
    const IImageEventHandler & ) [protected]
```

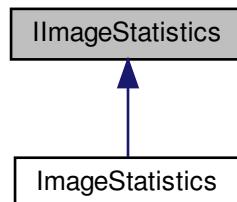
The documentation for this class was generated from the following file:

- include/Interface/IImageEventHandler.h

## 14.103 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 \(const IImageStatistics &\)](#)

### 14.103.1 Detailed Description

The interface file for image statistics.

## 14.103.2 Constructor & Destructor Documentation

### 14.103.2.1 ~IImageStatistics()

```
virtual ~IImageStatistics ( ) [inline], [virtual]
```

### 14.103.2.2 IImageStatistics() [1/2]

```
IImageStatistics ( ) [inline], [protected]
```

### 14.103.2.3 IImageStatistics() [2/2]

```
IImageStatistics (   
     const IImageStatistics & ) [inline], [protected]
```

## 14.103.3 Member Function Documentation

### 14.103.3.1 DisableAll()

```
virtual void DisableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

### 14.103.3.2 EnableAll()

```
virtual void EnableAll ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

### 14.103.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.4 EnableHSLOnly()**

```
virtual void EnableHSLOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.5 EnableRGBOnly()**

```
virtual void EnableRGBOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.6 GetChannelStatus()**

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.7 GetHistogram()**

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.8 GetMean()**

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.9 GetNumPixelValues()**

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.10 GetPixelValueRange()**

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.11 GetRange()**

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

**14.103.3.12 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](#).

#### 14.103.3.13 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [pure virtual]
```

Implemented in [ImageStatistics](#).

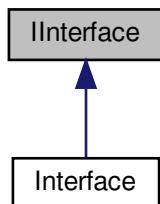
The documentation for this class was generated from the following file:

- include/Interface/[IImageStatistics.h](#)

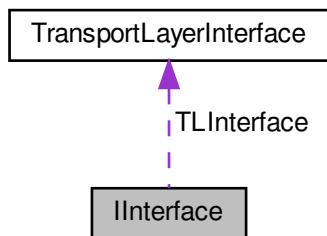
## 14.104 IIInterface Class Reference

The interface file for [Interface](#).

Inheritance diagram for IIInterface:



Collaboration diagram for IIInterface:



## 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 `SystemImpl`

### 14.104.1 Detailed Description

The interface file for `Interface`.

### 14.104.2 Constructor & Destructor Documentation

#### 14.104.2.1 `~IInterface()`

```
virtual ~IInterface ( ) [inline], [virtual]
```

**14.104.2.2 IInterface() [1/2]**

```
IInterface () [inline], [protected]
```

**14.104.2.3 IInterface() [2/2]**

```
IInterface (
    const IInterface & ) [inline], [protected]
```

**14.104.3 Member Function Documentation****14.104.3.1 GetCameras()**

```
virtual CameraList GetCameras (
    bool updateCameras = true ) const [pure virtual]
```

Implemented in [Interface](#).

**14.104.3.2 GetTLNodeMap()**

```
virtual GenApi::INodeMap& GetTLNodeMap () const [pure virtual]
```

Implemented in [Interface](#).

**14.104.3.3 IsInUse()**

```
virtual bool IsInUse () const [pure virtual]
```

Implemented in [Interface](#).

**14.104.3.4 IsValid()**

```
virtual bool IsValid () [pure virtual]
```

Implemented in [Interface](#).

#### 14.104.3.5 operator=( )

```
IInterface& operator= (
    const IInterface & ) [protected]
```

#### 14.104.3.6 RegisterEventHandler()

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [Interface](#).

#### 14.104.3.7 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](#).

#### 14.104.3.8 UnregisterEventHandler()

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [Interface](#).

#### 14.104.3.9 UpdateCameras()

```
virtual bool UpdateCameras ( ) [pure virtual]
```

Implemented in [Interface](#).

### 14.104.4 Friends And Related Function Documentation

#### 14.104.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

#### 14.104.4.2 SystemImpl

```
friend class SystemImpl [friend]
```

### 14.104.5 Member Data Documentation

#### 14.104.5.1 m\_pInterfaceData

```
InterfaceData* m_pInterfaceData [protected]
```

#### 14.104.5.2 TLInterface

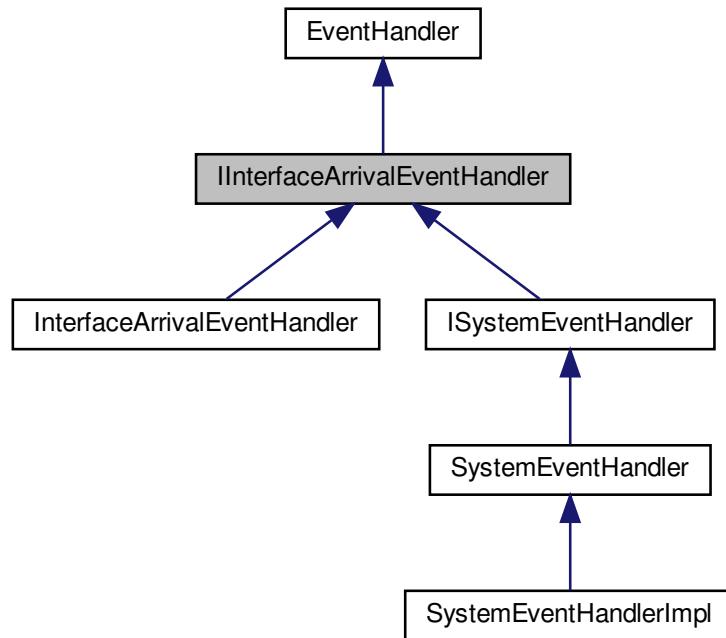
[TransportLayerInterface](#) TLInterface

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

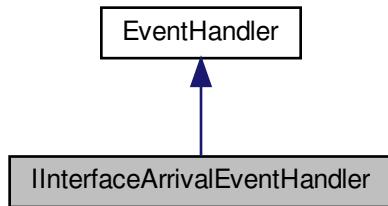
- include/Interface/IIInterface.h

## 14.105 IIInterfaceArrivalEventHandler Class Reference

Inheritance diagram for IIInterfaceArrivalEventHandler:



Collaboration diagram for IIInterfaceArrivalEventHandler:



### Public Member Functions

- virtual `~IIInterfaceArrivalEventHandler ()`
- virtual void `OnInterfaceArrival (std::string interfaceID)=0`

## Protected Member Functions

- `IIInterfaceArrivalEventHandler ()`
- `IIInterfaceArrivalEventHandler (const IIInterfaceArrivalEventHandler &)`
- `IIInterfaceArrivalEventHandler & operator= (const IIInterfaceArrivalEventHandler &)`

## Additional Inherited Members

### 14.105.1 Constructor & Destructor Documentation

#### 14.105.1.1 `~IIInterfaceArrivalEventHandler()`

```
virtual ~IIInterfaceArrivalEventHandler ( ) [inline], [virtual]
```

#### 14.105.1.2 `IIInterfaceArrivalEventHandler()` [1/2]

```
IIInterfaceArrivalEventHandler ( ) [inline], [protected]
```

#### 14.105.1.3 `IIInterfaceArrivalEventHandler()` [2/2]

```
IIInterfaceArrivalEventHandler (
    const IIInterfaceArrivalEventHandler & ) [inline], [protected]
```

### 14.105.2 Member Function Documentation

#### 14.105.2.1 `OnInterfaceArrival()`

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

Implemented in `InterfaceArrivalEventHandler`, `SystemEventHandler`, `ISystemEventHandler`, and `SystemEventHandlerImpl`.

#### 14.105.2.2 `operator=( )`

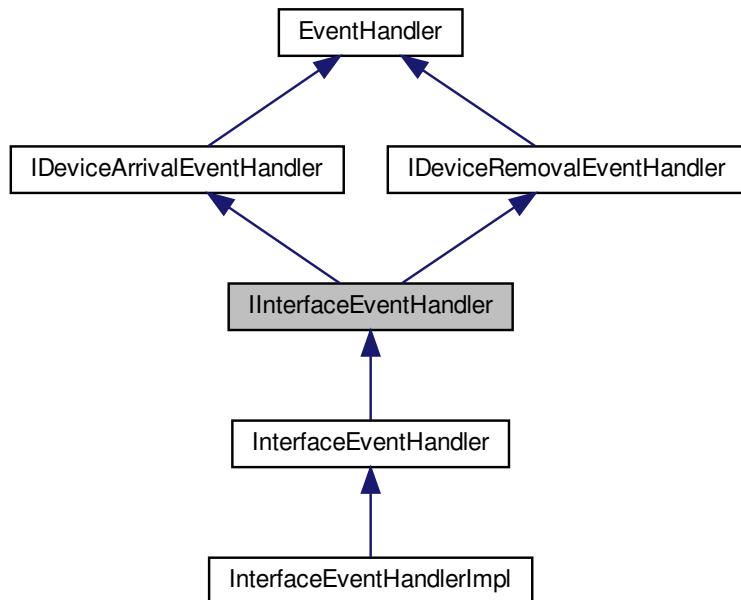
```
IInterfaceArrivalEventHandler& operator= (
    const IInterfaceArrivalEventHandler & ) [protected]
```

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

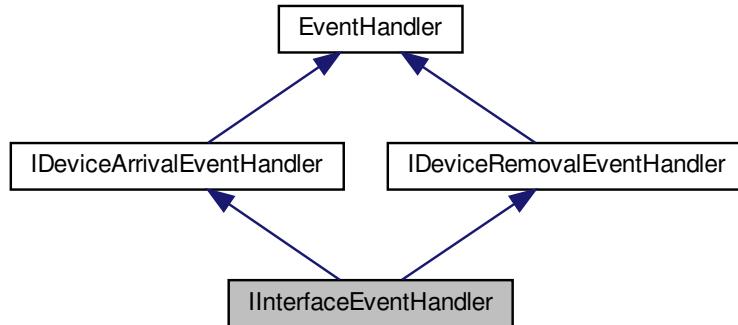
- include/Interface/IInterfaceArrivalEventHandler.h

## 14.106 IInterfaceEventHandler Class Reference

Inheritance diagram for IInterfaceEventHandler:



Collaboration diagram for IIInterfaceEventHandler:



## Public Member Functions

- virtual `~IIInterfaceEventHandler ()`
- virtual void `OnDeviceArrival (uint64_t serialNumber)=0`
- virtual void `OnDeviceRemoval (uint64_t serialNumber)=0`

## Protected Member Functions

- `IIInterfaceEventHandler ()`
- `IIInterfaceEventHandler (const IIInterfaceEventHandler &)`
- `IIInterfaceEventHandler & operator= (const IIInterfaceEventHandler &)`

## Additional Inherited Members

### 14.106.1 Constructor & Destructor Documentation

#### 14.106.1.1 `~IIInterfaceEventHandler()`

```
virtual ~IIInterfaceEventHandler ( ) [inline], [virtual]
```

#### 14.106.1.2 `IIInterfaceEventHandler()` [1/2]

```
IIInterfaceEventHandler ( ) [inline], [protected]
```

#### 14.106.1.3 `IInterfaceEventHandler()` [2/2]

```
IInterfaceEventHandler (
    const IInterfaceEventHandler & ) [inline], [protected]
```

### 14.106.2 Member Function Documentation

#### 14.106.2.1 `OnDeviceArrival()`

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IDeviceArrivalEventHandler](#).

Implemented in [InterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

#### 14.106.2.2 `OnDeviceRemoval()`

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IDeviceRemovalEventHandler](#).

Implemented in [InterfaceEventHandler](#), [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

#### 14.106.2.3 `operator=()`

```
IInterfaceEventHandler& operator=
    const IInterfaceEventHandler & ) [protected]
```

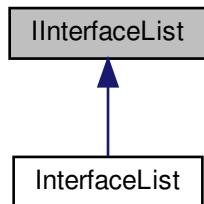
The documentation for this class was generated from the following file:

- include/Interface/IInterfaceEventHandler.h

## 14.107 IIInterfaceList Class Reference

The interface file for [InterfaceList](#) class.

Inheritance diagram for IIInterfaceList:



### Public Member Functions

- virtual [~IIInterfaceList](#) (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

### Protected Member Functions

- [IIInterfaceList](#) (void)
- [IIInterfaceList](#) (const [IIInterfaceList](#) &)
- [IIInterfaceList & operator=](#) (const [IIInterfaceList](#) &)

### Protected Attributes

- [InterfaceListData \\* m\\_pInterfaceListData](#)

### 14.107.1 Detailed Description

The interface file for [InterfaceList](#) class.

### 14.107.2 Constructor & Destructor Documentation

**14.107.2.1 ~IInterfaceList()**

```
virtual ~IInterfaceList (
    void ) [inline], [virtual]
```

**14.107.2.2 IInterfaceList() [1/2]**

```
IInterfaceList (
    void ) [inline], [protected]
```

**14.107.2.3 IInterfaceList() [2/2]**

```
IInterfaceList (
    const IInterfaceList & ) [inline], [protected]
```

**14.107.3 Member Function Documentation****14.107.3.1 Clear()**

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [InterfaceList](#).

**14.107.3.2 GetByIndex()**

```
virtual InterfacePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [InterfaceList](#).

**14.107.3.3 GetSize()**

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [InterfaceList](#).

#### 14.107.3.4 operator=( )

```
IInterfaceList& operator= (
    const IInterfaceList & ) [protected]
```

#### 14.107.3.5 operator[]( )

```
virtual InterfacePtr operator[ ] (
    unsigned int index ) [pure virtual]
```

Implemented in [InterfaceList](#).

### 14.107.4 Member Data Documentation

#### 14.107.4.1 m\_pInterfaceListData

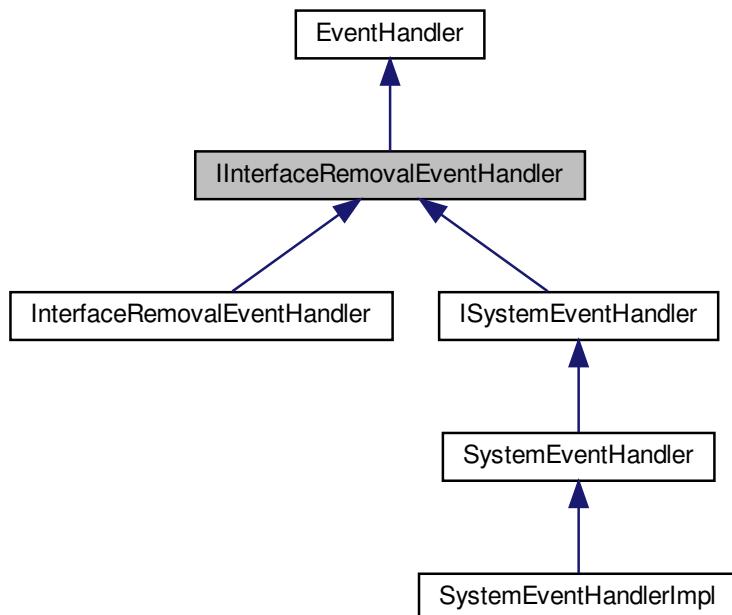
```
InterfaceListData* m_pInterfaceListData [protected]
```

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

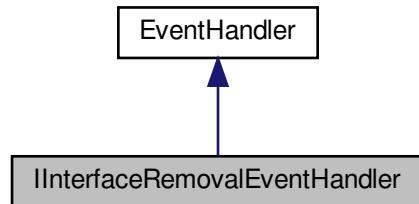
- include/Interface/IInterfaceList.h

## 14.108 IInterfaceRemovalEventHandler Class Reference

Inheritance diagram for IInterfaceRemovalEventHandler:



Collaboration diagram for `IInterfaceRemovalEventHandler`:



## Public Member Functions

- virtual `~IInterfaceRemovalEventHandler()`
- virtual void `OnInterfaceRemoval (std::string interfaceID)=0`

## Protected Member Functions

- `IInterfaceRemovalEventHandler ()`
- `IInterfaceRemovalEventHandler (const IInterfaceRemovalEventHandler &)`
- `IInterfaceRemovalEventHandler & operator= (const IInterfaceRemovalEventHandler &)`

## Additional Inherited Members

### 14.108.1 Constructor & Destructor Documentation

#### 14.108.1.1 `~IInterfaceRemovalEventHandler()`

```
virtual ~IInterfaceRemovalEventHandler ( ) [inline], [virtual]
```

#### 14.108.1.2 `IInterfaceRemovalEventHandler()` [1/2]

```
IInterfaceRemovalEventHandler ( ) [inline], [protected]
```

**14.108.1.3 IIInterfaceRemovalEventHandler() [2/2]**

```
IIInterfaceRemovalEventHandler (
    const IIInterfaceRemovalEventHandler & ) [inline], [protected]
```

**14.108.2 Member Function Documentation****14.108.2.1 OnInterfaceRemoval()**

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implemented in [SystemEventHandler](#), [InterfaceRemovalEventHandler](#), [ISystemEventHandler](#), and [SystemEventHandlerImpl](#).

**14.108.2.2 operator=( )**

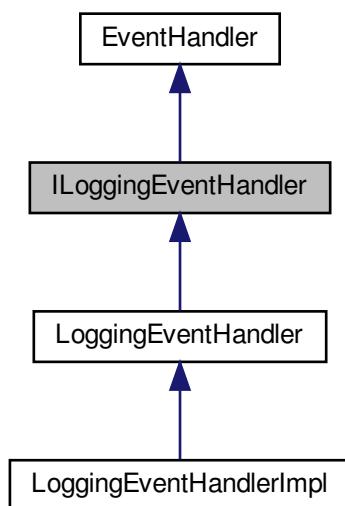
```
IIInterfaceRemovalEventHandler& operator= (
    const IIInterfaceRemovalEventHandler & ) [protected]
```

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

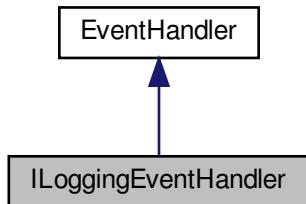
- [include/Interface/IIInterfaceRemovalEventHandler.h](#)

**14.109 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

### 14.109.1 Constructor & Destructor Documentation

#### 14.109.1.1 `~ILoggingEventHandler()`

```
virtual ~ILoggingEventHandler ( ) [inline], [virtual]
```

#### 14.109.1.2 `ILoggingEventHandler()` [1/2]

```
ILoggingEventHandler ( ) [inline], [protected]
```

### 14.109.1.3 `ILoggingEventHandler()` [2/2]

```
ILoggingEventHandler (
    const ILoggingEventHandler & ) [inline], [protected]
```

## 14.109.2 Member Function Documentation

### 14.109.2.1 `OnLogEvent()`

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

Implemented in [LoggingEventHandler](#).

### 14.109.2.2 `operator=()`

```
ILoggingEventHandler& operator= (
    const ILoggingEventHandler & ) [protected]
```

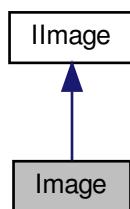
The documentation for this class was generated from the following file:

- include/Interface/ILoggingEventHandler.h

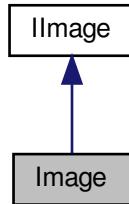
## 14.110 Image Class Reference

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.*
- `ImagePtr Convert (Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const`  
*Converts the current image buffer to the specified output pixel format and stores the result in the specified image.*
- `void Convert (ImagePtr destinationImage, Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm colorAlgorithm=DEFAULT) const`  
*Converts the current image buffer to the specified output pixel format and stores the result in the specified destination 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 Release ()`
- `uint64_t GetID () const`  
*Gets a unique ID for this image.*
- `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.*
- `PayloadTypeInfoIDs GetTLPayloadType () const`  
*Gets the GenTL specific payload type that was transmitted.*
- `uint64_t GetTLPixelFormat () const`  
*Gets the pixel format of the image.*
- `PixelFormatNamespaceID 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=FROM_FILE_EXT)`  
*Saves the image to the specified file name with the file format specified.*
- `void Save (const char *pFilename, PNGOption &pOption)`  
*Saves the image to the specified file name with the options specified.*
- `void Save (const char *pFilename, PPMOption &pOption)`  
*Saves the image to the specified file name with the options specified.*
- `void Save (const char *pFilename, PGMOption &pOption)`  
*Saves the image to the specified file name with the options specified.*
- `void Save (const char *pFilename, TIFFOption &pOption)`  
*Saves the image to the specified file name with the options specified.*
- `void Save (const char *pFilename, JPEGOption &pOption)`  
*Saves the image to the specified file name with the options specified.*

- **Saves the image to the specified file name with the options specified.**
- void [Save](#) (const char \*pFilename, [JPG2Option](#) &pOption)
  - Saves the image to the specified file name with the options specified.*
- void [Save](#) (const char \*pFilename, [BMPOption](#) &pOption)
  - Saves the image to the specified file name with the options specified.*
- 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 void [SetDefaultColorProcessing](#) ([ColorProcessingAlgorithm](#) colorAlgorithm)
  - Sets the default color processing algorithm.*
- static [ColorProcessingAlgorithm](#) [GetDefaultColorProcessing](#) ()
  - Gets the default color processing algorithm.*
- static const char \* [GetImageStatusDescription](#) ([ImageStatus](#) status)
  - Returns a string describing the meaning of the status enum.*

## Protected Member Functions

- [ImageData](#) \* [GetImageData](#) () const
- [Image](#) ()
  - [Image](#) ()
  - [Image](#) (const [ImagePtr](#) image)
  - [Image](#) (size\_t width, size\_t height, size\_t offsetX, size\_t offsetY, [PixelFormatEnums](#) pixelFormat, void \*pData)
- [ImagePtr](#) [CreateShared](#) () const
- void [DeepCopy](#) (const [Image](#) &pSrcImage)
- void [Convert](#) ([PixelFormatEnums](#) format, [Image](#) &pDestImage, [ColorProcessingAlgorithm](#) colorAlgorithm=[DEFAULT](#)) const

## Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageFiler](#)
- class [ImageStatsCalculator](#)
- class [ImageUtilityImpl](#)
- class [ImageUtilityPolarizationImpl](#)

### 14.110.1 Detailed Description

The image object class.

### 14.110.2 Constructor & Destructor Documentation

#### 14.110.2.1 ~Image()

```
virtual ~Image( ) [virtual]
```

Virtual destructor.

#### 14.110.2.2 Image() [1/3]

```
Image( ) [protected]
```

#### 14.110.2.3 Image() [2/3]

```
Image(   
        const ImagePtr image ) [protected]
```

#### 14.110.2.4 Image() [3/3]

```
Image(   
        size_t width,  
        size_t height,  
        size_t offsetX,  
        size_t offsetY,  
        PixelFormatEnums pixelFormat,  
        void * pData ) [protected]
```

### 14.110.3 Member Function Documentation

#### 14.110.3.1 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

|                          |                             |
|--------------------------|-----------------------------|
| <code>pStatistics</code> | The statistics of an image. |
|--------------------------|-----------------------------|

Implements [IImage](#).

#### 14.110.3.2 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 [IImage](#).

#### 14.110.3.3 Convert() [1/3]

```
void Convert (
    ImagePtr destinationImage,
    Spinnaker::PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [virtual]
```

Converts the current image buffer to the specified output pixel format and stores the result in the specified destination image.

The destination image buffer size must be sufficient to store the converted image data.

##### See also

[Create\(size\\_t width, size\\_t height, size\\_t offsetX, size\\_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void\\* pData\)](#)

### Parameters

|                         |                                                                        |
|-------------------------|------------------------------------------------------------------------|
| <i>destinationImage</i> | Destination image where the converted output result will be stored.    |
| <i>format</i>           | Output format of the converted image.                                  |
| <i>colorAlgorithm</i>   | Optional color processing algorithm for producing the converted image. |

Implements [IImage](#).

#### 14.110.3.4 Convert() [2/3]

```
void Convert (
    PixelFormatEnums format,
    Image & pDestImage,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [protected]
```

#### 14.110.3.5 Convert() [3/3]

```
ImagePtr Convert (
    Spinnaker::PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [virtual]
```

Converts the current image buffer to the specified output pixel format and stores the result in the specified image.

The destination image does not need to be configured in any way before the call is made.

### See also

[PixelFormatEnums](#)

### Parameters

|                       |                                                                       |
|-----------------------|-----------------------------------------------------------------------|
| <i>format</i>         | Output format of the converted image.                                 |
| <i>colorAlgorithm</i> | Optional color processing algorithm for producing the converted image |

### Returns

The converted image.

Implements [IImage](#).

#### 14.110.3.6 Create() [1/3]

```
static ImagePtr Create () [static]
```

Create an image object.

#### 14.110.3.7 Create() [2/3]

```
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 |
|--------------|-------------------------|

#### 14.110.3.8 Create() [3/3]

```
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             |

#### 14.110.3.9 CreateShared()

```
ImagePtr CreateShared ( ) const [protected]
```

#### 14.110.3.10 DeepCopy() [1/2]

```
void DeepCopy (
    const Image & pSrcImage ) [protected]
```

**14.110.3.11 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**

|                        |                                                  |
|------------------------|--------------------------------------------------|
| <code>pSrcImage</code> | The <a href="#">Image</a> to copy the data from. |
|------------------------|--------------------------------------------------|

Implements [IImage](#).

**14.110.3.12 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 [IImage](#).

**14.110.3.13 GetBufferSize()**

```
size_t GetBufferSize ( ) const  [virtual]
```

Gets the size of the buffer associated with the image in bytes.

**Returns**

The size of the buffer, in bytes.

Implements [IImage](#).

#### 14.110.3.14 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](#).

#### 14.110.3.15 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](#).

#### 14.110.3.16 GetColorProcessing()

```
ColorProcessingAlgorithm GetColorProcessing ( ) const [virtual]
```

Gets the color algorithm used to produce the image.

**See also**

[Convert\(\)](#)

**Returns**

The color processing algorithm used to produce the image.

Implements [IImage](#).

**14.110.3.17 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 [IImage](#).

**14.110.3.18 GetDataAbsoluteMax()**

```
float GetDataAbsoluteMax ( ) const [virtual]
```

Get the value for which no image data will exceed.

**Returns**

the maximum theoretical image data value

Implements [IImage](#).

**14.110.3.19 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 [IImage](#).

#### 14.110.3.20 GetDefaultColorProcessing()

```
static ColorProcessingAlgorithm GetDefaultColorProcessing ( ) [static]
```

Gets the default color processing algorithm.

##### See also

[SetDefaultColorProcessing\(\)](#)

##### Returns

The default color processing algorithm.

#### 14.110.3.21 GetFrameID()

```
uint64_t GetFrameID ( ) const [virtual]
```

Gets the frame ID for this image.

##### Returns

The frame ID.

Implements [IImage](#).

#### 14.110.3.22 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](#).

**14.110.3.23 GetID()**

```
uint64_t GetID ( ) const [virtual]
```

Gets a unique ID for this image.

Each image in a steam will have a unique ID to help identify it.

**Returns**

The 64 bit unique id for this image.

Implements [IImage](#).

**14.110.3.24 GetImageData()**

```
ImageData* GetImageData ( ) const [protected], [virtual]
```

Implements [IImage](#).

**14.110.3.25 GetImageSize()**

```
size_t GetImageSize ( ) const [virtual]
```

Returns the size of the image.

**Returns**

The image size in bytes.

Implements [IImage](#).

**14.110.3.26 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](#).

#### 14.110.3.27 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.

#### 14.110.3.28 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 [IImage](#).

#### 14.110.3.29 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 [IImage](#).

### 14.110.3.30 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 [Convert\(\)](#) function.

See also

[Convert\(\)](#)

Returns

enum value representing the PixelFormat.

Implements [IImage](#).

### 14.110.3.31 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](#).

### 14.110.3.32 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](#).

**14.110.3.33 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](#).

**14.110.3.34 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](#).

**14.110.3.35 GetTimeStamp()**

```
uint64_t GetTimeStamp ( ) const [virtual]
```

Gets the time stamp for the image in nanoseconds.

**Returns**

The time stamp of the image.

Implements [IImage](#).

#### 14.110.3.36 GetTLPayloadType()

```
PayloadTypeInfoIDs 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](#).

#### 14.110.3.37 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](#).

#### 14.110.3.38 GetTLPixelFormatNamespace()

```
PixelFormatNamespaceID 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](#).

#### 14.110.3.39 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. [GetBufferSize\(\)](#) returns the total size of bytes allocated for the image.

See also

[GetBufferSize\(\)](#)

Returns

size\_t value representing valid payload.

Implements [IImage](#).

#### 14.110.3.40 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](#).

#### 14.110.3.41 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 [IImage](#).

**14.110.3.42 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 [IImage](#).

**14.110.3.43 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 [IImage](#).

**14.110.3.44 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 [IImage](#).

**14.110.3.45 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 [IImage](#).

**14.110.3.46 IsCompressed()**

```
bool IsCompressed () const
```

Returns a boolean value indicating whether this image is compressed.

**Returns**

Returns true if image is compressed, false otherwise.

**14.110.3.47 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](#).

**14.110.3.48 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](#).

**14.110.3.49 Release()**

```
void Release ( ) [virtual]
```

Implements [IImage](#).

**14.110.3.50 ResetImage() [1/2]**

```
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 [IImage](#).

**14.110.3.51 ResetImage() [2/2]**

```
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 [IImage](#).

#### 14.110.3.52 Save() [1/8]

```
void Save (
    const char * pFilename,
    BMPOption & pOption ) [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](#).

#### 14.110.3.53 Save() [2/8]

```
void Save (
    const char * pFilename,
    ImageFileFormat format = FROM\_FILE\_EXT ) [virtual]
```

Saves the image to the specified file name with the file format specified.

##### Parameters

|                  |                              |
|------------------|------------------------------|
| <i>pFilename</i> | Filename to save image with. |
| <i>format</i>    | File format to save in.      |

Implements [IImage](#).

#### 14.110.3.54 Save() [3/8]

```
void Save (
    const char * pFilename,
    JPEGOption & pOption ) [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](#).

#### 14.110.3.55 Save() [4/8]

```
void Save (
    const char * pFilename,
    JPG2Option & pOption ) [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](#).

#### 14.110.3.56 Save() [5/8]

```
void Save (
    const char * pFilename,
    PGMOption & pOption ) [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](#).

#### 14.110.3.57 Save() [6/8]

```
void Save (
    const char * pFilename,
    PNGOption & pOption ) [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](#).

#### 14.110.3.58 Save() [7/8]

```
void Save (
    const char * pFilename,
    PPMOption & pOption ) [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](#).

#### 14.110.3.59 Save() [8/8]

```
void Save (
    const char * pFilename,
    TIFFOption & pOption ) [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](#).

#### 14.110.3.60 SetDefaultColorProcessing()

```
static void SetDefaultColorProcessing (
    ColorProcessingAlgorithm colorAlgorithm ) [static]
```

Sets the default color processing algorithm.

This method will be used for any image with the DEFAULT algorithm set. The method used is determined at the time of the [Convert\(\)](#) call, therefore the most recent execution of this function will take precedence. The default setting is shared within the current process.

**Parameters**

|                       |                                        |
|-----------------------|----------------------------------------|
| <i>colorAlgorithm</i> | The color processing algorithm to set. |
|-----------------------|----------------------------------------|

**See also**

[GetDefaultColorProcessing\(\)](#)

## 14.110.4 Friends And Related Function Documentation

### 14.110.4.1 IDataStream

```
friend class IDataStream [friend]
```

### 14.110.4.2 ImageConverter

```
friend class ImageConverter [friend]
```

### 14.110.4.3 ImageFiler

```
friend class ImageFiler [friend]
```

### 14.110.4.4 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

### 14.110.4.5 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

#### 14.110.4.6 ImageUtilityPolarizationImpl

```
friend class ImageUtilityPolarizationImpl [friend]
```

#### 14.110.4.7 Stream

```
friend class Stream [friend]
```

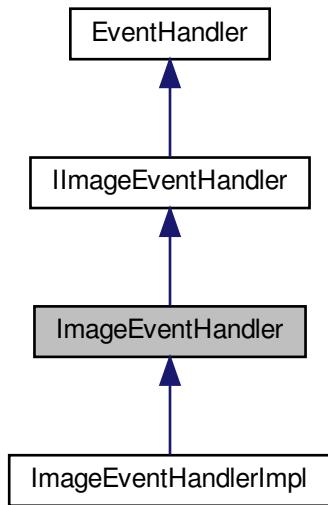
The documentation for this class was generated from the following file:

- [include/Image.h](#)

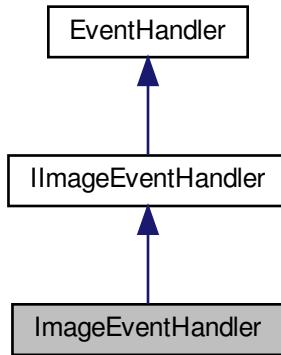
## 14.111 ImageEventHandler Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEventHandler:



Collaboration diagram for ImageEventHandler:



## Public Member Functions

- [ImageEventHandler \(\)](#)  
*Default Constructor.*
- virtual [~ImageEventHandler \(\)](#)  
*Virtual Destructor.*
- virtual void [OnImageEvent \(ImagePtr image\)=0](#)  
*Image event callback.*

## Protected Member Functions

- [ImageEventHandler & operator= \(const ImageEventHandler &\)](#)  
*Assignment operator.*

## Additional Inherited Members

### 14.111.1 Detailed Description

A handler for capturing image arrival events.

### 14.111.2 Constructor & Destructor Documentation

#### 14.111.2.1 `ImageEventHandler()`

```
ImageEventHandler ( )
```

Default Constructor.

#### 14.111.2.2 `~ImageEventHandler()`

```
virtual ~ImageEventHandler ( ) [virtual]
```

Virtual Destructor.

### 14.111.3 Member Function Documentation

#### 14.111.3.1 `OnImageEvent()`

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

`Image` event callback.

Parameters

|                    |                                  |
|--------------------|----------------------------------|
| <code>image</code> | The <code>ImagePtr</code> object |
|--------------------|----------------------------------|

Implements `IImageEventHandler`.

Implemented in `ImageEventHandlerImpl`, and `ImageEventHandlerImpl`.

#### 14.111.3.2 `operator=( )`

```
ImageEventHandler& operator=
    const ImageEventHandler & ) [protected]
```

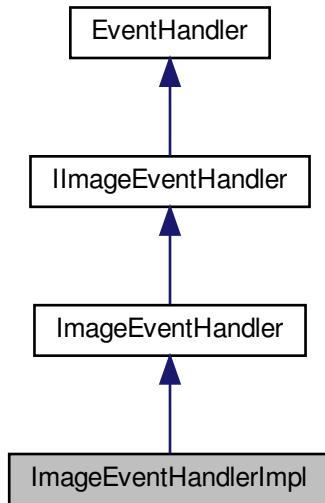
Assignment operator.

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

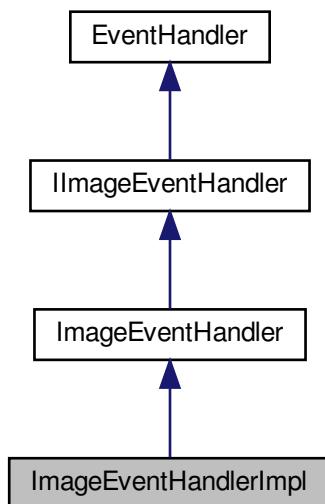
- `include/ImageEventHandler.h`

## 14.112 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

### 14.112.1 Constructor & Destructor Documentation

#### 14.112.1.1 `ImageEventHandlerImpl()` [1/2]

```
ImageEventHandlerImpl (
    string deviceSerial ) [inline]
```

#### 14.112.1.2 `~ImageEventHandlerImpl()` [1/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

#### 14.112.1.3 `ImageEventHandlerImpl()` [2/2]

```
ImageEventHandlerImpl (
    CameraPtr pCam ) [inline]
```

#### 14.112.1.4 `~ImageEventHandlerImpl()` [2/2]

```
~ImageEventHandlerImpl ( ) [inline]
```

### 14.112.2 Member Function Documentation

**14.112.2.1 getImageCount()**

```
int getImageCount ( ) [inline]
```

**14.112.2.2 getMaxImages()**

```
int getMaxImages ( ) [inline]
```

**14.112.2.3 OnImageEvent() [1/2]**

```
void OnImageEvent (
    ImagePtr image ) [virtual]
```

Image event callback.

**Parameters**

|              |                     |
|--------------|---------------------|
| <i>image</i> | The ImagePtr object |
|--------------|---------------------|

Implements [ImageEventHandler](#).

**14.112.2.4 OnImageEvent() [2/2]**

```
void OnImageEvent (
    ImagePtr image ) [inline], [virtual]
```

Image event callback.

**Parameters**

|              |                     |
|--------------|---------------------|
| <i>image</i> | The ImagePtr object |
|--------------|---------------------|

Implements [ImageEventHandler](#).

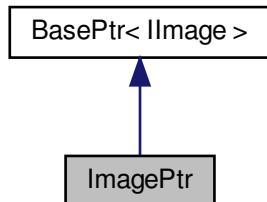
The documentation for this class was generated from the following files:

- src/AcquisitionMultipleCameraRecovery/[AcquisitionMultipleCameraRecovery.cpp](#)
- src/ImageEvents/[ImageEvents.cpp](#)

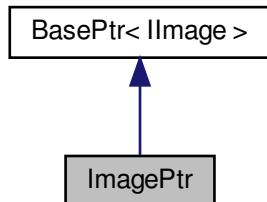
## 14.113 ImagePtr Class Reference

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

### 14.113.1 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.

### 14.113.2 Constructor & Destructor Documentation

#### 14.113.2.1 [ImagePtr\(\)](#) [1/4]

```
ImagePtr ( )
```

Default constructor.

#### 14.113.2.2 [ImagePtr\(\)](#) [2/4]

```
ImagePtr ( const int )
```

Default constructor with argument.

#### 14.113.2.3 [ImagePtr\(\)](#) [3/4]

```
ImagePtr ( const long )
```

Default constructor with argument.

#### 14.113.2.4 [ImagePtr\(\)](#) [4/4]

```
ImagePtr ( const std::nullptr_t )
```

Default constructor with argument.

#### 14.113.2.5 ~ImagePtr()

```
virtual ~ImagePtr (
    void ) [virtual]
```

Virtual destructor.

### 14.113.3 Member Function Documentation

#### 14.113.3.1 operator=()

```
virtual ImagePtr& operator=
    const ImagePtr & ) [virtual]
```

Assignment operator.

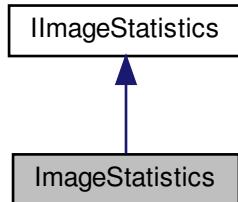
The documentation for this class was generated from the following file:

- [include/ImagePtr.h](#)

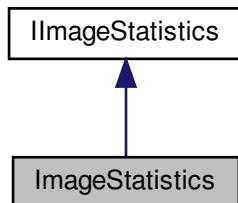
## 14.114 ImageStatistics Class Reference

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 EnableRGBOnly ()`  
*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

### 14.114.1 Detailed Description

Represents image statistics for an image.

## 14.114.2 Constructor & Destructor Documentation

### 14.114.2.1 `ImageStatistics()` [1/2]

```
ImageStatistics ( )
```

Default constructor.

### 14.114.2.2 `~ImageStatistics()`

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

### 14.114.2.3 `ImageStatistics()` [2/2]

```
ImageStatistics (   
    const ImageStatistics & other )
```

Copy constructor.

## 14.114.3 Member Function Documentation

### 14.114.3.1 `DisableAll()`

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements [IImageStatistics](#).

### 14.114.3.2 `EnableAll()`

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements [IImageStatistics](#).

#### 14.114.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements [IImageStatistics](#).

#### 14.114.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements [IImageStatistics](#).

#### 14.114.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements [IImageStatistics](#).

#### 14.114.3.6 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](#).

#### 14.114.3.7 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](#).

#### 14.114.3.8 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](#).

#### 14.114.3.9 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](#).

**14.114.3.10 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](#).

**14.114.3.11 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](#).

**14.114.3.12 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](#).

### 14.114.3.13 operator=()

```
ImageStatistics& operator= (
    const ImageStatistics & other )
```

Assignment operator.

#### Parameters

|              |                                                          |
|--------------|----------------------------------------------------------|
| <i>other</i> | The <a href="#">ImageStatistics</a> object to copy from. |
|--------------|----------------------------------------------------------|

### 14.114.3.14 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](#).

## 14.114.4 Friends And Related Function Documentation

### 14.114.4.1 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

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

- [include/IImageStatistics.h](#)

## 14.115 ImageUtility Class Reference

Static helper functions for the image object class.

### Public Types

- enum [ImageScalingAlgorithm](#) { [NEAREST\\_NEIGHBOR](#) }  
*Image scaling algorithms.*
- enum [SourceDataRange](#) {  
  [IMAGE\\_DATA\\_RANGE](#),  
  [ABSOLUTE\\_DATA\\_RANGE](#),  
  [IMAGE\\_MIN\\_ABSOLUTE\\_MAX](#),  
  [ABSOLUTE\\_MIN\\_IMAGE\\_MAX](#) }  
*Image normalization source data options.*

### Static Public Member Functions

- static [ImagePtr CreateScaled](#) (const [ImagePtr](#) &srclImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)  
*Computes a scaled image using the specified parameters.*
- static void [CreateScaled](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)  
*Computes a scaled image using the specified parameters.*
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=[IMAGE\\_DATA\\_RANGE](#))  
*Computes a normalized image.*
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const double min, const double max, [SourceDataRange](#) srcDataRange=[IMAGE\\_DATA\\_RANGE](#))  
*Computes a normalized image.*
- static [ImagePtr CreateNormalized](#) (const [ImagePtr](#) &srclImage, const double min, const double max, const [PixelFormatEnums](#) destPixelFormat, [SourceDataRange](#) srcDataRange=[IMAGE\\_DATA\\_RANGE](#))  
*Computes a normalized image.*
- static void [CreateNormalized](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, [SourceDataRange](#) srcDataRange=[IMAGE\\_DATA\\_RANGE](#))  
*Computes a normalized image.*
- static void [CreateNormalized](#) (const [ImagePtr](#) &srclImage, [ImagePtr](#) &destImage, const double min, const double max, [SourceDataRange](#) srcDataRange=[IMAGE\\_DATA\\_RANGE](#))  
*Computes a normalized image.*

### 14.115.1 Detailed Description

Static helper functions for the image object class.

### 14.115.2 Member Enumeration Documentation

#### 14.115.2.1 ImageScalingAlgorithm

enum `ImageScalingAlgorithm`

[Image](#) scaling algorithms.

Enumerator

|                  |  |
|------------------|--|
| NEAREST_NEIGHBOR |  |
|------------------|--|

#### 14.115.2.2 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

|                                     |                                                                                    |
|-------------------------------------|------------------------------------------------------------------------------------|
| <code>IMAGE_DATA_RANGE</code>       | Normalize based on the actual max and min values for the source image.             |
| <code>ABSOLUTE_DATA_RANGE</code>    | Normalize based on the theoretical max and min values for the source image.        |
| <code>IMAGE_MIN_ABSOLUTE_MAX</code> | Normalize based on the actual min and theoretical max values for the source image. |
| <code>ABSOLUTE_MIN_IMAGE_MAX</code> | Normalize based on the theoretical min and actual max values for the source image. |

### 14.115.3 Member Function Documentation

#### 14.115.3.1 CreateNormalized() [1/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = 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

#### 14.115.3.2 CreateNormalized() [2/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = 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

**14.115.3.3 CreateNormalized() [3/5]**

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const PixelFormatEnums destPixelFormat,
    SourceDataRange srcDataRange = 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

**14.115.3.4 CreateNormalized() [4/5]**

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const double min,
    const double max,
    SourceDataRange srcDataRange = 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 |

### 14.115.3.5 CreateNormalized() [5/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    SourceDataRange srcDataRange = 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 |

### 14.115.3.6 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            |

### 14.115.3.7 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

|                            |                                                    |
|----------------------------|----------------------------------------------------|
| <code>srclImage</code>     | The source image from which to create scaled image |
| <code>scalingAlg</code>    | The desired image scaling algorithm to use         |
| <code>scalingFactor</code> | 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](#)

## 14.116 ImageUtilityHeatmap Class Reference

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

### Public Types

- enum `HeatmapColor` {
 `HEATMAP_BLACK` = 1,  
`HEATMAP_BLUE` = 2,  
`HEATMAP_CYAN` = 3,  
`HEATMAP_GREEN` = 4,  
`HEATMAP_YELLOW` = 5,  
`HEATMAP_RED` = 6,  
`HEATMAP_WHITE` = 7 }

*Color specifiers for the heatmap color gradient.*

### Static Public Member Functions

- static `ImagePtr CreateHeatmap` (const `ImagePtr` &`srclImage`)  
*Computes a heatmap image.*
- static void `CreateHeatmap` (const `ImagePtr` &`srclImage`, `ImagePtr` &`destlImage`)  
*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.*

### 14.116.1 Detailed Description

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

### 14.116.2 Member Enumeration Documentation

#### 14.116.2.1 HeatmapColor

enum `HeatmapColor`

Color specifiers for the heatmap color gradient.

Enumerator

|                |  |
|----------------|--|
| HEATMAP_BLACK  |  |
| HEATMAP_BLUE   |  |
| HEATMAP_CYAN   |  |
| HEATMAP_GREEN  |  |
| HEATMAP_YELLOW |  |
| HEATMAP_RED    |  |
| HEATMAP_WHITE  |  |

### 14.116.3 Member Function Documentation

#### 14.116.3.1 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

**14.116.3.2 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\(\)](#)

**14.116.3.3 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\(\)](#)

#### 14.116.3.4 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\(\)](#)

#### 14.116.3.5 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.   |

#### 14.116.3.6 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**

|                           |                                                   |
|---------------------------|---------------------------------------------------|
| <code>newLowValue</code>  | New value at which to begin color representation. |
| <code>newHighValue</code> | New value at which to end color representation.   |

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

- [include/ImageUtilityHeatmap.h](#)

## 14.117 ImageUtilityPolarization Class Reference

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

### Public Types

- enum `PolarizationQuadrant` {
   
    `QUADRANT_I0`,
   
    `QUADRANT_I45`,
   
    `QUADRANT_I90`,
   
    `QUADRANT_I135` }

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

### Static Public Member Functions

- static `ImagePtr ExtractPolarQuadrant` (const `ImagePtr` &`srclImage`, const `PolarizationQuadrant` `desiredQuadrant`)
   
*Extracts all pixels of a specified degree of linear polarization into a new image object.*
- static void `ExtractPolarQuadrant` (const `ImagePtr` &`srclImage`, `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` &`srclImage`)
   
*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` &`srclImage`, `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` &`srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
   
*Computes an image representing the overall intensity of light from a polarized image.*
- static void `CreateStokesS0` (const `ImagePtr` &`srclImage`, `ImagePtr` &`destStokesS0Image`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
   
*Computes an image representing the overall intensity of light from a polarized image.*
- static `ImagePtr CreateStokesS1` (const `ImagePtr` &`srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
   
*Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.*
- static void `CreateStokesS1` (const `ImagePtr` &`srclImage`, `ImagePtr` &`destStokesS1Image`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
   
*Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.*

- static `ImagePtr CreateStokesS2` (const `ImagePtr &srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the difference in intensity accepted through the polarizers at 0 and 90 to the horizontal.*
- static void `CreateStokesS2` (const `ImagePtr &srclImage`, `ImagePtr &destStokesS2Image`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.*
- static `ImagePtr CreateDolp` (const `ImagePtr &srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the difference in intensity accepted through the polarizers.*
- static void `CreateDolp` (const `ImagePtr &srclImage`, `ImagePtr &destDolpImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the fraction of incident light intensity in the linear polarization states.*
- static `ImagePtr CreateAolp` (const `ImagePtr &srclImage`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.*
- static void `CreateAolp` (const `ImagePtr &srclImage`, `ImagePtr &destAolpImg`, const `ColorProcessingAlgorithm` `colorProcessingAlg=DEFAULT`)
 

*Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.*

### 14.117.1 Detailed Description

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

### 14.117.2 Member Enumeration Documentation

#### 14.117.2.1 PolarizationQuadrant

```
enum PolarizationQuadrant
```

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

##### Enumerator

|                            |                                 |
|----------------------------|---------------------------------|
| <code>QUADRANT_I0</code>   | The 0 degree of polarization.   |
| <code>QUADRANT_I45</code>  | The 45 degree of polarization.  |
| <code>QUADRANT_I90</code>  | The 90 degree of polarization.  |
| <code>QUADRANT_I135</code> | The 135 degree of polarization. |

### 14.117.3 Member Function Documentation

### 14.117.3.1 CreateAolp() [1/2]

```
static ImagePtr CreateAolp (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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

### 14.117.3.2 CreateAolp() [2/2]

```
static void CreateAolp (
    const ImagePtr & srcImage,
    ImagePtr & destAolpImg,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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                                |

### 14.117.3.3 CreateDolp() [1/2]

```
static ImagePtr CreateDolp (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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

### 14.117.3.4 CreateDolp() [2/2]

```
static void CreateDolp (
    const ImagePtr & srcImage,
    ImagePtr & destDolpImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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                                 |

### 14.117.3.5 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

**14.117.3.6 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 |

**14.117.3.7 CreateStokesS0() [1/2]**

```
static ImagePtr CreateStokesS0 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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

**14.117.3.8 CreateStokesS0() [2/2]**

```
static void CreateStokesS0 (
    const ImagePtr & srcImage,
```

```
ImagePtr & destStokesS0Image,
const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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       |

#### 14.117.3.9 CreateStokesS1() [1/2]

```
static ImagePtr CreateStokesS1 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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

#### 14.117.3.10 CreateStokesS1() [2/2]

```
static void CreateStokesS1 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS1Image,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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       |

**14.117.3.11 CreateStokesS2() [1/2]**

```
static ImagePtr CreateStokesS2 (
    const ImagePtr & srcImage,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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

**14.117.3.12 CreateStokesS2() [2/2]**

```
static void CreateStokesS2 (
    const ImagePtr & srcImage,
    ImagePtr & destStokesS2Image,
    const ColorProcessingAlgorithm colorProcessingAlg = DEFAULT ) [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       |

### 14.117.3.13 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

### 14.117.3.14 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                                        |

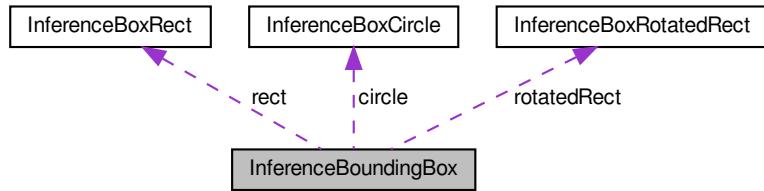
The documentation for this class was generated from the following file:

- [include/ImageUtilityPolarization.h](#)

## 14.118 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`

### 14.118.1 Detailed Description

Inference Bounding Boxes data structure.

### 14.118.2 Member Data Documentation

#### 14.118.2.1 `boxType`

`InferenceBoxType boxType`

#### 14.118.2.2 `circle`

`InferenceBoxCircle circle`

#### 14.118.2.3 `classId`

`int16_t classId`

#### 14.118.2.4 confidence

```
float32_t confidence
```

#### 14.118.2.5 rect

```
InferenceBoxRect rect
```

#### 14.118.2.6 rotatedRect

```
InferenceBoxRotatedRect rotatedRect
```

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

- [include/ChunkDataInference.h](#)

## 14.119 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.*

### 14.119.1 Detailed Description

An inference bounding boxes object which holds information about the detected bounding boxes.

### 14.119.2 Constructor & Destructor Documentation

#### 14.119.2.1 InferenceBoundingBoxResult() [1/3]

```
InferenceBoundingBoxResult ( )
```

Default Constructor.

#### 14.119.2.2 ~InferenceBoundingBoxResult()

```
~InferenceBoundingBoxResult ( )
```

Destructor.

#### 14.119.2.3 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. |

#### 14.119.2.4 InferenceBoundingBoxResult() [3/3]

```
InferenceBoundingBoxResult (
    const InferenceBoundingBoxResult & other )
```

Copy Constructor.

### 14.119.3 Member Function Documentation

#### 14.119.3.1 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. |
|--------------|--------------------------------------|

#### 14.119.3.2 GetBoxCount()

```
int16_t GetBoxCount ( ) const
```

Returns the number of bounding boxes.

#### 14.119.3.3 GetBoxSize()

```
int8_t GetBoxSize ( ) const
```

Returns the number of bytes allocated for one bounding box.

#### 14.119.3.4 GetVersion()

```
int8_t GetVersion ( ) const
```

Returns the bounding box format version number.

#### 14.119.3.5 operator=( )

```
InferenceBoundingBoxResult& operator= (
    const InferenceBoundingBoxResult & rhs )
```

Assignment Operator.

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

- [include/ChunkDataInference.h](#)

## 14.120 InferenceBoxCircle Struct Reference

### Public Attributes

- int16\_t centerXCoord
- int16\_t centerYCoord
- int16\_t radius

#### 14.120.1 Member Data Documentation

##### 14.120.1.1 centerXCoord

```
int16_t centerXCoord
```

##### 14.120.1.2 centerYCoord

```
int16_t centerYCoord
```

##### 14.120.1.3 radius

```
int16_t radius
```

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

- [include/ChunkDataInference.h](#)

## 14.121 InferenceBoxRect Struct Reference

Inference Bounding Box Type Data Structures.

### Public Attributes

- int16\_t topLeftXCoord
- int16\_t topLeftYCoord
- int16\_t bottomRightXCoord
- int16\_t bottomRightYCoord

### 14.121.1 Detailed Description

Inference Bounding Box Type Data Structures.

### 14.121.2 Member Data Documentation

#### 14.121.2.1 bottomRightXCoord

```
int16_t bottomRightXCoord
```

#### 14.121.2.2 bottomRightYCoord

```
int16_t bottomRightYCoord
```

#### 14.121.2.3 topLeftXCoord

```
int16_t topLeftXCoord
```

#### 14.121.2.4 topLeftYCoord

```
int16_t topLeftYCoord
```

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

- [include/ChunkDataInference.h](#)

## 14.122 InferenceBoxRotatedRect Struct Reference

### Public Attributes

- `int16_t topLeftXCoord`
- `int16_t topLeftYCoord`
- `int16_t bottomRightXCoord`
- `int16_t bottomRightYCoord`
- `short rotationAngle`

### 14.122.1 Member Data Documentation

#### 14.122.1.1 bottomRightXCoord

```
int16_t bottomRightXCoord
```

#### 14.122.1.2 bottomRightYCoord

```
int16_t bottomRightYCoord
```

#### 14.122.1.3 rotationAngle

```
short rotationAngle
```

#### 14.122.1.4 topLeftXCoord

```
int16_t topLeftXCoord
```

#### 14.122.1.5 topLeftYCoord

```
int16_t topLeftYCoord
```

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

- [include/ChunkDataInference.h](#)

## 14.123 int64\_avector\_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`

### 14.123.1 Detailed Description

Vector of integers with reference counting.

### 14.123.2 Constructor & Destructor Documentation

#### 14.123.2.1 int64\_autovector\_t() [1/3]

```
int64_autovector_t ( )
```

#### 14.123.2.2 int64\_autovector\_t() [2/3]

```
int64_autovector_t (
    const int64_autovector_t & obj )
```

#### 14.123.2.3 int64\_autovector\_t() [3/3]

```
int64_autovector_t (
    size_t n ) [explicit]
```

**14.123.2.4 ~int64\_autovector\_t()**

```
virtual ~int64_autovector_t (
    void ) [virtual]
```

**14.123.3 Member Function Documentation****14.123.3.1 operator delete()**

```
void operator delete (
    void * pWhere )
```

**14.123.3.2 operator new()**

```
void* operator new (
    size_t uiSize )
```

**14.123.3.3 operator=( )**

```
int64_autovector_t& operator= (
    const int64_autovector_t & obj )
```

**14.123.3.4 operator[]( ) [1/2]**

```
int64_t& operator[] (
    size_t uiIndex )
```

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

```
const int64_t& operator[] (
    size_t uiIndex ) const
```

### 14.123.3.6 size()

```
size_t size ( ) const
```

## 14.123.4 Member Data Documentation

### 14.123.4.1 \_pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

### 14.123.4.2 \_pv

```
std::vector<int64_t>* _pv [protected]
```

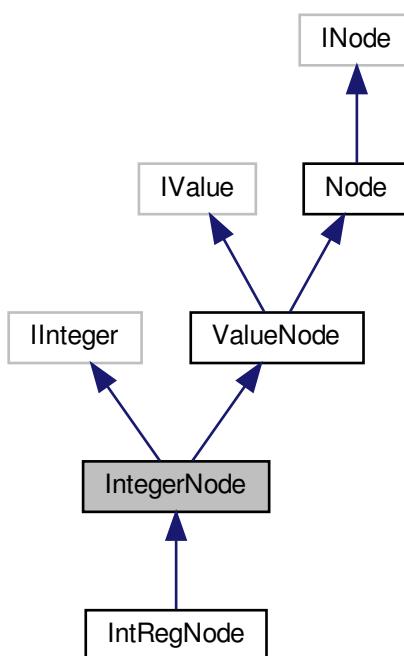
The documentation for this class was generated from the following file:

- include/SpinGenApi/Autovector.h

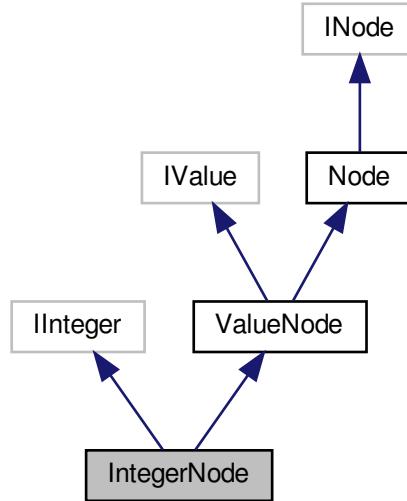
## 14.124 IntegerNode Class Reference

[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 IIInteger & 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_autovector_t GetListOfValidValues (bool bounded=true)`  
*Get list of valid value.*
- `virtual ERepresentation GetRepresentation ()`  
*Get recommended representation.*

- virtual `GenICam::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

### 14.124.1 Detailed Description

`Interface` for string properties.

### 14.124.2 Constructor & Destructor Documentation

#### 14.124.2.1 `IntegerNode()` [1/2]

```
IntegerNode ( )
```

#### 14.124.2.2 `IntegerNode()` [2/2]

```
IntegerNode ( std::shared_ptr< Node::NodeImpl > pInteger )
```

#### 14.124.2.3 `~IntegerNode()`

```
virtual ~IntegerNode ( ) [virtual]
```

### 14.124.3 Member Function Documentation

**14.124.3.1 GetFloatAlias()**

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

**14.124.3.2 GetInc()**

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

Get increment.

**14.124.3.3 GetIncMode()**

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

Get increment mode.

**14.124.3.4 GetListOfValidValues()**

```
virtual int64_avector_t GetListOfValidValues ( bool bounded = true ) [virtual]
```

Get list of valid value.

**14.124.3.5 GetMax()**

```
virtual int64_t GetMax ( ) [virtual]
```

Get maximum value allowed.

**14.124.3.6 GetMin()**

```
virtual int64_t GetMin ( ) [virtual]
```

Get minimum value allowed.

#### 14.124.3.7 GetRepresentation()

```
virtual ERepresentation GetRepresentation ( ) [virtual]
```

Get recommended representation.

#### 14.124.3.8 GetUnit()

```
virtual GenICam::gcstring GetUnit ( ) [virtual]
```

Get the physical unit name.

#### 14.124.3.9 GetValue()

```
virtual int64_t GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

##### Parameters

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

##### Returns

The value read

#### 14.124.3.10 ImposeMax()

```
virtual void ImposeMax (
    int64_t Value ) [virtual]
```

Restrict maximum value.

#### 14.124.3.11 ImposeMin()

```
virtual void ImposeMin (
    int64_t Value ) [virtual]
```

Restrict minimum value.

**14.124.3.12 operator()()**

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

**14.124.3.13 operator\*()**

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

**14.124.3.14 operator=(())**

```
virtual IInteger& operator= (
    int64_t Value ) [virtual]
```

Set node value.

**14.124.3.15 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

**14.124.3.16 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) |

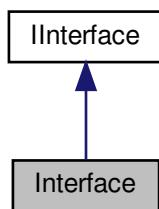
The documentation for this class was generated from the following file:

- include/SpinGenApi/[IntegerNode.h](#)

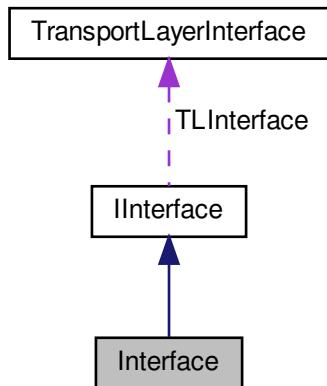
## 14.125 Interface Class Reference

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

### 14.125.1 Detailed Description

An interface object which holds a list of cameras.

### 14.125.2 Constructor & Destructor Documentation

#### 14.125.2.1 ~Interface()

```
virtual ~Interface (
    void ) [virtual]
```

Virtual Destructor.

### 14.125.3 Member Function Documentation

### 14.125.3.1 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](#).

### 14.125.3.2 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](#).

### 14.125.3.3 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](#).

#### 14.125.3.4 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](#).

#### 14.125.3.5 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.

##### Parameters

|                                   |                                                 |
|-----------------------------------|-------------------------------------------------|
| <code>evtHandlerToRegister</code> | The event handler to register for the interface |
|-----------------------------------|-------------------------------------------------|

Implements [IInterface](#).

#### 14.125.3.6 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

|                         |                                                                              |
|-------------------------|------------------------------------------------------------------------------|
| <code>deviceKey</code>  | The Action Command's device key                                              |
| <code>groupKey</code>   | The Action Command's group key                                               |
| <code>groupMask</code>  | The Action Command's group mask                                              |
| <code>actionTime</code> | (Optional) Time when to assert a future action. Zero means immediate action. |

**Parameters**

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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](#).

**14.125.3.7 UnregisterEventHandler()**

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the interface.

**Parameters**

|                               |                                                    |
|-------------------------------|----------------------------------------------------|
| <i>evtHandlerToUnregister</i> | The event handler to unregister from the interface |
|-------------------------------|----------------------------------------------------|

Implements [IInterface](#).

**14.125.3.8 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](#).

#### 14.125.4 Friends And Related Function Documentation

##### 14.125.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

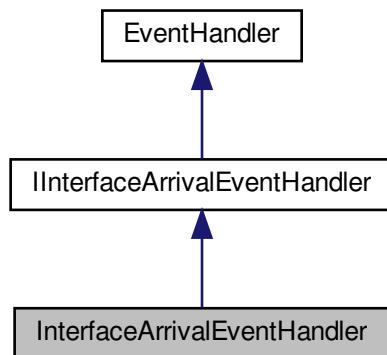
The documentation for this class was generated from the following file:

- [include/Interface.h](#)

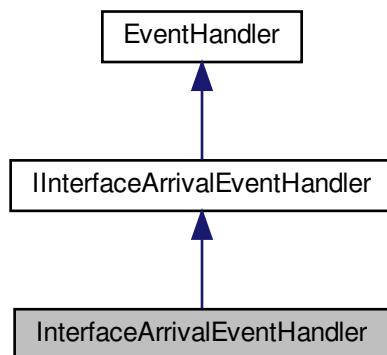
### 14.126 InterfaceArrivalEventHandler Class Reference

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 (std::string interfaceID)=0`  
*Interface arrival event callback.*

## Protected Member Functions

- `InterfaceArrivalEventHandler & operator= (const InterfaceArrivalEventHandler &)`  
*Assignment operator.*

## Additional Inherited Members

### 14.126.1 Detailed Description

An event handler for capturing the interface arrival event.

Note that only GEV interface arrivals are currently handled.

### 14.126.2 Constructor & Destructor Documentation

#### 14.126.2.1 InterfaceArrivalEventHandler()

```
InterfaceArrivalEventHandler ( )
```

Default constructor.

#### 14.126.2.2 ~InterfaceArrivalEventHandler()

```
virtual ~InterfaceArrivalEventHandler ( ) [virtual]
```

Virtual destructor.

### 14.126.3 Member Function Documentation

#### 14.126.3.1 OnInterfaceArrival()

```
virtual void OnInterfaceArrival (
    std::string interfaceID ) [pure virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

**Parameters**

|                    |                                      |
|--------------------|--------------------------------------|
| <i>interfaceID</i> | The ID of the interface that arrived |
|--------------------|--------------------------------------|

Implements [IInterfaceArrivalEventHandler](#).

#### 14.126.3.2 `operator=()`

```
InterfaceArrivalEventHandler& operator= (
    const InterfaceArrivalEventHandler & ) [protected]
```

Assignment operator.

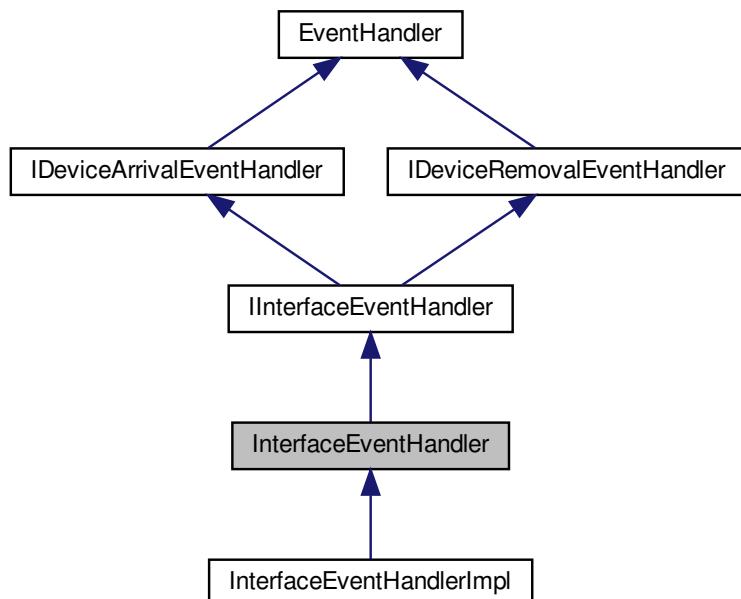
The documentation for this class was generated from the following file:

- [include/InterfaceArrivalEventHandler.h](#)

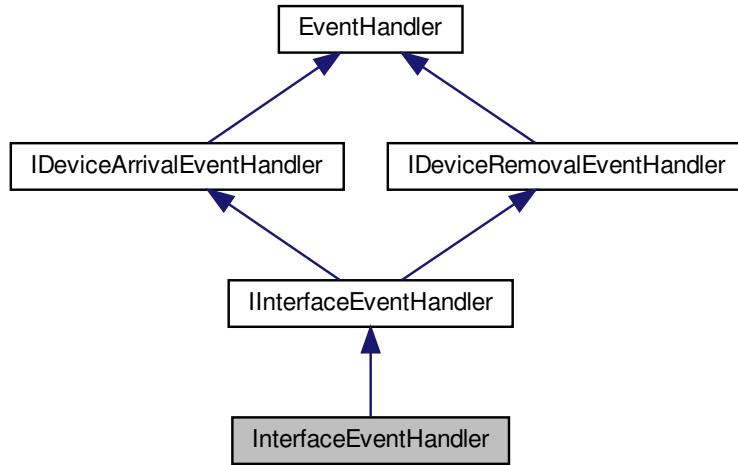
## 14.127 InterfaceEventHandler Class Reference

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 (uint64_t serialNumber)=0`  
*Device arrival event callback.*
- `virtual void OnDeviceRemoval (uint64_t serialNumber)=0`  
*Callback to the device removal event.*

## Protected Member Functions

- `InterfaceEventHandler & operator= (const InterfaceEventHandler &)`  
*Assignment operator.*

## Additional Inherited Members

### 14.127.1 Detailed Description

A handler to device arrival and removal events on all interfaces.

### 14.127.2 Constructor & Destructor Documentation

#### 14.127.2.1 InterfaceEventHandler()

```
InterfaceEventHandler ( )
```

Default constructor.

#### 14.127.2.2 ~InterfaceEventHandler()

```
virtual ~InterfaceEventHandler ( ) [virtual]
```

Virtual destructor.

### 14.127.3 Member Function Documentation

#### 14.127.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Device arrival event callback.

Implements [IInterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

#### 14.127.3.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device removal event.

##### Parameters

|                           |                                         |
|---------------------------|-----------------------------------------|
| <code>serialNumber</code> | The serial number of the removed device |
|---------------------------|-----------------------------------------|

Implements [IInterfaceEventHandler](#).

Implemented in [InterfaceEventHandlerImpl](#), and [InterfaceEventHandlerImpl](#).

### 14.127.3.3 operator=( )

```
InterfaceEventHandler& operator= (
    const InterfaceEventHandler & ) [protected]
```

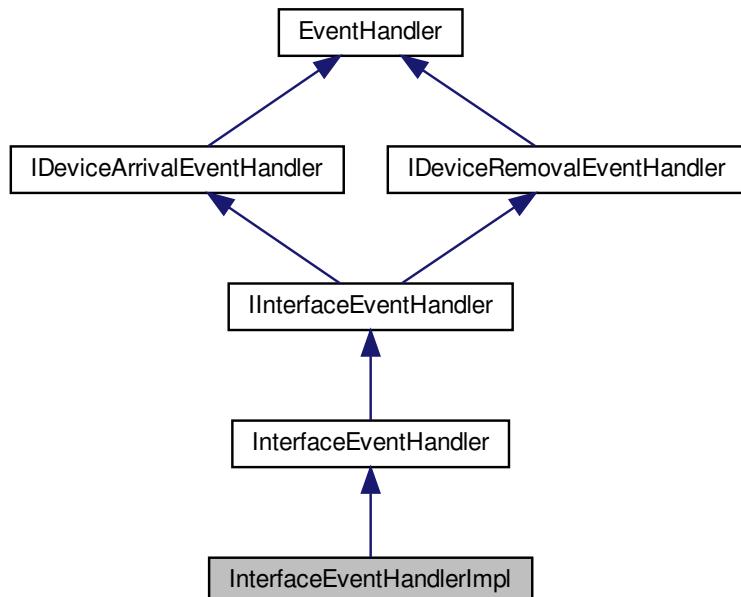
Assignment operator.

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

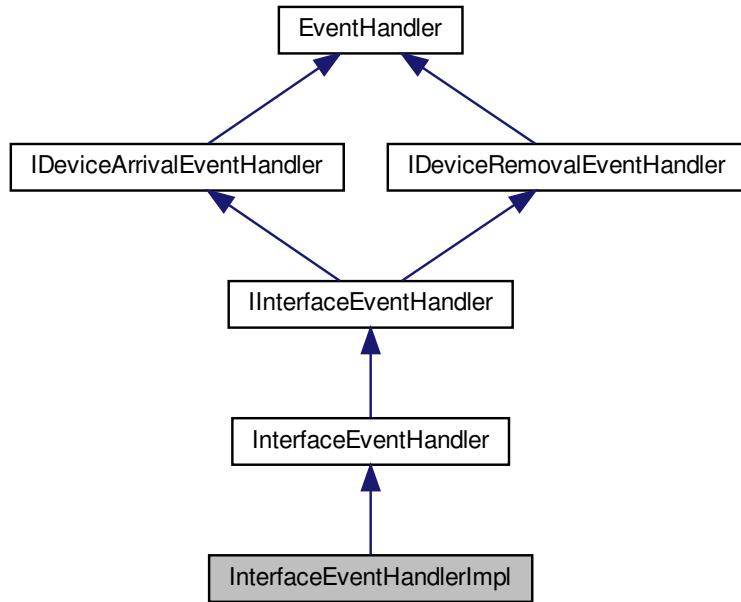
- include/InterfaceEventHandler.h

## 14.128 InterfaceEventHandlerImpl Class Reference

Inheritance diagram for InterfaceEventHandlerImpl:



Collaboration diagram for InterfaceEventHandlerImpl:



## Public Member Functions

- `InterfaceEventHandlerImpl (SystemPtr system)`
- `~InterfaceEventHandlerImpl ()`
- `void OnDeviceArrival (uint64_t deviceSerialNumber)`  
`Device arrival event callback.`
- `void OnDeviceRemoval (uint64_t deviceSerialNumber)`  
`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 (uint64_t deviceSerialNumber)`  
`Device arrival event callback.`
- `void OnDeviceRemoval (uint64_t deviceSerialNumber)`  
`Callback to the device removal event.`
- `std::string GetInterfaceId ()`

## Additional Inherited Members

### 14.128.1 Constructor & Destructor Documentation

**14.128.1.1 InterfaceEventHandlerImpl() [1/3]**

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

**14.128.1.2 ~InterfaceEventHandlerImpl() [1/2]**

```
~InterfaceEventHandlerImpl ( ) [inline]
```

**14.128.1.3 InterfaceEventHandlerImpl() [2/3]**

```
InterfaceEventHandlerImpl (
    SystemPtr system ) [inline]
```

**14.128.1.4 InterfaceEventHandlerImpl() [3/3]**

```
InterfaceEventHandlerImpl (
    InterfacePtr iface,
    std::string interfaceID ) [inline]
```

**14.128.1.5 ~InterfaceEventHandlerImpl() [2/2]**

```
~InterfaceEventHandlerImpl ( ) [inline]
```

**14.128.2 Member Function Documentation****14.128.2.1 GetInterfaceId()**

```
std::string GetInterfaceId ( ) [inline]
```

**14.128.2.2 OnDeviceArrival() [1/2]**

```
void OnDeviceArrival (
    uint64_t serialNumber ) [inline], [virtual]
```

Device arrival event callback.

Implements [InterfaceEventHandler](#).

**14.128.2.3 OnDeviceArrival() [2/2]**

```
void OnDeviceArrival (
    uint64_t serialNumber ) [inline], [virtual]
```

Device arrival event callback.

Implements [InterfaceEventHandler](#).

**14.128.2.4 OnDeviceRemoval() [1/2]**

```
void OnDeviceRemoval (
    uint64_t serialNumber ) [inline], [virtual]
```

Callback to the device removal event.

**Parameters**

|                     |                                         |
|---------------------|-----------------------------------------|
| <i>serialNumber</i> | The serial number of the removed device |
|---------------------|-----------------------------------------|

Implements [InterfaceEventHandler](#).

**14.128.2.5 OnDeviceRemoval() [2/2]**

```
void OnDeviceRemoval (
    uint64_t serialNumber ) [inline], [virtual]
```

Callback to the device removal event.

**Parameters**

|                     |                                         |
|---------------------|-----------------------------------------|
| <i>serialNumber</i> | The serial number of the removed device |
|---------------------|-----------------------------------------|

Implements [InterfaceEventHandler](#).

#### 14.128.2.6 PrintGenericHandlerMessage()

```
void PrintGenericHandlerMessage (
    const unsigned long long deviceCount ) [inline]
```

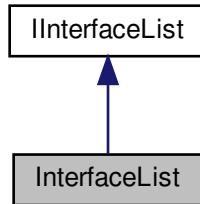
The documentation for this class was generated from the following files:

- src/AcquisitionMultipleCameraRecovery/[AcquisitionMultipleCameraRecovery.cpp](#)
- src/EnumerationEvents/[EnumerationEvents.cpp](#)

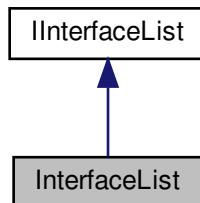
## 14.129 InterfaceList Class Reference

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".*
- `void Clear ()`  
*Clears the list of interfaces and destroys their corresponding objects.*

## Friends

- class `SystemImpl`

## Additional Inherited Members

### 14.129.1 Detailed Description

A list of the available interfaces on the system.

### 14.129.2 Constructor & Destructor Documentation

#### 14.129.2.1 `InterfaceList()` [1/2]

```
InterfaceList (
    void  )
```

#### 14.129.2.2 `~InterfaceList()`

```
virtual ~InterfaceList (
    void  ) [virtual]
```

### 14.129.2.3 InterfaceList() [2/2]

```
InterfaceList (
    const InterfaceList & iface )
```

## 14.129.3 Member Function Documentation

### 14.129.3.1 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](#).

### 14.129.3.2 GetByIndex()

```
InterfacePtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to an [Interface](#) object at the "index".

#### Parameters

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| <code>index</code> | The index at which to retrieve the <a href="#">Interface</a> object |
|--------------------|---------------------------------------------------------------------|

#### Returns

A pointer to an [Interface](#) object.

Implements [IInterfaceList](#).

### 14.129.3.3 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](#).

**14.129.3.4 operator=( )**

```
InterfaceList& operator= (
    const InterfaceList & iface )
```

Assignment operator.

**14.129.3.5 operator[]( )**

```
InterfacePtr operator[ ] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IInterfaceList](#).

**14.129.4 Friends And Related Function Documentation****14.129.4.1 SystemImpl**

```
friend class SystemImpl [friend]
```

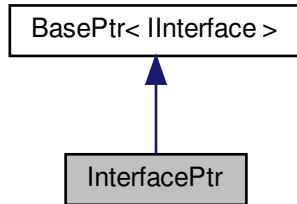
The documentation for this class was generated from the following file:

- [include/InterfaceList.h](#)

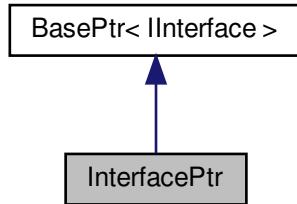
## 14.130 InterfacePtr Class Reference

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

#### 14.130.1 Detailed Description

A reference tracked pointer to the interface object.

## 14.130.2 Constructor & Destructor Documentation

### 14.130.2.1 InterfacePtr() [1/4]

```
InterfacePtr ( ) throw ( )    [inline]
```

Default Constructor.

### 14.130.2.2 InterfacePtr() [2/4]

```
InterfacePtr (
    const int   ) throw ( )    [inline]
```

Default Constructor with argument.

### 14.130.2.3 InterfacePtr() [3/4]

```
InterfacePtr (
    const long  ) throw ( )    [inline]
```

### 14.130.2.4 InterfacePtr() [4/4]

```
InterfacePtr (
    const std::nullptr_t  ) throw ( )    [inline]
```

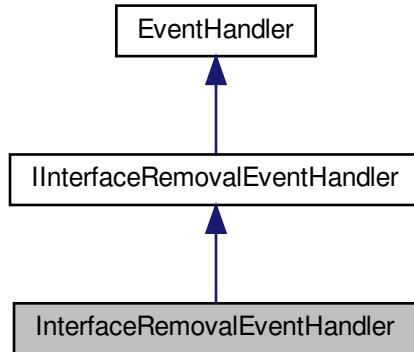
The documentation for this class was generated from the following file:

- [include/InterfacePtr.h](#)

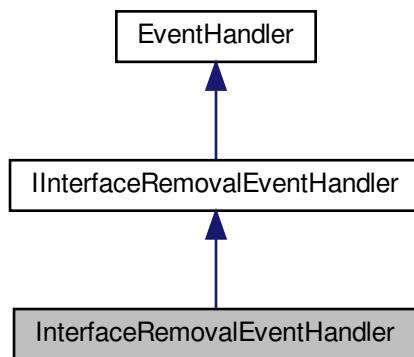
## 14.131 InterfaceRemovalEventHandler Class Reference

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 \(std::string interfaceID\)=0`](#)  
*Interface removal event callback.*

## Protected Member Functions

- `InterfaceRemovalEventHandler & operator= (const InterfaceRemovalEventHandler &)`  
*Assignment operator.*

## Additional Inherited Members

### 14.131.1 Detailed Description

An event handler for capturing the interface removal event.

Note that only GEV interface removals are currently handled.

### 14.131.2 Constructor & Destructor Documentation

#### 14.131.2.1 `InterfaceRemovalEventHandler()`

```
InterfaceRemovalEventHandler ( )
```

Default Constructor.

#### 14.131.2.2 `~InterfaceRemovalEventHandler()`

```
virtual ~InterfaceRemovalEventHandler ( ) [virtual]
```

Virtual Destructor.

### 14.131.3 Member Function Documentation

#### 14.131.3.1 `OnInterfaceRemoval()`

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

`Interface` removal event callback.

Note that only GEV interface removals are currently handled.

**Parameters**

|                    |                                 |
|--------------------|---------------------------------|
| <i>interfaceID</i> | The ID of the interface removed |
|--------------------|---------------------------------|

Implements [IInterfaceRemovalEventHandler](#).

**14.131.3.2 operator=()**

```
InterfaceRemovalEventHandler& operator= (
    const InterfaceRemovalEventHandler & ) [protected]
```

Assignment operator.

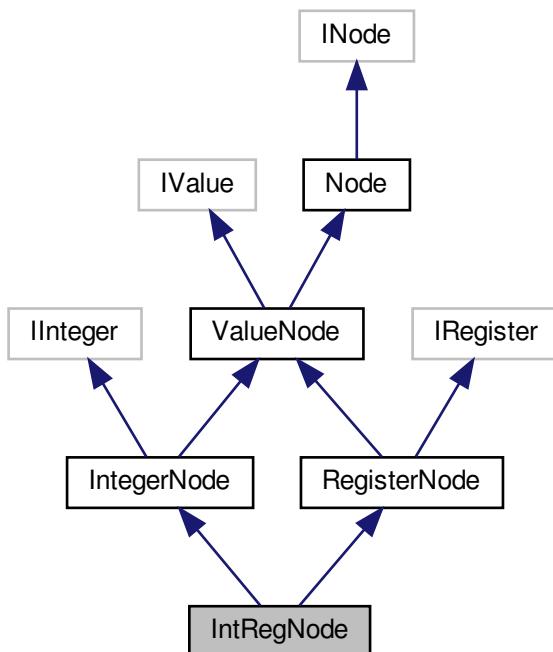
The documentation for this class was generated from the following file:

- [include/InterfaceRemovalEventHandler.h](#)

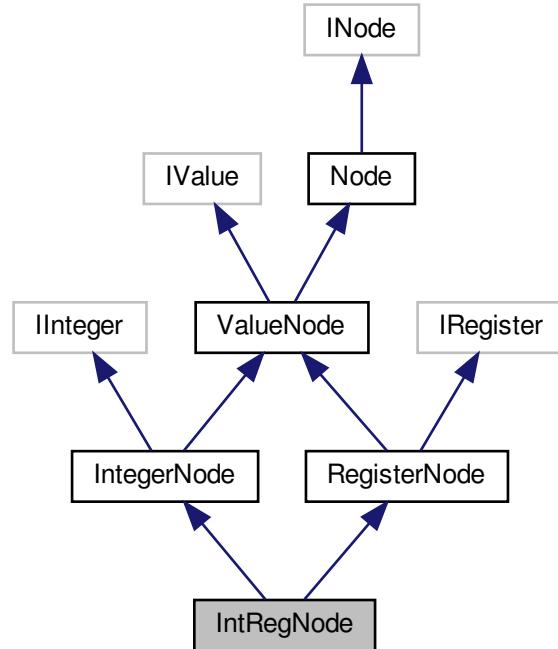
**14.132 IntRegNode Class Reference**

[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

### 14.132.1 Detailed Description

[Interface](#) for string properties.

### 14.132.2 Constructor & Destructor Documentation

**14.132.2.1 IntRegNode() [1/2]**

```
IntRegNode ( )
```

**14.132.2.2 IntRegNode() [2/2]**

```
IntRegNode ( std::shared_ptr< Node::NodeImpl > pInteger )
```

**14.132.2.3 ~IntRegNode()**

```
virtual ~IntRegNode ( ) [virtual]
```

**14.132.3 Member Function Documentation****14.132.3.1 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](#)

**14.133 IpInfo Struct Reference****Public Member Functions**

- [IpInfo \(\)](#)

**Public Attributes**

- std::string [ipAddress](#)
- std::string [subnetMask](#)
- std::string [gateway](#)
- unsigned int [subnetLength](#)

### 14.133.1 Constructor & Destructor Documentation

#### 14.133.1.1 `IpInfo()`

```
IpInfo () [inline]
```

### 14.133.2 Member Data Documentation

#### 14.133.2.1 `gateway`

```
std::string gateway
```

#### 14.133.2.2 `ipAddress`

```
std::string ipAddress
```

#### 14.133.2.3 `subnetLength`

```
unsigned int subnetLength
```

#### 14.133.2.4 `subnetMask`

```
std::string subnetMask
```

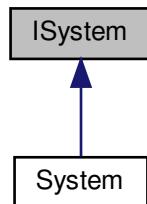
The documentation for this struct was generated from the following file:

- [include/AdapterConfig.h](#)

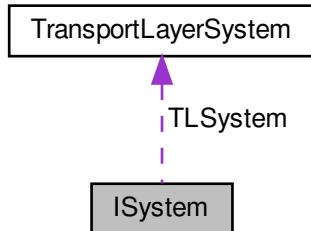
## 14.134 ISystem Class Reference

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 [`CameraList GetCameras\(bool updateInterfaces=true, bool updateCameras=true\)=0`](#)
- virtual bool [`UpdateCameras\(bool updateInterfaces=true\)=0`](#)
- virtual void [`UpdateInterfaceList\(\)=0`](#)
- virtual void [`RegisterEventHandler\(EventHandler &evtHandlerToRegister\)=0`](#)
- virtual void [`UnregisterEventHandler\(EventHandler &evtHandlerToUnregister\)=0`](#)
- virtual void [`RegisterInterfaceEventHandler\(EventHandler &evtHandlerToRegister, bool updateInterface=true\)=0`](#)
- virtual void [`UnregisterInterfaceEventHandler\(EventHandler &evtHandlerToUnregister\)=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`

### 14.134.1 Detailed Description

The interface file for `System`.

### 14.134.2 Constructor & Destructor Documentation

#### 14.134.2.1 ~ISystem()

```
virtual ~ISystem ( ) [inline], [virtual]
```

#### 14.134.2.2 ISystem() [1/2]

```
ISystem ( ) [inline], [protected]
```

**14.134.2.3 ISystem() [2/2]**

```
ISystem (
    const ISystem & ) [inline], [protected]
```

**14.134.3 Member Function Documentation****14.134.3.1 GetCameras()**

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.2 GetInterfaces()**

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.3 GetLibraryVersion()**

```
virtual const LibraryVersion GetLibraryVersion () [pure virtual]
```

Implemented in [System](#).

**14.134.3.4 GetLoggingEventPriorityLevel()**

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel () [pure virtual]
```

Implemented in [System](#).

**14.134.3.5 GetTLNodeMap()**

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [System](#).

**14.134.3.6 IsInUse()**

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.7 operator=( )**

```
ISystem& operator= (
    const ISystem & ) [protected]
```

**14.134.3.8 RegisterEventHandler()**

```
virtual void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.9 RegisterInterfaceEventHandler()**

```
virtual void RegisterInterfaceEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.10 RegisterLoggingEventHandler()**

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.11 ReleaseInstance()**

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.12 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](#).

**14.134.3.13 SetLoggingEventPriorityLevel()**

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.14 UnregisterAllLoggingEventHandlers()**

```
virtual void UnregisterAllLoggingEventHandlers ( ) [pure virtual]
```

Implemented in [System](#).

**14.134.3.15 UnregisterEventHandler()**

```
virtual void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [System](#).

#### 14.134.3.16 UnregisterInterfaceEventHandler()

```
virtual void UnregisterInterfaceEventHandler (
    EventHandler & evtHandlerToUnregister ) [pure virtual]
```

Implemented in [System](#).

#### 14.134.3.17 UnregisterLoggingEventHandler()

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [pure virtual]
```

Implemented in [System](#).

#### 14.134.3.18 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [pure virtual]
```

Implemented in [System](#).

#### 14.134.3.19 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [pure virtual]
```

Implemented in [System](#).

### 14.134.4 Friends And Related Function Documentation

#### 14.134.4.1 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

### 14.134.5 Member Data Documentation

#### 14.134.5.1 TLSysystem

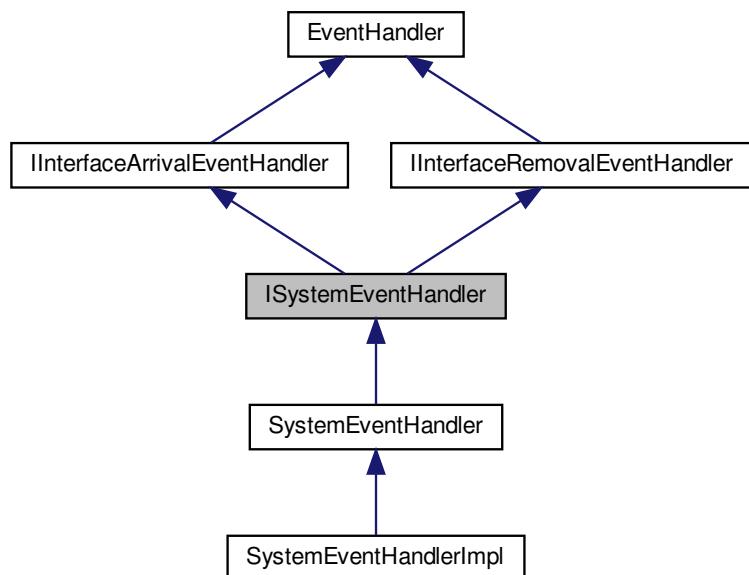
`TransportLayerSystem TLSysystem`

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

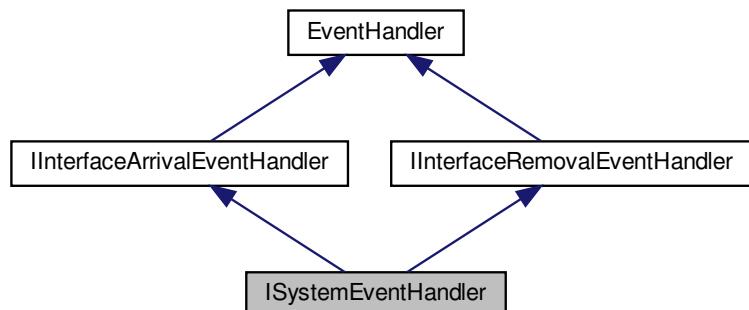
- `include/Interface/ISystem.h`

## 14.135 ISystemEventHandler Class Reference

Inheritance diagram for ISystemEventHandler:



Collaboration diagram for ISystemEventHandler:



## Public Member Functions

- virtual `~ISystemEventHandler ()`
- virtual void `OnInterfaceArrival (std::string interfaceID)=0`
- virtual void `OnInterfaceRemoval (std::string interfaceID)=0`

## Protected Member Functions

- `ISystemEventHandler ()`
- `ISystemEventHandler (const ISystemEventHandler &)`
- `ISystemEventHandler & operator= (const ISystemEventHandler &)`

## Additional Inherited Members

### 14.135.1 Constructor & Destructor Documentation

#### 14.135.1.1 `~ISystemEventHandler()`

```
virtual ~ISystemEventHandler ( ) [inline], [virtual]
```

#### 14.135.1.2 `ISystemEventHandler() [1/2]`

```
ISystemEventHandler ( ) [inline], [protected]
```

#### 14.135.1.3 `ISystemEventHandler() [2/2]`

```
ISystemEventHandler ( 
    const ISystemEventHandler & ) [inline], [protected]
```

### 14.135.2 Member Function Documentation

#### 14.135.2.1 `OnInterfaceArrival()`

```
virtual void OnInterfaceArrival ( 
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceArrivalEventHandler](#).

Implemented in [SystemEventHandler](#), and [SystemEventHandlerImpl](#).

### 14.135.2.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

Implements [IInterfaceRemovalEventHandler](#).

Implemented in [SystemEventHandler](#), and [SystemEventHandlerImpl](#).

### 14.135.2.3 operator=( )

```
ISystemEventHandler& operator=
    const ISystemEventHandler & ) [protected]
```

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

- include/Interface/[ISystemEventHandler.h](#)

## 14.136 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.*

### 14.136.1 Detailed Description

Options for saving JPEG image.

### 14.136.2 Constructor & Destructor Documentation

#### 14.136.2.1 **JPEGOption()**

```
JPEGOption ( ) [inline]
```

### 14.136.3 Member Data Documentation

#### 14.136.3.1 **progressive**

```
bool progressive
```

Whether to save as a progressive JPEG file.

#### 14.136.3.2 **quality**

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

#### 14.136.3.3 **reserved**

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 14.137 **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.*

### 14.137.1 Detailed Description

Options for saving JPEG2000 image.

### 14.137.2 Constructor & Destructor Documentation

#### 14.137.2.1 [JPG2Option\(\)](#)

[JPG2Option \(\)](#) [inline]

### 14.137.3 Member Data Documentation

#### 14.137.3.1 `quality`

`unsigned int quality`

JPEG saving quality in range (1-512).

#### 14.137.3.2 `reserved`

`unsigned int reserved[16]`

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 14.138 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.*

### 14.138.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

### 14.138.2 Member Data Documentation

#### 14.138.2.1 build

`unsigned int build`

Build number of the library.

#### 14.138.2.2 major

`unsigned int major`

Major version of the library.

#### 14.138.2.3 minor

`unsigned int minor`

Minor version of the library.

#### 14.138.2.4 type

```
unsigned int type
```

Version type of the library.

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

- include/SpinnakerDefs.h

## 14.139 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

### Public Member Functions

- [Lock](#) (const [LockableObject< Object >](#) &obj)
- [~Lock](#) ()

#### 14.139.1 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.

### 14.139.2 Constructor & Destructor Documentation

#### 14.139.2.1 Lock()

```
Lock (
    const LockableObject< Object > & obj ) [inline]
```

#### 14.139.2.2 ~Lock()

```
~Lock ( ) [inline]
```

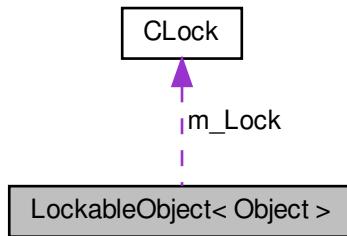
The documentation for this class was generated from the following file:

- include/SpinGenApi/[GCSynch.h](#)

## 14.140 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](#)

### 14.140.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

## 14.140.2 Member Function Documentation

### 14.140.2.1 GetLock()

```
Lock GetLock ( ) const [inline]
```

Get a new lock.

## 14.140.3 Friends And Related Function Documentation

### 14.140.3.1 Lock

```
friend class Lock [friend]
```

## 14.140.4 Member Data Documentation

### 14.140.4.1 m\_Lock

```
CLock m_Lock [mutable]
```

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

- include/SpinGenApi/GCSynch.h

## 14.141 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](#)

### 14.141.1 Detailed Description

The [LoggingEventData](#) object.

### 14.141.2 Constructor & Destructor Documentation

#### 14.141.2.1 ~LoggingEventData()

[~LoggingEventData \( \)](#)

Default Destructor.

### 14.141.2.2 LoggingEventData()

```
LoggingEventData (
    void * data ) [protected]
```

Default Constructor.

## 14.141.3 Member Function Documentation

### 14.141.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

#### Returns

The category name

### 14.141.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

#### Returns

The log message

### 14.141.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

#### Returns

The log event's NDC

#### 14.141.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

##### Returns

The log priority

#### 14.141.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

##### Returns

The priority name of the log

#### 14.141.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

##### Returns

The thread name

#### 14.141.3.7 GetTimestamp()

```
const char* GetTimestamp ( )
```

Gets the logging event time stamp.

##### Returns

The time stamp of the log

### 14.141.4 Friends And Related Function Documentation

#### 14.141.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

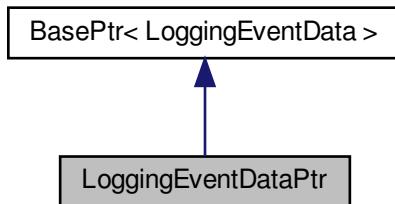
The documentation for this class was generated from the following file:

- [include/LoggingEventData.h](#)

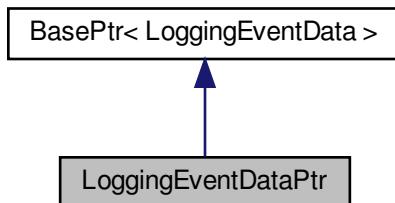
## 14.142 LoggingEventDataPtr Class Reference

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

### 14.142.1 Detailed Description

A reference tracked pointer to the LoggingEvent object.

### 14.142.2 Constructor & Destructor Documentation

#### 14.142.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw ( )    [inline]
```

Default Constructor.

#### 14.142.2.2 LoggingEventDataPtr() [2/4]

```
LoggingEventDataPtr (
    const int    ) throw ( )    [inline]
```

Default Constructor with argument.

#### 14.142.2.3 LoggingEventDataPtr() [3/4]

```
LoggingEventDataPtr (
    const long   ) throw ( )    [inline]
```

Default Constructor with argument.

#### 14.142.2.4 LoggingEventDataPtr() [4/4]

```
LoggingEventDataPtr (
    const std::nullptr_t ) throw ( )    [inline]
```

Default Constructor with argument.

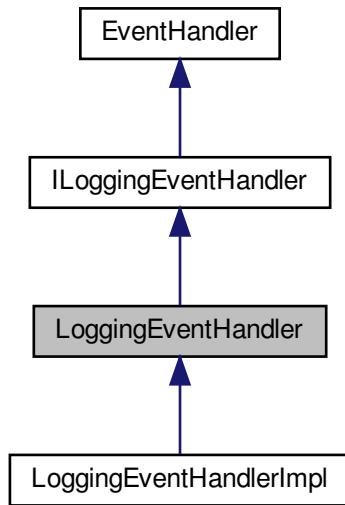
The documentation for this class was generated from the following file:

- [include/LoggingEventDataPtr.h](#)

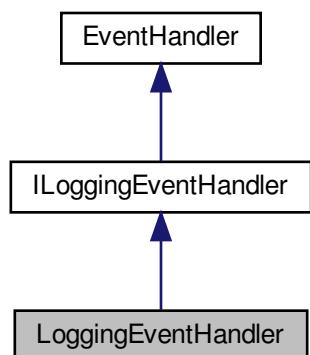
## 14.143 LoggingEventHandler Class Reference

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

### 14.143.1 Detailed Description

An event handler for capturing the device logging event.

### 14.143.2 Constructor & Destructor Documentation

#### 14.143.2.1 LoggingEventHandler()

`LoggingEventHandler ()`

Default constructor.

#### 14.143.2.2 ~LoggingEventHandler()

`~LoggingEventHandler ()`

Virtual destructor.

### 14.143.3 Member Function Documentation

#### 14.143.3.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

The callback for the log event.

**Parameters**

|                       |                           |
|-----------------------|---------------------------|
| <code>eventPtr</code> | The logging event pointer |
|-----------------------|---------------------------|

Implements [ILoggingEventHandler](#).

#### 14.143.3.2 `operator=()`

```
LoggingEventHandler& operator= (
    const LoggingEventHandler & ) [protected]
```

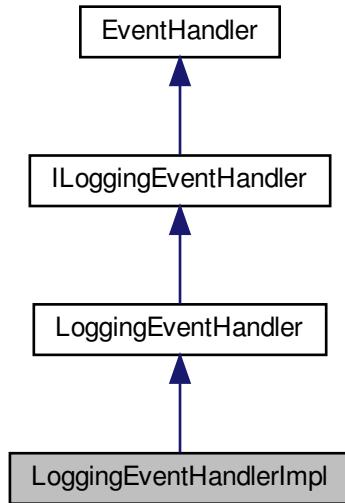
Assignment operator.

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

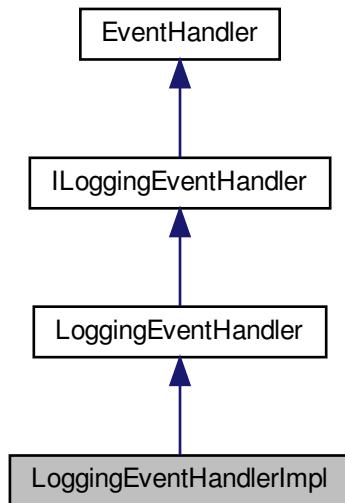
- [include/LoggingEventHandler.h](#)

## 14.144 LoggingEventHandlerImpl Class Reference

Inheritance diagram for LoggingEventHandlerImpl:



Collaboration diagram for LoggingEventHandlerImpl:



### Additional Inherited Members

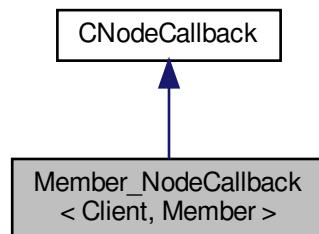
The documentation for this class was generated from the following file:

- src/Logging/[Logging.cpp](#)

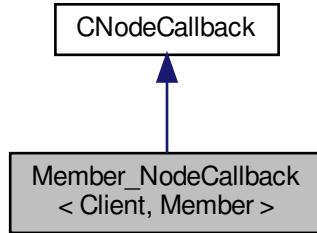
### 14.145 Member\_NodeCallback< Client, Member > Class Template Reference

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

### 14.145.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

### 14.145.2 Member Typedef Documentation

#### 14.145.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC (INode *)
```

Member function type.

### 14.145.3 Constructor & Destructor Documentation

#### 14.145.3.1 Member\_NodeCallback()

```
Member_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType ) [inline]
```

Constructor.

### 14.145.4 Member Function Documentation

#### 14.145.4.1 Destroy()

```
virtual void Destroy () [inline], [virtual]
```

destroys the object

Implements [CNodeCallback](#).

#### 14.145.4.2 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](#)

## 14.146 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 [reserved](#) [256]

### 14.146.1 Detailed Description

Options for saving MJPG files.

### 14.146.2 Constructor & Destructor Documentation

#### 14.146.2.1 MJPGOption()

[MJPGOption \(\)](#) [inline]

### 14.146.3 Member Data Documentation

#### 14.146.3.1 frameRate

float [frameRate](#)

Frame rate of the stream.

#### 14.146.3.2 quality

unsigned int [quality](#)

*Image quality (1-100)*

### 14.146.3.3 reserved

```
unsigned int reserved[256]
```

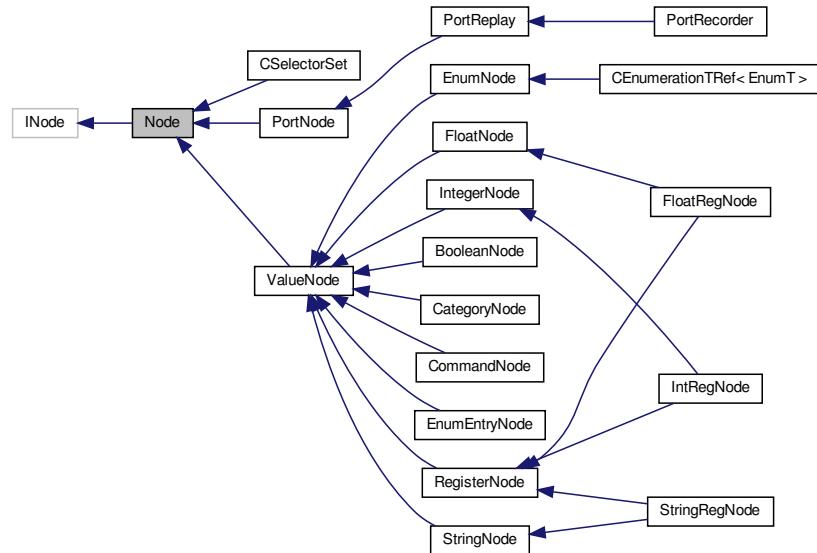
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

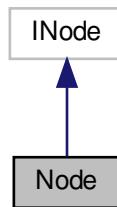
## 14.147 Node Class Reference

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 IsCachable () 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 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 [GenICam::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 [lingroup 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](#)

### 14.147.1 Detailed Description

class common to all nodes

### 14.147.2 Constructor & Destructor Documentation

#### 14.147.2.1 Node() [1/2]

```
Node ()
```

Constructor.

#### 14.147.2.2 Node() [2/2]

```
Node (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Constructor.

#### 14.147.2.3 ~Node()

```
~Node ()
```

Destructor.

### 14.147.3 Member Function Documentation

#### 14.147.3.1 DeregisterCallback()

```
virtual bool DeregisterCallback (
    CallbackHandleType hCallback ) [virtual]
```

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

##### Returns

true if the callback handle was valid

#### 14.147.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#), and [PortReplay](#).

#### 14.147.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const [virtual]
```

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

#### 14.147.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const [virtual]
```

Get Caching Mode.

#### 14.147.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const [virtual]
```

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

#### 14.147.3.6 GetChildren()

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

Get all nodes this node directly depends on.

##### Parameters

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

#### 14.147.3.7 GetDescription()

```
virtual GenICam::gcstring GetDescription() const [virtual]
```

Get a long description of the node.

#### 14.147.3.8 GetDeviceName()

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

Get a name of the device.

#### 14.147.3.9 GetDisplayName()

```
virtual GenICam::gcstring GetDisplayName() const [virtual]
```

Get a name string for display.

#### 14.147.3.10 GetDocuURL()

```
virtual GenICam::gcstring GetDocuURL() const [virtual]
```

Gets a URL pointing to the documentation of that feature.

#### 14.147.3.11 GetEventID()

```
virtual GenICam::gcstring GetEventID() const [virtual]
```

Get the EventId of the node.

#### 14.147.3.12 GetName()

```
virtual GenICam::gcstring GetName(
    bool FullQualified = false) const [virtual]
```

Get node name.

#### 14.147.3.13 GetNameSpace()

```
virtual GenApi::ENameSpace GetNameSpace ( ) const [virtual]
```

Get name space.

#### 14.147.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const
```

Get [Node](#) handle.

#### 14.147.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap ( ) const [virtual]
```

Retrieves the central node map.

#### 14.147.3.16 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 |
|-----|----------------|----------------------|

#### 14.147.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const [virtual]
```

recommended polling time (for not cacheable nodes)

**14.147.3.18 GetPrincipalInterfaceType()**

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

**14.147.3.19 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.

**14.147.3.20 GetPropertyNames()**

```
virtual void GetPropertyNames (
    GenICam::gcstring_vector & PropertyNames ) const [virtual]
```

Returns a list of the names all properties set during initialization.

**14.147.3.21 GetSelectedFeatures()**

```
virtual void GetSelectedFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of selected features

**14.147.3.22 GetSelectingFeatures()**

```
virtual void GetSelectingFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of features selecting this node

**14.147.3.23 GetToolTip()**

```
virtual GenICam::gcstring GetToolTip( ) const [virtual]
```

Get a short description of the node.

**14.147.3.24 GetVisibility()**

```
virtual EVisibility GetVisibility( ) const [virtual]
```

Get the recommended visibility of the node.

**14.147.3.25 ImposeAccessMode()**

```
virtual void ImposeAccessMode( EAccessMode ImposedAccessMode ) [virtual]
```

Imposes an access mode to the natural access mode of the node.

**14.147.3.26 ImposeVisibility()**

```
virtual void ImposeVisibility( EVisibility ImposedVisibility ) [virtual]
```

Imposes a visibility to the natural visibility of the node.

**14.147.3.27 InvalidateNode()**

```
virtual void InvalidateNode( ) [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

**14.147.3.28 IsAccessModeCacheable()**

```
virtual EYesNo IsAccessModeCacheable( ) const [virtual]
```

True if the AccessMode can be cached.

**14.147.3.29 IsCachable()**

```
virtual bool IsCachable () const [virtual]
```

Is the node value cacheable.

**14.147.3.30 IsDeprecated()**

```
virtual bool IsDeprecated () const [virtual]
```

True if the node should not be used any more.

**14.147.3.31 IsFeature()**

```
virtual bool IsFeature () const [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

**14.147.3.32 IsSelector()**

```
virtual bool IsSelector () const [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

**14.147.3.33 IsStreamable()**

```
virtual bool IsStreamable () const [virtual]
```

True if the node is streamable.

**14.147.3.34 operator"!=()**

```
virtual bool operator!= (
    int nullPtr ) const [virtual]
```

#### 14.147.3.35 operator==( )

```
virtual bool operator== (
    int nullPtr ) const [virtual]
```

#### 14.147.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
    CNodeCallback * pCallback ) [virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

#### 14.147.3.37 SetNodeHandle()

```
void SetNodeHandle (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set [Node](#) handle.

#### 14.147.3.38 SetNodeMap()

```
void SetNodeMap (
    INodeMap * pNodeMap )
```

#### 14.147.3.39 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](#).

#### 14.147.3.40 SetReference() [2/2]

```
virtual void SetReference (
    ISelector * pBase ) [virtual]
```

#### 14.147.4 Member Data Documentation

##### 14.147.4.1 m\_Callbacks

```
std::list<CallbackHandleType_t*> m_Callbacks [protected]
```

List of callbacks.

##### 14.147.4.2 m\_pNodeData

```
std::shared_ptr<Node::NodeImpl> m_pNodeData [protected]
```

##### 14.147.4.3 m\_pNodeMap

```
INodeMap* m_pNodeMap [protected]
```

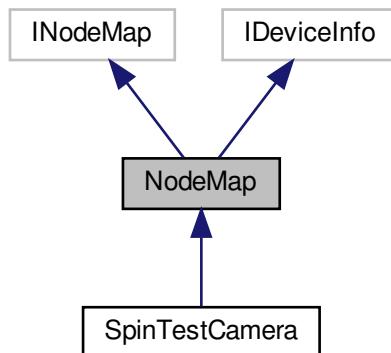
The documentation for this class was generated from the following file:

- include/SpinGenApi/[Node.h](#)

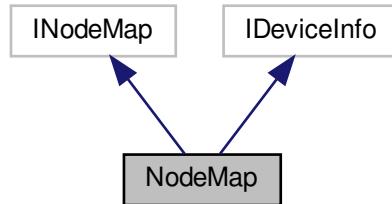
## 14.148 NodeMap Class Reference

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](#).*

### 14.148.1 Detailed Description

Smart pointer template for NodeMaps with create function.

#### Parameters

|                               |                                                                           |
|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">TCameraParams</a> | The camera specific parameter class (auto generated from camera xml file) |
|-------------------------------|---------------------------------------------------------------------------|

### 14.148.2 Constructor & Destructor Documentation

#### 14.148.2.1 NodeMap()

```
NodeMap (
    GenICam::gcstring DeviceName = "Device" )
```

Constructor.

#### 14.148.2.2 ~NodeMap()

```
virtual ~NodeMap ( ) [virtual]
```

Destructor.

### 14.148.3 Member Function Documentation

#### 14.148.3.1 ClearXMLCache()

```
static bool ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

#### 14.148.3.2 Connect() [1/2]

```
virtual bool Connect (
    IPort * pPort ) const [virtual]
```

Connects a port to the standard port "Device".

#### 14.148.3.3 Connect() [2/2]

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

Connects a port to a port node with given name.

**14.148.3.4 Destroy()**

```
void Destroy ( )
```

Destroys the node map.

**14.148.3.5 GetDeviceName()**

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

Get device name.

**14.148.3.6 GetDeviceVersion()**

```
virtual void GetDeviceVersion ( GenICam::Version\_t & Version ) [virtual]
```

Get the version of the device description file.

**14.148.3.7 GetGenApiVersion()**

```
virtual void GetGenApiVersion ( GenICam::Version\_t & Version, uint16_t & Build ) [virtual]
```

Get the version of the DLL's [GenApi](#) implementation.

**14.148.3.8 GetLock()**

```
virtual CLock& GetLock ( ) const [virtual]
```

Returns the lock which guards the node map.

**14.148.3.9 GetModelName()**

```
virtual GenICam::gcstring GetModelName ( ) [virtual]
```

Get the model name.

**14.148.3.10 GetNode()**

```
virtual INode* GetNode (
    const GenICam::gcstring & key ) const [virtual]
```

Retrieves the node from the central map by name.

**14.148.3.11 GetNodeMapHandle()**

```
void* GetNodeMapHandle ( ) const
```

**14.148.3.12 GetNodes()**

```
virtual void GetNodes (
    NodeList_t & Nodes ) const [virtual]
```

Retrieves all nodes in the node map.

**14.148.3.13 GetNumNodes()**

```
virtual uint64_t GetNumNodes ( ) const [virtual]
```

Get the number of nodes in the map.

**14.148.3.14 GetProductGuid()**

```
virtual GenICam::gcstring GetProductGuid ( ) [virtual]
```

Get the GUID describing the product.

**14.148.3.15 GetSchemaVersion()**

```
virtual void GetSchemaVersion (
    GenICam::Version_t & Version ) [virtual]
```

Get the schema version number.

**14.148.3.16 GetStandardNameSpace()**

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

**14.148.3.17 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}" were {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

**14.148.3.18 GetToolTip()**

```
virtual GenICam::gcstring GetToolTip ( ) [virtual]
```

Get tool tip.

**14.148.3.19 GetVendorName()**

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

**14.148.3.20 GetVersionGuid()**

```
virtual GenICam::gcstring GetVersionGuid ( ) [virtual]
```

Get the GUID describing the product version.

**14.148.3.21 InvalidateNodes()**

```
virtual void InvalidateNodes ( ) const [virtual]
```

Invalidates all nodes.

**14.148.3.22 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*);

**14.148.3.23 LoadXMLFromFileInject()**

```
void LoadXMLFromFileInject (   
    GenICam::gcstring TargetFileName,  
    GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

**14.148.3.24 LoadXMLFromString()**

```
void LoadXMLFromString (   
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

**14.148.3.25 LoadXMLFromStringInject()**

```
void LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLData const,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

**14.148.3.26 LoadXMLFromZIPData()**

```
void LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

**14.148.3.27 LoadXMLFromZIPFile()**

```
void LoadXMLFromZIPFile (
    GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

**14.148.3.28 Poll()**

```
virtual void Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

**14.148.4 Member Data Documentation****14.148.4.1 \_Ptr**

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

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

- include/SpinGenApi/[NodeMap.h](#)

## 14.149 CNodeMapFactory::NodeStatistics\_t Struct Reference

### Public Attributes

- `uint32_t NumNodes`
- `uint32_t NumProperties`
- `uint32_t NumLinks`
- `uint32_t NumStrings`

#### 14.149.1 Member Data Documentation

##### 14.149.1.1 NumLinks

```
uint32_t NumLinks
```

##### 14.149.1.2 NumNodes

```
uint32_t NumNodes
```

##### 14.149.1.3 NumProperties

```
uint32_t NumProperties
```

##### 14.149.1.4 NumStrings

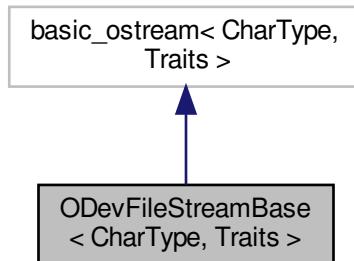
```
uint32_t NumStrings
```

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

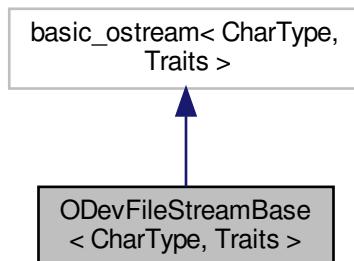
- `include/SpinGenApi/NodeMapFactory.h`

## 14.150 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`  
*Open file on device in write mode.*
- `bool is_open () const`
- `void open (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out|std::ios_base::trunc)`  
*Close the file on device.*
- `void close ()`

### 14.150.1 Member Typedef Documentation

#### 14.150.1.1 filebuf\_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

#### 14.150.1.2 ios\_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

#### 14.150.1.3 ostream\_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

### 14.150.2 Member Function Documentation

#### 14.150.2.1 close()

```
void close () [inline]
```

Close the file on device.

#### 14.150.2.2 is\_open()

```
bool is_open () const [inline]
```

#### 14.150.2.3 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> | <a href="#">NodeMap</a> of the device to which the <a href="#">FileProtocolAdapter</a> is attached |
| <i>pFileName</i>  | Name of the file to open                                                                           |
| <i>mode</i>       | open mode                                                                                          |

## 14.150.2.4 rdbuf()

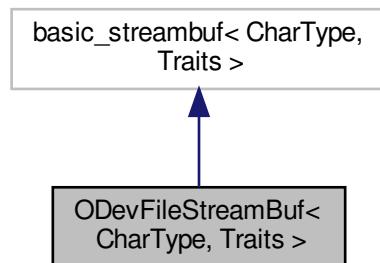
```
filebuf_type* rdbuf ( ) const [inline]
```

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

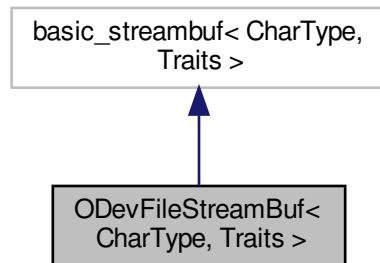
- include/SpinGenApi/[Filestream.h](#)

## 14.151 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 xputn (const char_type *s, std::streamsize n)`
- `int_type overflow (int_type c=traits_type::eof())`
- `int sync ()`

### 14.151.1 Constructor & Destructor Documentation

#### 14.151.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

#### 14.151.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

### 14.151.2 Member Function Documentation

#### 14.151.2.1 close()

```
filebuf_type* close ( ) [inline]
```

#### 14.151.2.2 is\_open()

```
bool is_open ( ) const [inline]
```

### 14.151.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode ) [inline]
```

### 14.151.2.4 overflow()

```
int_type overflow (
    int_type c = traits_type::eof() ) [inline], [protected]
```

### 14.151.2.5 sync()

```
int sync () [inline], [protected]
```

### 14.151.2.6 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

## 14.152 PGMOOption Struct Reference

Options for saving PGM images.

### Public Member Functions

- [PGMOOption \(\)](#)

### Public Attributes

- bool [binaryFile](#)  
*Whether to save the PPM as a binary file.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 14.152.1 Detailed Description

Options for saving PGM images.

### 14.152.2 Constructor & Destructor Documentation

#### 14.152.2.1 PGMOption()

[PGMOption \(\)](#) [inline]

### 14.152.3 Member Data Documentation

#### 14.152.3.1 binaryFile

bool binaryFile

Whether to save the PPM as a binary file.

#### 14.152.3.2 reserved

unsigned int reserved[16]

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 14.153 PNGOption Struct Reference

Options for saving PNG images.

### 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.*

### 14.153.1 Detailed Description

Options for saving PNG images.

### 14.153.2 Constructor & Destructor Documentation

#### 14.153.2.1 [PNGOption\(\)](#)

```
PNGOption () [inline]
```

### 14.153.3 Member Data Documentation

#### 14.153.3.1 [compressionLevel](#)

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

#### 14.153.3.2 [interlaced](#)

```
bool interlaced
```

Whether to save the PNG as interlaced.

### 14.153.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

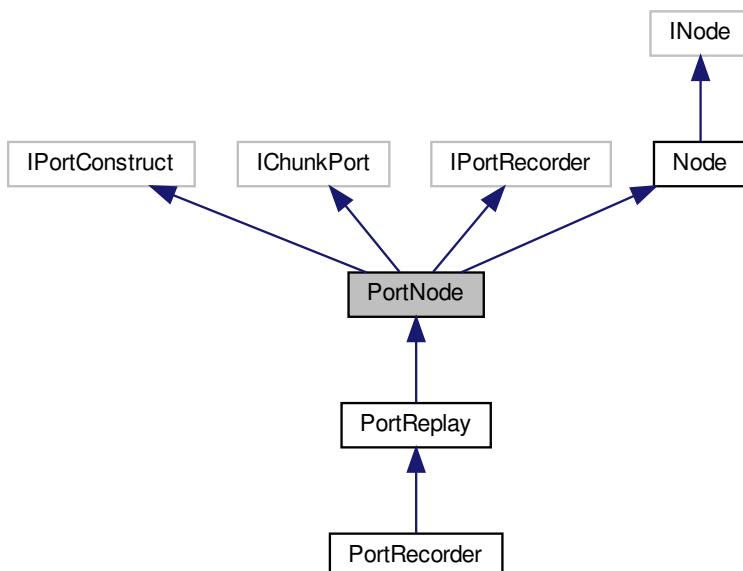
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

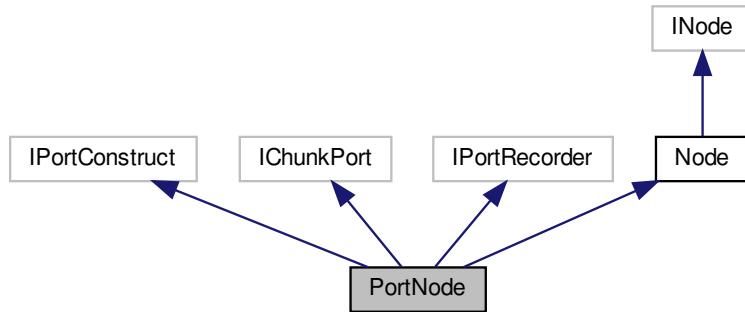
## 14.154 PortNode Class Reference

[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 GetSwapEndianess ()`  
*Determines if the port adapter must perform an endianess 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

### 14.154.1 Detailed Description

[Interface](#) for value properties.

### 14.154.2 Constructor & Destructor Documentation

#### 14.154.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

#### 14.154.2.2 PortNode() [2/2]

```
PortNode ( std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam IValue](#)

#### 14.154.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

### 14.154.3 Member Function Documentation

#### 14.154.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const [virtual]
```

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

#### 14.154.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const [virtual]
```

Get the Id of the chunk the port should be attached to.

#### 14.154.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle ( ) [inline]
```

#### 14.154.3.4 GetSwapEndianess()

```
virtual EYesNo GetSwapEndianess ( ) [virtual]
```

Determines if the port adapter must perform an endianess swap.

#### 14.154.3.5 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](#).

#### 14.154.3.6 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](#).

**14.154.3.7 SetPortImpl()**

```
void SetPortImpl (
    IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

**14.154.3.8 SetReference() [1/5]**

```
virtual void SetReference (
    IChunkPort * pBase ) [virtual]
```

overload SetReference for Value

**14.154.3.9 SetReference() [2/5]**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [Node](#).

**14.154.3.10 SetReference() [3/5]**

```
virtual void SetReference
```

Reference interface overrides \ingroup Spinnaker\_GenApi\_PublicImpl.

**14.154.3.11 SetReference() [4/5]**

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

**14.154.3.12 SetReference() [5/5]**

```
virtual void SetReference
```

**14.154.3.13 StartRecording()**

```
virtual void StartRecording (   
    IPortWriteList * pPortRecorder ) [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

**14.154.3.14 StopRecording()**

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in [PortRecorder](#).

**14.154.3.15 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](#).

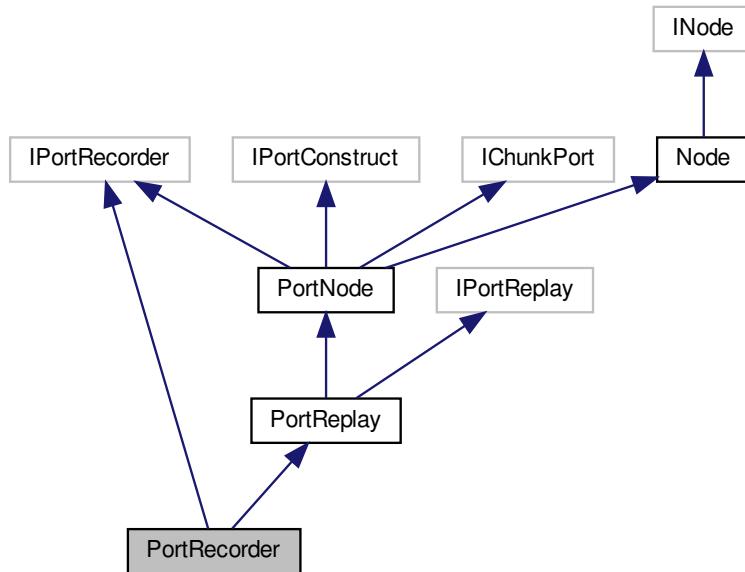
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortNode.h](#)

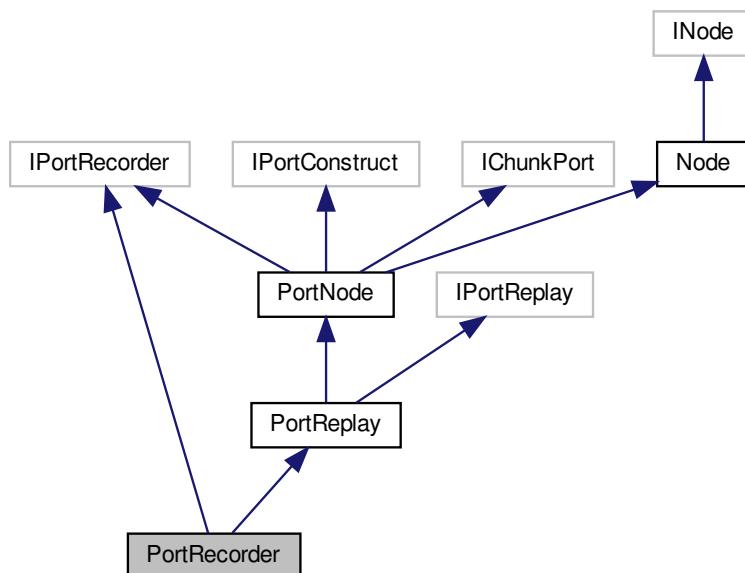
## 14.155 PortRecorder Class Reference

[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

### 14.155.1 Detailed Description

[Interface](#) for recording write commands on a port.

### 14.155.2 Constructor & Destructor Documentation

#### 14.155.2.1 PortRecorder()

`PortRecorder ()`

#### 14.155.2.2 ~PortRecorder()

`virtual ~PortRecorder () [virtual]`

### 14.155.3 Member Function Documentation

#### 14.155.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

Reimplemented from [PortReplay](#).

#### 14.155.3.2 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](#).

#### 14.155.3.3 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](#).

#### 14.155.3.4 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortReplay](#).

#### 14.155.3.5 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

#### 14.155.3.6 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

stops recording

Reimplemented from [PortNode](#).

#### 14.155.3.7 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](#).

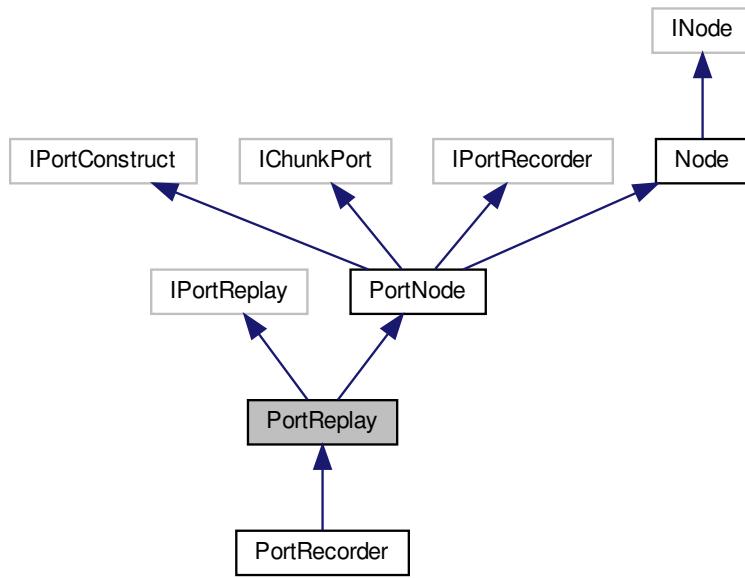
The documentation for this class was generated from the following file:

- include/SpinGenApi/[PortRecorder.h](#)

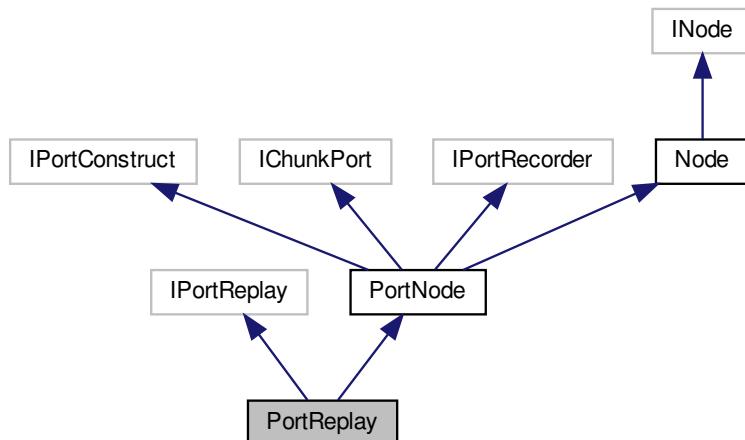
## 14.156 PortReplay Class Reference

[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

### 14.156.1 Detailed Description

[Interface](#) for replaying write commands on a port.

### 14.156.2 Constructor & Destructor Documentation

#### 14.156.2.1 PortReplay()

[PortReplay \(\)](#)

#### 14.156.2.2 ~PortReplay()

[virtual ~PortReplay \(\) \[virtual\]](#)

### 14.156.3 Member Function Documentation

#### 14.156.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented from [Node](#).

Reimplemented in [PortRecorder](#).

#### 14.156.3.2 GetPortReplayHandle()

```
void* GetPortReplayHandle ( )
```

#### 14.156.3.3 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](#).

#### 14.156.3.4 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](#).

#### 14.156.3.5 SetReference()

```
virtual void SetReference (
    IPort * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

#### 14.156.3.6 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](#)

## 14.157 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.*

#### 14.157.1 Detailed Description

Options for saving PPM images.

## 14.157.2 Constructor & Destructor Documentation

### 14.157.2.1 PPMOption()

```
PPMOption ( ) [inline]
```

## 14.157.3 Member Data Documentation

### 14.157.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

### 14.157.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

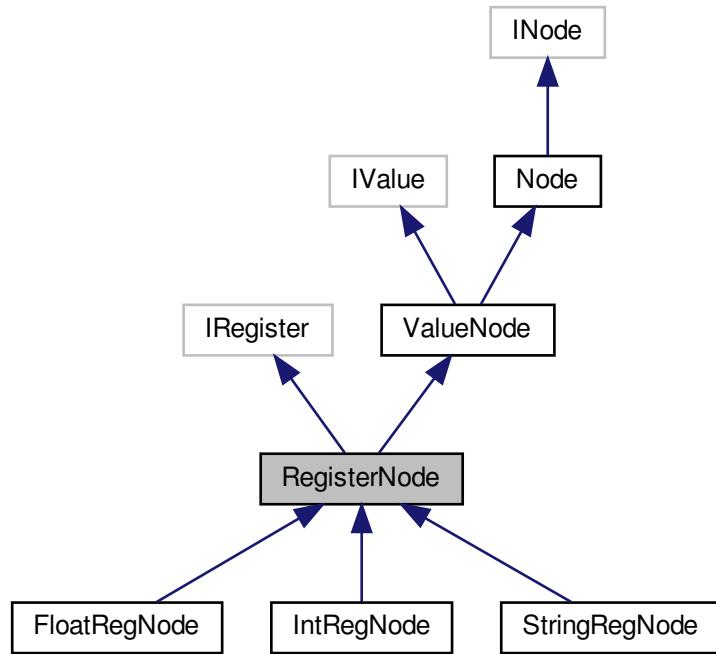
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

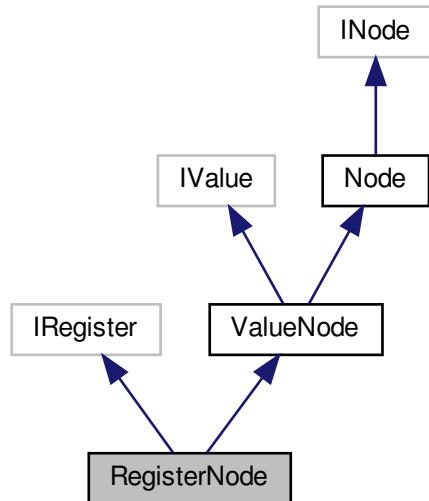
## 14.158 RegisterNode Class Reference

[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

### 14.158.1 Detailed Description

[Interface](#) for string properties.

### 14.158.2 Constructor & Destructor Documentation

#### 14.158.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

#### 14.158.2.2 RegisterNode() [2/2]

```
RegisterNode ( std::shared_ptr< Node::NodeImpl > pRegister )
```

#### 14.158.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

### 14.158.3 Member Function Documentation

#### 14.158.3.1 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

**14.158.3.2 GetAddress()**

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

**14.158.3.3 GetLength()**

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

**14.158.3.4 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) |

### 14.158.3.5 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

## 14.159 SingleChunkData\_t Struct Reference

### Public Attributes

- uint64\_t [ChunkID](#)
- ptrdiff\_t [ChunkOffset](#)
- size\_t [ChunkLength](#)

### 14.159.1 Member Data Documentation

#### 14.159.1.1 ChunkID

```
uint64_t ChunkID
```

#### 14.159.1.2 ChunkLength

```
size_t ChunkLength
```

#### 14.159.1.3 ChunkOffset

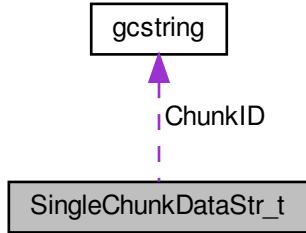
```
ptrdiff_t ChunkOffset
```

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

- include/SpinGenApi/ChunkAdapterGeneric.h

## 14.160 SingleChunkDataStr\_t Struct Reference

Collaboration diagram for SingleChunkDataStr\_t:



### Public Attributes

- `GenICam::gcstring ChunkID`
- `ptrdiff_t ChunkOffset`
- `size_t ChunkLength`

#### 14.160.1 Member Data Documentation

##### 14.160.1.1 ChunkID

`GenICam::gcstring ChunkID`

##### 14.160.1.2 ChunkLength

`size_t ChunkLength`

##### 14.160.1.3 ChunkOffset

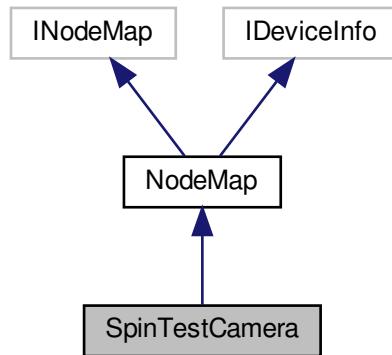
`ptrdiff_t ChunkOffset`

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

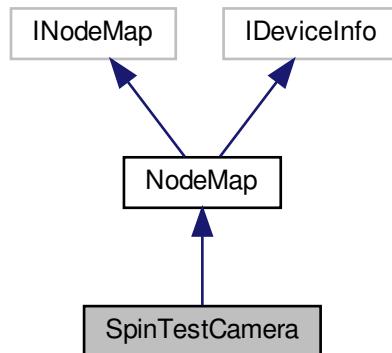
- `include/SpinGenApi/ChunkAdapterGeneric.h`

## 14.161 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](#)

## 14.162 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.*

### 14.162.1 Detailed Description

Provides the functionality for the user to record images to an AVI/MP4 file.

### 14.162.2 Constructor & Destructor Documentation

#### 14.162.2.1 SpinVideo()

`SpinVideo ()`

Default constructor.

#### 14.162.2.2 ~SpinVideo()

`virtual ~SpinVideo () [virtual]`

Default destructor.

### 14.162.3 Member Function Documentation

#### 14.162.3.1 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. |
|---------------|----------------------|

**14.162.3.2 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**

|                                         |
|-----------------------------------------|
| <a href="#">Open()</a>                  |
| <a href="#">Append(ImagePtr pImage)</a> |

**14.162.3.3 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**

|                                         |
|-----------------------------------------|
| <a href="#">Close()</a>                 |
| <a href="#">Append(ImagePtr pImage)</a> |

**14.162.3.4 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](#)**14.162.3.5 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](#)**14.162.3.6 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.

**Parameters**

|             |                                    |
|-------------|------------------------------------|
| <i>size</i> | The maximum video file size in MB. |
|-------------|------------------------------------|

See also

[Append\(ImagePtr plImage\)](#)

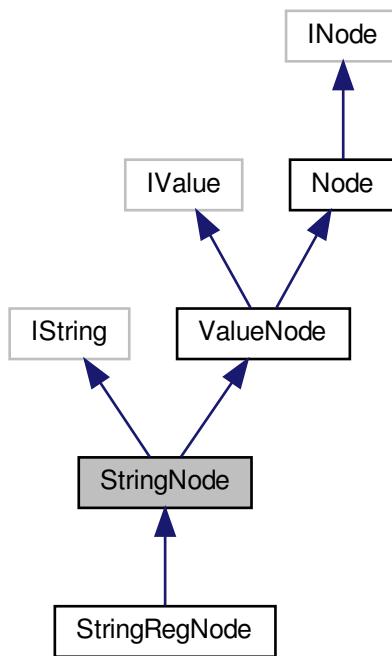
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

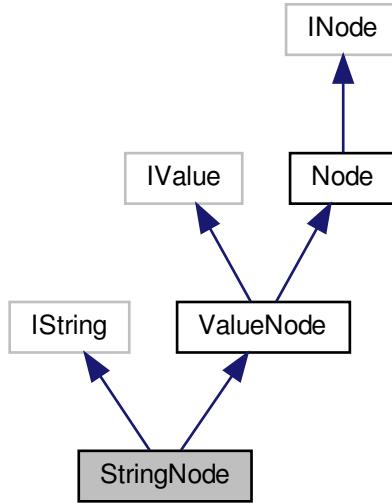
## 14.163 StringNode Class Reference

[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

### 14.163.1 Detailed Description

[Interface](#) for string properties.

## 14.163.2 Constructor & Destructor Documentation

### 14.163.2.1 `StringNode()` [1/2]

```
StringNode ( )
```

### 14.163.2.2 `StringNode()` [2/2]

```
StringNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

### 14.163.2.3 `~StringNode()`

```
virtual ~StringNode ( ) [virtual]
```

## 14.163.3 Member Function Documentation

### 14.163.3.1 `GetMaxLength()`

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

### 14.163.3.2 `GetValue()`

```
virtual GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

#### Parameters

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

**Returns**

The value read

**14.163.3.3 operator()**

```
virtual GenICam::gcstring operator() () [virtual]
```

Get node value.

**14.163.3.4 operator\*()**

```
virtual GenICam::gcstring operator* () [virtual]
```

Get node value.

**14.163.3.5 operator=( )**

```
virtual IString& operator= (
    const GenICam::gcstring & Value ) [virtual]
```

Set node value.

**14.163.3.6 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

**14.163.3.7 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) |

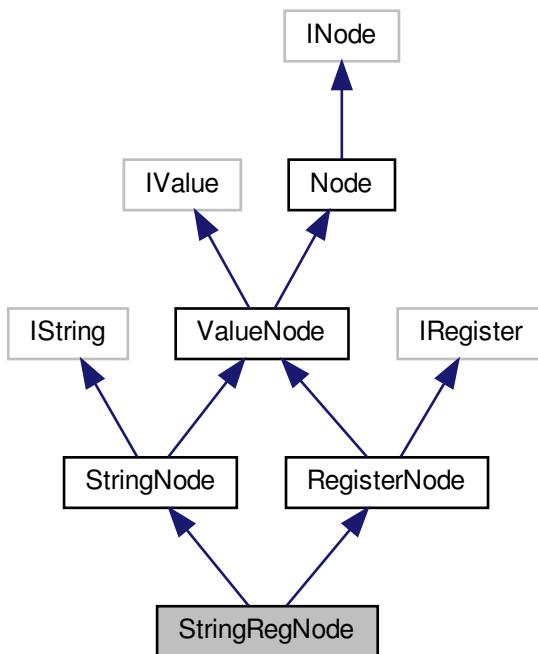
The documentation for this class was generated from the following file:

- include/SpinGenApi/StringNode.h

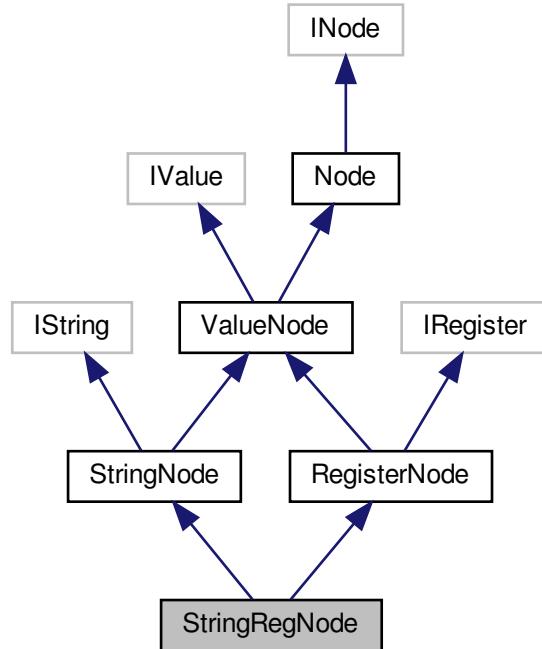
## 14.164 StringRegNode Class Reference

[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

### 14.164.1 Detailed Description

[Interface](#) for string properties.

### 14.164.2 Constructor & Destructor Documentation

**14.164.2.1 StringRegNode() [1/2]**

```
StringRegNode ( )
```

**14.164.2.2 StringRegNode() [2/2]**

```
StringRegNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

**14.164.2.3 ~StringRegNode()**

```
virtual ~StringRegNode ( ) [virtual]
```

**14.164.3 Member Function Documentation****14.164.3.1 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [RegisterNode](#).

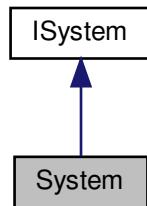
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringRegNode.h](#)

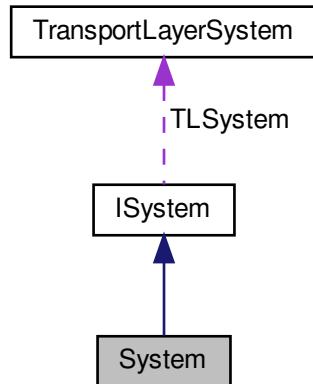
## 14.165 System Class Reference

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 `ReleaselInstance ()`  
*This call releases the instance of the `System` Singleton for this process.*
- virtual `InterfaceList GetInterfaces (bool updateInterface=true)`  
*Returns a list of interfaces available 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.*
- virtual void [UpdateInterfaceList](#) ()  
*Updates the list of interfaces on the system.*
- void [RegisterEventHandler](#) (EventHandler &evtHandlerToRegister)  
*Registers an event handler for the system.*
- void [UnregisterEventHandler](#) (EventHandler &evtHandlerToUnregister)  
*Unregisters an event handler for the system.*
- virtual void [RegisterInterfaceEventHandler](#) (EventHandler &evtHandlerToRegister, bool updateInterface=true)  
*Registers event handlers for all available interfaces that are found on the system If new interfaces are detected by the system after [RegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.*
- void [UnregisterInterfaceEventHandler](#) (EventHandler &evtHandlerToUnregister)  
*Unregisters event handlers for all available interfaces that are found on 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

### 14.165.1 Detailed Description

The system object is used to retrieve the list of interfaces and cameras available.

## 14.165.2 Constructor & Destructor Documentation

### 14.165.2.1 ~System()

```
virtual ~System ( ) [virtual]
```

Default destructor.

### 14.165.2.2 System()

```
System ( ) [protected]
```

Default constructor.

## 14.165.3 Member Function Documentation

### 14.165.3.1 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 <a href="#">updateInterfaceList()</a> is called before getting cameras from available interfaces on the system |
| <i>updateCameras</i>    | Determines whether or not <a href="#">UpdateCameras()</a> is called before getting cameras from available interfaces on the system       |

**Returns**

An [CameraList](#) object that contains a list of all cameras.

Implements [ISystem](#).

**14.165.3.2 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.

**14.165.3.3 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**

|                              |                                                                                                               |
|------------------------------|---------------------------------------------------------------------------------------------------------------|
| <code>updateInterface</code> | Determines whether or not <a href="#">UpdateInterfaceList()</a> is called before getting available interfaces |
|------------------------------|---------------------------------------------------------------------------------------------------------------|

**Returns**

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

#### 14.165.3.4 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](#).

#### 14.165.3.5 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel () [virtual]
```

Retrieves the current logging event priority level.

[Spinnaker](#) uses five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- 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](#).

#### 14.165.3.6 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](#).

#### 14.165.3.7 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](#).

#### 14.165.3.8 RegisterEventHandler()

```
void RegisterEventHandler (
    EventHandler & evtHandlerToRegister ) [virtual]
```

Registers an event handler for the system.

##### Parameters

|                                   |                                              |
|-----------------------------------|----------------------------------------------|
| <code>evtHandlerToRegister</code> | The event handler to register for the system |
|-----------------------------------|----------------------------------------------|

Implements [ISystem](#).

#### 14.165.3.9 RegisterInterfaceEventHandler()

```
virtual void RegisterInterfaceEventHandler (
    EventHandler & evtHandlerToRegister,
    bool updateInterface = true ) [virtual]
```

Registers event handlers for all available interfaces that are found on the system If new interfaces are detected by the system after [RegisterInterfaceEventHandler\(\)](#) is called, those interfaces will be automatically registered with this event.

Note that only GEV interface arrivals and removals are currently handled.

##### Parameters

|                                   |                                                                                                                                           |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>evtHandlerToRegister</code> | The event handler to register for the available interfaces                                                                                |
| <code>updateInterface</code>      | Determines whether or not <a href="#">UpdateInterfaceList()</a> is called before registering event for available interfaces on the system |

Implements [ISystem](#).

**14.165.3.10 RegisterLoggingEventHandler()**

```
virtual void RegisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Registers a logging event.

**Parameters**

|                |                                       |
|----------------|---------------------------------------|
| <i>handler</i> | The logging event handler to register |
|----------------|---------------------------------------|

Implements [ISystem](#).

**14.165.3.11 ReleaseInstance()**

```
virtual void ReleaseInstance ( ) [virtual]
```

This call releases the instance of the [System](#) Singleton for this process.

After successfully releasing the [System](#) instance the pointer returned by [GetInstance\(\)](#) will be invalid. Calling [ReleaseInstance](#) while a camera reference is still held will throw an error of type SPINNAKER\_ERR\_RESOU↔RCE\_IN\_USE.

**See also**

[Error](#)

[GetInstance\(\)](#)

Implements [ISystem](#).

**14.165.3.12 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 |

**Parameters**

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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](#).

**14.165.3.13 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 five levels of logging:

- Error - failures that are non-recoverable without user intervention.
- 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](#).

**14.165.3.14 UnregisterAllLoggingEventHandlers()**

```
virtual void UnregisterAllLoggingEventHandlers ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements [ISystem](#).

**14.165.3.15 UnregisterEventHandler()**

```
void UnregisterEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters an event handler for the system.

**Parameters**

|                                     |                                                 |
|-------------------------------------|-------------------------------------------------|
| <code>evtHandlerToUnregister</code> | The event handler to unregister from the system |
|-------------------------------------|-------------------------------------------------|

Implements [ISystem](#).

**14.165.3.16 UnregisterInterfaceEventHandler()**

```
void UnregisterInterfaceEventHandler (
    EventHandler & evtHandlerToUnregister ) [virtual]
```

Unregisters event handlers for all available interfaces that are found on the system.

**Parameters**

|                                     |                                                               |
|-------------------------------------|---------------------------------------------------------------|
| <code>evtHandlerToUnregister</code> | The event handler to unregister from the available interfaces |
|-------------------------------------|---------------------------------------------------------------|

Implements [ISystem](#).

**14.165.3.17 UnregisterLoggingEventHandler()**

```
virtual void UnregisterLoggingEventHandler (
    LoggingEventHandler & handler ) [virtual]
```

Unregisters a logging event.

**Parameters**

|                |                                         |
|----------------|-----------------------------------------|
| <i>handler</i> | The logging event handler to unregister |
|----------------|-----------------------------------------|

Implements [ISystem](#).

**14.165.3.18 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 <a href="#">UpdateInterfaceList()</a> is called before updating cameras for available interfaces on the system |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------|

**Returns**

True if cameras changed on interface and false otherwise.

Implements [ISystem](#).

**14.165.3.19 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](#).

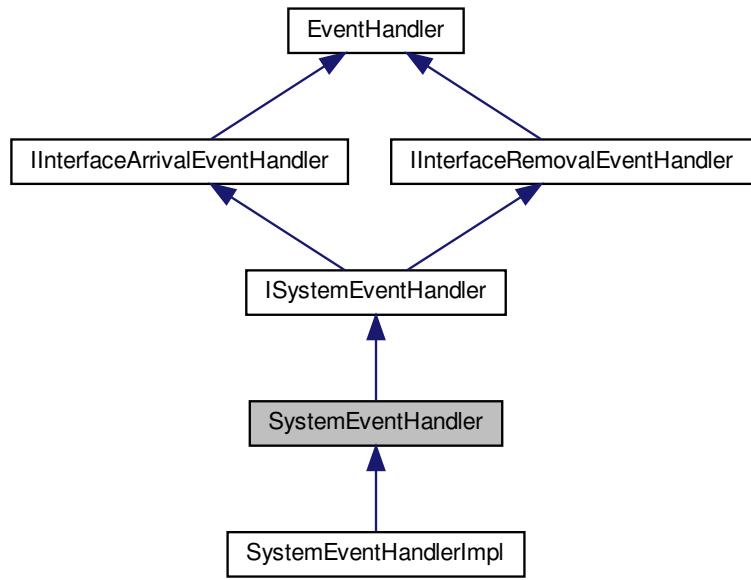
The documentation for this class was generated from the following file:

- [include/System.h](#)

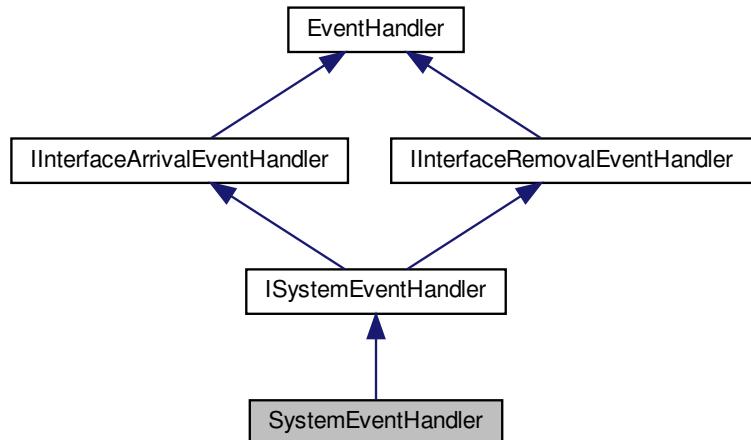
## 14.166 SystemEventHandler Class Reference

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 (std::string interfaceID)=0`  
*Interface arrival event callback.*
- `virtual void OnInterfaceRemoval (std::string interfaceID)=0`  
*Interface removal event callback.*

## Protected Member Functions

- `SystemEventHandler & operator= (const SystemEventHandler &)`  
*Assignment operator.*

## Additional Inherited Members

### 14.166.1 Detailed Description

A handler to interface arrival and removal events on the system.

Note that only GEV interface arrivals and removals are currently handled.

### 14.166.2 Constructor & Destructor Documentation

#### 14.166.2.1 SystemEventHandler()

`SystemEventHandler ()`

Default constructor.

#### 14.166.2.2 ~SystemEventHandler()

`virtual ~SystemEventHandler () [virtual]`

Virtual destructor.

### 14.166.3 Member Function Documentation

#### 14.166.3.1 OnInterfaceArrival()

`virtual void OnInterfaceArrival (`  
`std::string interfaceID ) [pure virtual]`

*Interface arrival event callback.*

Note that only GEV interface arrivals are currently handled.

**Parameters**

|                    |                                 |
|--------------------|---------------------------------|
| <i>interfaceID</i> | The ID of the arrived interface |
|--------------------|---------------------------------|

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

#### 14.166.3.2 OnInterfaceRemoval()

```
virtual void OnInterfaceRemoval (
    std::string interfaceID ) [pure virtual]
```

[Interface](#) removal event callback.

Note that only GEV interface removals are currently handled.

**Parameters**

|                    |                                 |
|--------------------|---------------------------------|
| <i>interfaceID</i> | The ID of the removed interface |
|--------------------|---------------------------------|

Implements [ISystemEventHandler](#).

Implemented in [SystemEventHandlerImpl](#).

#### 14.166.3.3 operator=( )

```
SystemEventHandler& operator= (
    const SystemEventHandler & ) [protected]
```

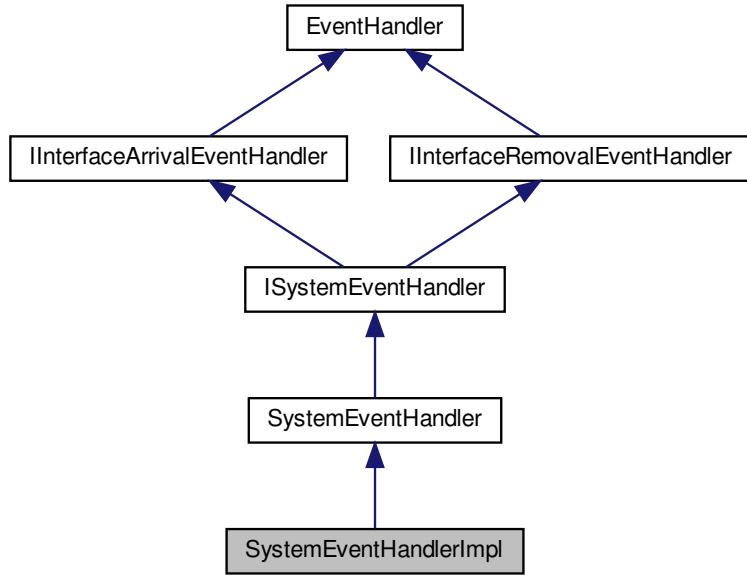
Assignment operator.

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

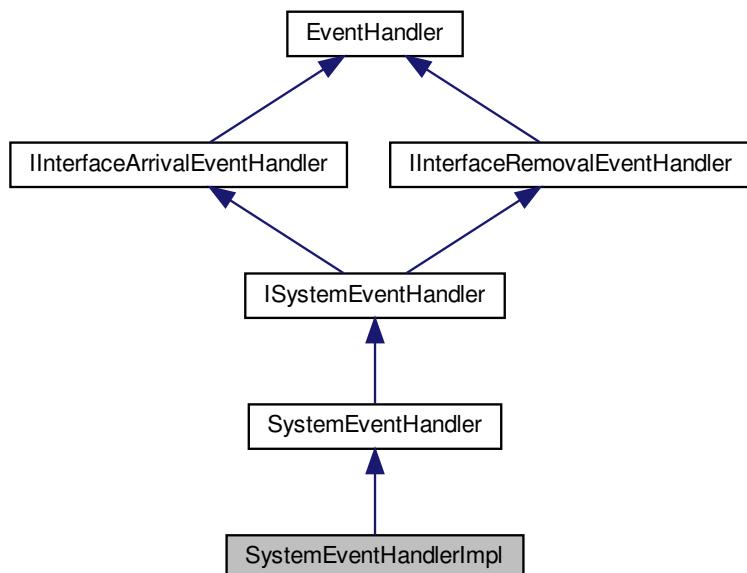
- [include/SystemEventHandler.h](#)

## 14.167 SystemEventHandlerImpl Class Reference

Inheritance diagram for SystemEventHandlerImpl:



Collaboration diagram for SystemEventHandlerImpl:



## Public Member Functions

- `SystemEventHandlerImpl (SystemPtr system)`
- `~SystemEventHandlerImpl ()`
- `void LockEventHandlerMutex ()`
- `void UnlockEventHandlerMutex ()`
- `void OnInterfaceArrival (std::string interfaceID)`  
*Interface arrival event callback.*
- `void OnInterfaceRemoval (std::string interfaceID)`  
*Interface removal event callback.*
- `void RegisterInterfaceEventToSystem ()`
- `void UnregisterInterfaceEventFromSystem ()`
- `void RegisterAllInterfaceEvents ()`
- `void UnregisterAllInterfaceEvents ()`

## Additional Inherited Members

### 14.167.1 Constructor & Destructor Documentation

#### 14.167.1.1 SystemEventHandlerImpl()

```
SystemEventHandlerImpl (
    SystemPtr system ) [inline]
```

#### 14.167.1.2 ~SystemEventHandlerImpl()

```
~SystemEventHandlerImpl ( ) [inline]
```

### 14.167.2 Member Function Documentation

#### 14.167.2.1 LockEventHandlerMutex()

```
void LockEventHandlerMutex ( ) [inline]
```

#### 14.167.2.2 OnInterfaceArrival()

```
void OnInterfaceArrival (
    std::string interfaceID ) [inline], [virtual]
```

Interface arrival event callback.

Note that only GEV interface arrivals are currently handled.

**Parameters**

|                    |                                 |
|--------------------|---------------------------------|
| <i>interfaceID</i> | The ID of the arrived interface |
|--------------------|---------------------------------|

Implements [SystemEventHandler](#).

**14.167.2.3 OnInterfaceRemoval()**

```
void OnInterfaceRemoval (
    std::string interfaceID) [inline], [virtual]
```

Interface removal event callback.

Note that only GEV interface removals are currently handled.

**Parameters**

|                    |                                 |
|--------------------|---------------------------------|
| <i>interfaceID</i> | The ID of the removed interface |
|--------------------|---------------------------------|

Implements [SystemEventHandler](#).

**14.167.2.4 RegisterAllInterfaceEvents()**

```
void RegisterAllInterfaceEvents () [inline]
```

**14.167.2.5 RegisterInterfaceEventToSystem()**

```
void RegisterInterfaceEventToSystem () [inline]
```

**14.167.2.6 UnlockEventHandlerMutex()**

```
void UnlockEventHandlerMutex () [inline]
```

**14.167.2.7 UnregisterAllInterfaceEvents()**

```
void UnregisterAllInterfaceEvents () [inline]
```

#### 14.167.2.8 UnregisterInterfaceEventFromSystem()

```
void UnregisterInterfaceEventFromSystem ( ) [inline]
```

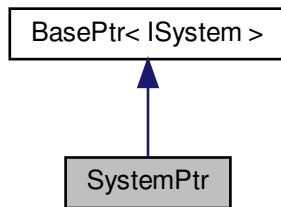
The documentation for this class was generated from the following file:

- [src/EnumerationEvents/EnumerationEvents.cpp](#)

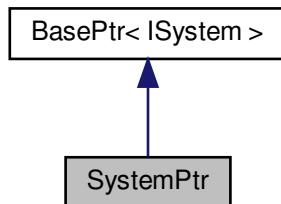
## 14.168 SystemPtr Class Reference

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

### 14.168.1 Detailed Description

A reference tracked pointer to a system object.

### 14.168.2 Constructor & Destructor Documentation

#### 14.168.2.1 SystemPtr() [1/4]

```
SystemPtr ( ) [inline]
```

Default constructor.

#### 14.168.2.2 SystemPtr() [2/4]

```
SystemPtr ( const int ) [inline]
```

Copy constructor.

#### 14.168.2.3 SystemPtr() [3/4]

```
SystemPtr ( const long ) [inline]
```

Copy constructor.

#### 14.168.2.4 SystemPtr() [4/4]

```
SystemPtr ( const std::nullptr_t ) [inline]
```

Copy constructor.

### 14.168.2.5 ~SystemPtr()

```
virtual ~SystemPtr (
    void ) [inline], [virtual]
```

Virtual destructor.

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

- include/SystemPtr.h

## 14.169 TIFFOption Struct Reference

Options for saving TIFF images.

### Public Types

- enum [CompressionMethod](#) {  
  [NONE](#) = 1,  
  [PACKBITS](#),  
  [DEFLATE](#),  
  [ADOBE\\_DEFLATE](#),  
  [CCITTFAZ3](#),  
  [CCITTFAZ4](#),  
  [LZW](#),  
  [JPEG](#) }

### Public Member Functions

- [TIFFOption \(\)](#)

### Public Attributes

- [CompressionMethod compression](#)  
*Compression method to use for encoding TIFF images.*
- unsigned int [reserved](#) [16]  
*Reserved for future use.*

### 14.169.1 Detailed Description

Options for saving TIFF images.

### 14.169.2 Member Enumeration Documentation

#### 14.169.2.1 CompressionMethod

```
enum CompressionMethod
```

## Enumerator

|               |                                                                                                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------------|
| NONE          | Save without any compression.                                                                                             |
| PACKBITS      | Save using PACKBITS compression.                                                                                          |
| DEFLATE       | Save using DEFLATE compression (ZLIB compression).                                                                        |
| ADOBE_DEFLATE | Save using ADOBE DEFLATE compression.                                                                                     |
| CCITTFAEX3    | Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.     |
| CCITTFAEX4    | Save using CCITT Group 4 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths.     |
| LZW           | Save using LZW compression.                                                                                               |
| JPEG          | Save using JPEG compression. This is only valid for 8-bit greyscale and 24-bit only. Default to LZW for other bit depths. |

**14.169.3 Constructor & Destructor Documentation****14.169.3.1 TiffOption()**

```
TiffOption ( ) [inline]
```

**14.169.4 Member Data Documentation****14.169.4.1 compression**

```
CompressionMethod compression
```

Compression method to use for encoding TIFF images.

**14.169.4.2 reserved**

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

## 14.170 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 & DeviceVendorName](#)  
*Description: Name of the remote device vendor.*
- [GenApi::IString & DeviceModelName](#)  
*Description: Name of the remote device model.*
- [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::IString & DeviceVersion](#)  
*Description: Version of the device.*
- [GenApi::IString & DeviceUserID](#)  
*Description: User Defined Name.*
- [GenApi::IString & DeviceDriverVersion](#)  
*Description: Version of the device driver.*
- [GenApi::IBoolean & DeviceIsUpdater](#)  
*Description: Indicates whether the device is in updater mode.*
- [GenApi::IEnumerationT< GevCCPEnum > & GevCCP](#)  
*Description: Controls the device access privilege of an application.*
- [GenApi::IEnumerationT< GUIXMLLocationEnum > & GUIXMLLocation](#)  
*Description: Sets the location to load GUI XML.*
- [GenApi::IString & GUIXMLPath](#)  
*Description: GUI XML Path.*
- [GenApi::IEnumerationT< GenICamXMLLocationEnum > & GenICamXMLLocation](#)  
*Description: Sets the location to load GenICam XML.*
- [GenApi::IString & GenICamXMLPath](#)  
*Description: GenICam XML Path.*
- [GenApi::IInteger & GevDeviceIPAddress](#)  
*Description: Current IP address of the GVCP interface of the selected remote device.*
- [GenApi::IInteger & GevDeviceSubnetMask](#)  
*Description: Current subnet mask of the GVCP interface of the selected remote device.*
- [GenApi::IInteger & GevDeviceMACAddress](#)  
*Description: 48-bit MAC address of the GVCP interface of the selected remote device.*

- [GenApi::IInteger & GevDeviceGateway](#)  
*Description: Current gateway IP address of the GVCP interface of the remote device.*
- [GenApi::IInteger & DeviceLinkSpeed](#)  
*Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.*
- [GenApi::IInteger & GevVersionMajor](#)  
*Description: Major version of the specification.*
- [GenApi::IInteger & 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::IInteger & GevDeviceReadAndWriteTimeout](#)  
*Description: The timeout in us for read/write operations to the camera.*
- [GenApi::IInteger & GevDeviceMaximumRetryCount](#)  
*Description: Maximum number of times to retry a read/write operation.*
- [GenApi::IInteger & 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::IInteger & 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::IInteger & GevDeviceForceIPAddress](#)  
*Description: Static IP address to set for the GVCP interface of the remote device.*
- [GenApi::IInteger & GevDeviceForceSubnetMask](#)  
*Description: Static subnet mask to set for GVCP interface of the remote device.*
- [GenApi::IInteger & GevDeviceForceGateway](#)  
*Description: Static gateway IP address to set for the GVCP interface of the remote device.*
- [GenApi::IBoolean & DeviceMulticastMonitorMode](#)  
*Description: Controls and indicates if the device is operating in as a Multicast Monitor.*
- [GenApi::IEnumerationT< DeviceEndianessMechanismEnum > & DeviceEndianessMechanism](#)  
*Description: Identifies the endianness handling mode.*
- [GenApi::IString & DeviceInstanceld](#)  
*Description: Visibility: Invisible.*
- [GenApi::IString & DeviceLocation](#)  
*Description: Device Location.*
- [GenApi::IEnumerationT< DeviceCurrentSpeedEnum > & DeviceCurrentSpeed](#)  
*Description: The USB Speed that the device is currently operating at.*
- [GenApi::IBoolean & DeviceU3VProtocol](#)  
*Description: Indicates whether the device is communicating in U3V Protocol.*

## Protected Member Functions

- [TransportLayerDevice \(\)](#)

## Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameralInternal](#)

### 14.170.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### 14.170.2 Constructor & Destructor Documentation

#### 14.170.2.1 [TransportLayerDevice\(\)](#) [1/2]

```
TransportLayerDevice (
    GenApi::INodeMap * nodeMapTLDevice )
```

#### 14.170.2.2 [~TransportLayerDevice\(\)](#)

```
~TransportLayerDevice ( )
```

#### 14.170.2.3 [TransportLayerDevice\(\)](#) [2/2]

```
TransportLayerDevice ( ) [protected]
```

### 14.170.3 Friends And Related Function Documentation

#### 14.170.3.1 [CameraBase](#)

```
friend class CameraBase [friend]
```

#### 14.170.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

#### 14.170.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

### 14.170.4 Member Data Documentation

#### 14.170.4.1 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

#### 14.170.4.2 DeviceCurrentSpeed

```
GenApi::IEnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed
```

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

#### 14.170.4.3 DeviceDisplayName

```
GenApi::IString& DeviceDisplayName
```

Description: User readable name of the device.

If this is not defined in the device this should be "VENDOR MODEL (ID)". Visibility: Expert

#### 14.170.4.4 DeviceDriverVersion

```
GenApi::IString& DeviceDriverVersion
```

Description: Version of the device driver.

Visibility: Expert

#### 14.170.4.5 DeviceEndianessMechanism

```
GenApi::IEnumeration<DeviceEndianessMechanismEnum>& DeviceEndianessMechanism
```

Description: Identifies the endianness handling mode.

Visibility: Expert

#### 14.170.4.6 DeviceID

```
GenApi::IString& DeviceID
```

Description: Interface-wide unique identifier of this device.

Visibility: Expert

#### 14.170.4.7 DeviceInstanceId

```
GenApi::IString& DeviceInstanceId
```

Description: Visibility: Invisible.

#### 14.170.4.8 DeviceIsUpdater

```
GenApi::IBoolean& DeviceIsUpdater
```

Description: Indicates whether the device is in updater mode.

Visibility: Expert

#### 14.170.4.9 DeviceLinkSpeed

```
GenApi::IInteger& DeviceLinkSpeed
```

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

#### 14.170.4.10 DeviceLocation

```
GenApi::IString& DeviceLocation
```

Description: Device Location.

Visibility: Expert

**14.170.4.11 DeviceModelName**

```
GenApi::IString& DeviceModelName
```

Description: Name of the remote device model.

Visibility: Beginner

**14.170.4.12 DeviceMulticastMonitorMode**

```
GenApi::IBoolean& DeviceMulticastMonitorMode
```

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

**14.170.4.13 DeviceSerialNumber**

```
GenApi::IString& DeviceSerialNumber
```

Description: Serial number of the remote device.

Visibility: Expert

**14.170.4.14 DeviceType**

```
GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType
```

Description: Transport layer type of the device.

Visibility: Expert

**14.170.4.15 DeviceU3VProtocol**

```
GenApi::IBoolean& DeviceU3VProtocol
```

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

**14.170.4.16 DeviceUserID**

```
GenApi::IString& DeviceUserID
```

Description: User Defined Name.

Visibility: Expert

#### 14.170.4.17 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the remote device vendor.

Visibility: Beginner

#### 14.170.4.18 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility: Beginner

#### 14.170.4.19 GenICamXMLLocation

`GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation`

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

Visibility: Beginner

#### 14.170.4.20 GenICamXMLPath

`GenApi::IString& GenICamXMLPath`

Description: [GenICam](#) XML Path.

Visibility: Beginner

#### 14.170.4.21 GevCCP

`GenApi::IEnumerationT<GevCCPEnum>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility: Beginner

#### 14.170.4.22 GevDeviceAutoForceIP

`GenApi:: ICommand& GevDeviceAutoForceIP`

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

**14.170.4.23 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

**14.170.4.24 GevDeviceForceGateway**

```
GenApi:: IInteger& GevDeviceForceGateway
```

Description: Static gateway IP address to set for the GVCP interface of the remote device.

Visibility: Expert

**14.170.4.25 GevDeviceForceIP**

```
GenApi:: ICommand& GevDeviceForceIP
```

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the Device using ForceIP command.

Visibility: Expert

**14.170.4.26 GevDeviceForceIPAddress**

```
GenApi:: IInteger& GevDeviceForceIPAddress
```

Description: Static IP address to set for the GVCP interface of the remote device.

Visibility: Expert

**14.170.4.27 GevDeviceForceSubnetMask**

```
GenApi:: IInteger& GevDeviceForceSubnetMask
```

Description: Static subnet mask to set for GVCP interface of the remote device.

Visibility: Expert

**14.170.4.28 GevDeviceGateway**

```
GenApi:: IInteger& GevDeviceGateway
```

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

#### 14.170.4.29 GevDeviceIPAddress

`GenApi::IInteger& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.170.4.30 GevDeviceIsWrongSubnet

`GenApi::IBoolean& GevDeviceIsWrongSubnet`

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

#### 14.170.4.31 GevDeviceMACAddress

`GenApi::IInteger& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.170.4.32 GevDeviceMaximumPacketSize

`GenApi::IInteger& GevDeviceMaximumPacketSize`

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

#### 14.170.4.33 GevDeviceMaximumRetryCount

`GenApi::IInteger& GevDeviceMaximumRetryCount`

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

#### 14.170.4.34 GevDeviceModeIsBigEndian

`GenApi::IBoolean& GevDeviceModeIsBigEndian`

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

#### 14.170.4.35 GevDevicePort

`GenApi::IInteger& GevDevicePort`

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.170.4.36 GevDeviceReadWriteTimeout

`GenApi::IInteger& GevDeviceReadWriteTimeout`

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

#### 14.170.4.37 GevDeviceSubnetMask

`GenApi::IInteger& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.170.4.38 GevVersionMajor

`GenApi::IInteger& GevVersionMajor`

Description: Major version of the specification.

Visibility: Expert

#### 14.170.4.39 GevVersionMinor

`GenApi::IInteger& GevVersionMinor`

Description: Minor version of the specification.

Visibility: Expert

#### 14.170.4.40 GUIXMLLocation

`GenApi::IEnumerationT<GUIXMLLocationEnum>& GUIXMLLocation`

Description: Sets the location to load GUI XML.

Visibility: Beginner

#### 14.170.4.41 GUIXMLPath

`GenApi::IString& GUIXMLPath`

Description: GUI XML Path.

Visibility: Beginner

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

- [include/TransportLayerDevice.h](#)

## 14.171 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::IInteger & GevInterfaceGatewaySelector](#)  
*Description: Selector for the different gateway entries for this interface.*
- [GenApi::IInteger & GevInterfaceGateway](#)  
*Description: IP address of the selected gateway entry of this interface.*
- [GenApi::IInteger & GevInterfaceMACAddress](#)  
*Description: 48-bit MAC address of this interface.*
- [GenApi::IInteger & GevInterfaceSubnetSelector](#)  
*Description: Selector for the subnet of this interface.*
- [GenApi::IInteger & GevInterfaceSubnetIPAddress](#)  
*Description: IP address of the selected subnet of this interface.*
- [GenApi::IInteger & GevInterfaceSubnetMask](#)  
*Description: Subnet mask of the selected subnet of this interface.*
- [GenApi::IInteger & GevInterfaceTransmitLinkSpeed](#)  
*Description: Transmit link speed of this interface in bits per second.*
- [GenApi::IInteger & GevInterfaceReceiveLinkSpeed](#)  
*Description: Receive link speed of this interface in bits per second.*
- [GenApi::IInteger & GevInterfaceMTU](#)  
*Description: Maximum transmission unit of this interface.*
- [GenApi::IEnumerationT< POEStatusEnum > & POEStatus](#)

- **GenApi::IEnumerationT< FilterDriverStatusEnum > & FilterDriverStatus**

Description: Reports and controls the interface's power over Ethernet status.
- **GenApi::IInteger & GevActionDeviceKey**

Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.
- **GenApi::IInteger & GevActionGroupKey**

Description: Key to authorize the action for the device.
- **GenApi::IInteger & GevActionGroupMask**

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- **GenApi::IInteger & GevActionTime**

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- **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::IInteger & DeviceCount**

Description: Number of compatible devices detected on current interface.
- **GenApi::IInteger & 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::IInteger & GevDeviceIPAddress**

Description: Current IP address of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceSubnetMask**

Description: Current subnet mask of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceGateway**

Description: Current gateway IP address of the GVCP interface of the selected remote device.
- **GenApi::IInteger & GevDeviceMACAddress**

Description: 48-bit MAC address of the GVCP interface of the selected remote device.
- **GenApi::IInteger & IncompatibleDeviceCount**

Description: Number of incompatible devices detected on current interface.
- **GenApi::IInteger & 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::IInteger & IncompatibleGevDeviceIPAddress](#)  
*Description:* Current IP address of the GVCP interface of the selected remote incompatible device.
- [GenApi::IInteger & IncompatibleGevDeviceSubnetMask](#)  
*Description:* Current subnet mask of the GVCP interface of the selected remote incompatible device.
- [GenApi::IInteger & 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::IInteger & GevDeviceForceIPAddress](#)  
*Description:* Static IP address to set for the GVCP interface of the selected remote device.
- [GenApi::IInteger & GevDeviceForceSubnetMask](#)  
*Description:* Static subnet mask to set for GVCP interface of the selected remote device.
- [GenApi::IInteger & 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](#)

### 14.171.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### 14.171.2 Constructor & Destructor Documentation

#### 14.171.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
    GenApi::INodeMap * nodeMapTLDevice )
```

#### 14.171.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

#### 14.171.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

### 14.171.3 Friends And Related Function Documentation

#### 14.171.3.1 IInterface

```
friend class IInterface [friend]
```

#### 14.171.3.2 Interface

```
friend class Interface [friend]
```

#### 14.171.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

### 14.171.4 Member Data Documentation

#### 14.171.4.1 ActionCommand

```
GenApi:: ICommand& ActionCommand
```

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

#### 14.171.4.2 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

#### 14.171.4.3 DeviceCount

```
GenApi::IInteger& DeviceCount
```

Description: Number of compatible devices detected on current interface.

Visibility: Expert

#### 14.171.4.4 DeviceID

```
GenApi::IString& DeviceID
```

Description: [Interface](#) wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.5 DeviceModelName

```
GenApi::IString& DeviceModelName
```

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.6 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

#### 14.171.4.7 DeviceSerialNumber

```
GenApi::IString& DeviceSerialNumber
```

Description: Serial number of the selected remote device.

Visibility: Expert

#### 14.171.4.8 DeviceUnlock

`GenApi::IString& DeviceUnlock`

Description: Unlocks devices for internal use.

Visibility: Expert

#### 14.171.4.9 DeviceUpdateList

`GenApi:: ICommand& DeviceUpdateList`

Description: Updates the internal device list.

Visibility: Expert

#### 14.171.4.10 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.11 FilterDriverStatus

`GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus`

Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.

Visibility: Expert

#### 14.171.4.12 GevActionDeviceKey

`GenApi::IInteger& GevActionDeviceKey`

Description: Key to authorize the action for the device.

Visibility: Expert

#### 14.171.4.13 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

#### 14.171.4.14 GevActionGroupMask

`GenApi::IInteger& GevActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

#### 14.171.4.15 GevActionTime

`GenApi::IInteger& GevActionTime`

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

#### 14.171.4.16 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

#### 14.171.4.17 GevDeviceForceGateway

`GenApi::IInteger& GevDeviceForceGateway`

Description: Static gateway IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.18 GevDeviceForceIP

`GenApi:: ICommand& GevDeviceForceIP`

Description: Apply the force IP settings (GevDeviceForceIPAddress, GevDeviceForceSubnetMask and GevDeviceForceGateway) in the selected remote device using ForceIP command.

Visibility: Expert

#### 14.171.4.19 GevDeviceForceIPAddress

`GenApi::IInteger& GevDeviceForceIPAddress`

Description: Static IP address to set for the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.20 GevDeviceForceSubnetMask

`GenApi::IInteger& GevDeviceForceSubnetMask`

Description: Static subnet mask to set for GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.21 GevDeviceGateway

`GenApi::IInteger& GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.22 GevDeviceIPAddress

`GenApi::IInteger& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.23 GevDeviceMACAddress

`GenApi::IInteger& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.24 GevDeviceSubnetMask

`GenApi::IInteger& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

#### 14.171.4.25 GevInterfaceGateway

`GenApi::IInteger& GevInterfaceGateway`

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

**14.171.4.26 GevInterfaceGatewaySelector**

`GenApi::IInteger& GevInterfaceGatewaySelector`

Description: Selector for the different gateway entries for this interface.

Visibility: Expert

**14.171.4.27 GevInterfaceMACAddress**

`GenApi::IInteger& GevInterfaceMACAddress`

Description: 48-bit MAC address of this interface.

Visibility: Expert

**14.171.4.28 GevInterfaceMTU**

`GenApi::IInteger& GevInterfaceMTU`

Description: Maximum transmission unit of this interface.

Visibility: Expert

**14.171.4.29 GevInterfaceReceiveLinkSpeed**

`GenApi::IInteger& GevInterfaceReceiveLinkSpeed`

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

**14.171.4.30 GevInterfaceSubnetIPAddress**

`GenApi::IInteger& GevInterfaceSubnetIPAddress`

Description: IP address of the selected subnet of this interface.

Visibility: Expert

**14.171.4.31 GevInterfaceSubnetMask**

`GenApi::IInteger& GevInterfaceSubnetMask`

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

**14.171.4.32 GevInterfaceSubnetSelector**

`GenApi::IInteger& GevInterfaceSubnetSelector`

Description: Selector for the subnet of this interface.

Visibility: Expert

**14.171.4.33 GevInterfaceTransmitLinkSpeed**

`GenApi::IInteger& GevInterfaceTransmitLinkSpeed`

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

**14.171.4.34 HostAdapterDriverVersion**

`GenApi::IString& HostAdapterDriverVersion`

Description: Driver version of the interface's host adapter.

Visibility: Expert

**14.171.4.35 HostAdapterName**

`GenApi::IString& HostAdapterName`

Description: User readable name of the interface's host adapter.

Visibility: Expert

**14.171.4.36 HostAdapterVendor**

`GenApi::IString& HostAdapterVendor`

Description: User readable name of the host adapter's vendor.

Visibility: Expert

**14.171.4.37 IncompatibleDeviceCount**

`GenApi::IInteger& IncompatibleDeviceCount`

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

#### 14.171.4.38 IncompatibleDeviceID

`GenApi::IString& IncompatibleDeviceID`

Description: [Interface](#) wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.39 IncompatibleDeviceModelName

`GenApi::IString& IncompatibleDevicemodelName`

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.40 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

#### 14.171.4.41 IncompatibleDeviceVendorName

`GenApi::IString& IncompatibleDeviceVendorName`

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

#### 14.171.4.42 IncompatibleGevDeviceIPAddress

`GenApi::IInteger& IncompatibleGevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

#### 14.171.4.43 IncompatibleGevDeviceMACAddress

`GenApi::IInteger& IncompatibleGevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

#### 14.171.4.44 IncompatibleGevDeviceSubnetMask

`GenApi::IInteger& IncompatibleGevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

#### 14.171.4.45 InterfaceDisplayName

`GenApi::IString& InterfaceDisplayName`

Description: User readable name of the selected interface.

Visibility: Expert

#### 14.171.4.46 InterfaceID

`GenApi::IString& InterfaceID`

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

#### 14.171.4.47 InterfaceType

`GenApi::IEnumerationT<InterfaceTypeEnum>& InterfaceType`

Description: Transport layer type of the interface.

Visibility: Expert

#### 14.171.4.48 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](#)

### 14.172 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::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 & StreamLostFrameCount`  
*Description: Number of lost frames due to queue underrun.*
- `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 & StreamCRCCheckEnable`  
*Description: Enables or disables CRC checks on received images.*
- `GenApi::IBoolean & GevPacketResendMode`  
*Description: Enables or disables the packet resend mechanism.*
- `GenApi::IInteger & GevMaximumNumberResendRequests`  
*Description: Maximum number of resend requests per image.*
- `GenApi::IInteger & GevPacketResendTimeout`  
*Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.*
- `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::IInteger & GevTotalPacketCount`

- Description: Displays number of packets received on this stream.*
- [GenApi::IInteger & GevFailedPacketCount](#)
  - Description: Displays number of packets missed on this stream.*
- [GenApi::IInteger & GevResendPacketCount](#)
  - Description: Displays number of packets received after retransmit request on this stream.*
- [GenApi::IInteger & StreamFailedBufferCount](#)
  - Description: Displays number of incomplete images with missing leader/trailer information.*
- [GenApi::IInteger & GevResendRequestCount](#)
  - Description: Displays number of packets requested to be retransmitted on this stream.*
- [GenApi::IInteger & StreamBlockTransferSize](#)
  - Description: Controls the image breakup size that should be used on this stream.*

## Protected Member Functions

- [TransportLayerStream \(\)](#)

## Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameralInternal](#)

### 14.172.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### 14.172.2 Constructor & Destructor Documentation

#### 14.172.2.1 TransportLayerStream() [1/2]

```
TransportLayerStream (
    GenApi::INodeMap * nodeMapTLDevice )
```

#### 14.172.2.2 ~TransportLayerStream()

```
~TransportLayerStream ( )
```

#### 14.172.2.3 TransportLayerStream() [2/2]

```
TransportLayerStream ( ) [protected]
```

### 14.172.3 Friends And Related Function Documentation

#### 14.172.3.1 CameraBase

```
friend class CameraBase [friend]
```

#### 14.172.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

#### 14.172.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

### 14.172.4 Member Data Documentation

#### 14.172.4.1 GevFailedPacketCount

`GenApi::IInteger& GevFailedPacketCount`

Description: Displays number of packets missed on this stream.

Visibility: Expert

#### 14.172.4.2 GevMaximumNumberResendRequests

`GenApi::IInteger& GevMaximumNumberResendRequests`

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

#### 14.172.4.3 GevPacketResendMode

`GenApi::IBoolean& GevPacketResendMode`

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

#### 14.172.4.4 GevPacketResendTimeout

`GenApi::IInteger& GevPacketResendTimeout`

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

#### 14.172.4.5 GevResendPacketCount

`GenApi::IInteger& GevResendPacketCount`

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

#### 14.172.4.6 GevResendRequestCount

`GenApi::IInteger& GevResendRequestCount`

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

#### 14.172.4.7 GevTotalPacketCount

`GenApi::IInteger& GevTotalPacketCount`

Description: Displays number of packets received on this stream.

Visibility: Expert

#### 14.172.4.8 StreamAnnounceBufferMinimum

`GenApi::IInteger& StreamAnnounceBufferMinimum`

Description: Minimal number of buffers to announce to enable selected buffer handling mode.

Visibility: Expert

#### 14.172.4.9 StreamAnnouncedBufferCount

`GenApi::IInteger& 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

#### 14.172.4.10 StreamBlockTransferSize

`GenApi::IInteger& StreamBlockTransferSize`

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

#### 14.172.4.11 StreamBufferAlignment

`GenApi::IInteger& StreamBufferAlignment`

Description: Alignment size in bytes of the buffer passed to DSAnnounceBuffer.

Visibility: Expert

#### 14.172.4.12 StreamBufferCountManual

`GenApi::IInteger& StreamBufferCountManual`

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

#### 14.172.4.13 StreamBufferCountMax

`GenApi::IInteger& 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

#### 14.172.4.14 StreamBufferCountMode

`GenApi::IEnumerationT<StreamBufferCountModeEnum>& StreamBufferCountMode`

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

Locked to Manual mode on 32-bit Windows due to memory constraints. Visibility: Expert

#### 14.172.4.15 StreamBufferCountResult

`GenApi::IInteger& 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

#### 14.172.4.16 StreamBufferHandlingMode

`GenApi::IEnumeration<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode`

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

#### 14.172.4.17 StreamChunkCountMaximum

`GenApi::IInteger& StreamChunkCountMaximum`

Description: Maximum number of chunks to be expected in a buffer.

Visibility: Expert

#### 14.172.4.18 StreamCRCCheckEnable

`GenApi::IBoolean& StreamCRCCheckEnable`

Description: Enables or disables CRC checks on received images.

Visibility: Expert

#### 14.172.4.19 StreamDeliveredFrameCount

`GenApi::IInteger& StreamDeliveredFrameCount`

Description: Number of delivered frames since last acquisition start.

It is not reset until the stream is closed. Visibility: Expert

#### 14.172.4.20 StreamFailedBufferCount

`GenApi::IInteger& StreamFailedBufferCount`

Description: Displays number of incomplete images with missing leader/trailer information.

Visibility: Expert

#### 14.172.4.21 StreamID

`GenApi::IString& StreamID`

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

#### 14.172.4.22 StreamInputBufferCount

`GenApi::IInteger& StreamInputBufferCount`

Description: Number of buffers in the input buffer pool plus the buffers(s) currently being filled.

Visibility: Expert

#### 14.172.4.23 StreamIsGrabbing

`GenApi::IBoolean& StreamIsGrabbing`

Description: Flag indicating whether the acquisition engine is started or not.

Visibility: Expert

#### 14.172.4.24 StreamLostFrameCount

`GenApi::IInteger& StreamLostFrameCount`

Description: Number of lost frames due to queue underrun.

This number is initialized with zero at the time the stream is opened and incremented every time the data could not be acquired because there was no buffer in the input buffer pool. It is not reset until the stream is closed. Visibility: Expert

#### 14.172.4.25 StreamOutputBufferCount

`GenApi::IInteger& StreamOutputBufferCount`

Description: Number of buffers in the output buffer queue.

Visibility: Expert

#### 14.172.4.26 StreamStartedFrameCount

`GenApi::IInteger& 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

#### 14.172.4.27 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`

## 14.173 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::IBoolean & EnumerateGEVInterfaces`  
`Description: Enables or disables enumeration of GEV Interfaces.`
- `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 & TLD displayName`  
`Description: User readable name of the GenTL Producer.`
- `GenApi::IString & TLPPath`  
`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::IInteger & GenTLVersionMajor`  
`Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.`
- `GenApi::IInteger & GenTLVersionMinor`  
`Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.`
- `GenApi::IInteger & GenTLSFNCVersionMajor`

*Description:* Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- [GenApi::IInteger & GenTLSFNCVersionMinor](#)

*Description:* Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- [GenApi::IInteger & GenTLSFNCVersionSubMinor](#)

*Description:* Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

- [GenApi::IInteger & GevVersionMajor](#)

*Description:* Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

- [GenApi::IInteger & 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::IInteger & 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::IInteger & GevInterfaceMACAddress](#)

*Description:* 48-bit MAC address of the selected interface.

- [GenApi::IInteger & GevInterfaceDefaultIPAddress](#)

*Description:* IP address of the first subnet of the selected interface.

- [GenApi::IInteger & GevInterfaceDefaultSubnetMask](#)

*Description:* Subnet mask of the first subnet of the selected interface.

- [GenApi::IInteger & GevInterfaceDefaultGateway](#)

*Description:* Gateway of the selected interface.

## Protected Member Functions

- [TransportLayerSystem \(\)](#)

## Friends

- class [System](#)
- class [ISystem](#)
- class [SystemPtrInternal](#)

### 14.173.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

### 14.173.2 Constructor & Destructor Documentation

**14.173.2.1 TransportLayerSystem() [1/2]**

```
TransportLayerSystem (
    GenApi::INodeMap * nodeMapTLDevice )
```

**14.173.2.2 ~TransportLayerSystem()**

```
~TransportLayerSystem ( )
```

**14.173.2.3 TransportLayerSystem() [2/2]**

```
TransportLayerSystem ( ) [protected]
```

**14.173.3 Friends And Related Function Documentation****14.173.3.1 ISystem**

```
friend class ISystem [friend]
```

**14.173.3.2 System**

```
friend class System [friend]
```

**14.173.3.3 SystemPtrInternal**

```
friend class SystemPtrInternal [friend]
```

**14.173.4 Member Data Documentation**

#### 14.173.4.1 EnumerateGEVInterfaces

`GenApi::IBoolean& EnumerateGEVInterfaces`

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

#### 14.173.4.2 GenTSLFNCVersionMajor

`GenApi::IInteger& GenTSLFNCVersionMajor`

Description: Major version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

#### 14.173.4.3 GenTSLFNCVersionMinor

`GenApi::IInteger& GenTSLFNCVersionMinor`

Description: Minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

#### 14.173.4.4 GenTSLFNCVersionSubMinor

`GenApi::IInteger& GenTSLFNCVersionSubMinor`

Description: Sub minor version number of the GenTL Standard Features Naming Convention that was used to create the GenTL Producer's XML.

Visibility: Expert

#### 14.173.4.5 GenTlVersionMajor

`GenApi::IInteger& GenTlVersionMajor`

Description: Major version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

#### 14.173.4.6 GenTlVersionMinor

`GenApi::IInteger& GenTlVersionMinor`

Description: Minor version number of the GenTL specification the GenTL Producer implementation complies with.

Visibility: Expert

#### 14.173.4.7 GevInterfaceDefaultGateway

`GenApi::IInteger& GevInterfaceDefaultGateway`

Description: Gateway of the selected interface.

Visibility: Expert

#### 14.173.4.8 GevInterfaceDefaultIPAddress

`GenApi::IInteger& 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

#### 14.173.4.9 GevInterfaceDefaultSubnetMask

`GenApi::IInteger& 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

#### 14.173.4.10 GevInterfaceMACAddress

`GenApi::IInteger& 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

#### 14.173.4.11 GevVersionMajor

`GenApi::IInteger& GevVersionMajor`

Description: Major version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

#### 14.173.4.12 GevVersionMinor

`GenApi::IInteger& GevVersionMinor`

Description: Minor version number of the GigE Vision specification the GenTL Producer implementation complies to.

Visibility: Expert

**14.173.4.13 InterfaceDisplayName**

```
GenApi::IString& InterfaceDisplayName
```

Description: A user-friendly name of the selected [Interface](#).

Visibility: Beginner

**14.173.4.14 InterfaceID**

```
GenApi::IString& InterfaceID
```

Description: GenTL Producer wide unique identifier of the selected interface.

Visibility: Beginner

**14.173.4.15 InterfaceSelector**

```
GenApi::IInteger& 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

**14.173.4.16 InterfaceUpdateList**

```
GenApi:: ICommand& 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

**14.173.4.17 TLDisplayName**

```
GenApi::IString& TLDisplayName
```

Description: User readable name of the GenTL Producer.

Visibility: Expert

**14.173.4.18 TLFileName**

```
GenApi::IString& TLFileName
```

Description: Filename including extension of the GenTL Producer.

Visibility: Expert

#### 14.173.4.19 TLID

`GenApi::IString& TLID`

Description: Unique identifier of the GenTL Producer like a GUID.

Visibility: Expert

#### 14.173.4.20 TLModelName

`GenApi::IString& TLModelName`

Description: Name of the GenTL Producer to distinguish different kinds of GenTL Producer implementations from one vendor.

Visibility: Beginner

#### 14.173.4.21 TLPath

`GenApi::IString& TLPath`

Description: Full path to the GenTL Producer including filename and extension.

Visibility: Expert

#### 14.173.4.22 TLType

`GenApi::IEnumerationT<TLTypeEnum>& TLType`

Description: Transport layer type of the GenTL Producer implementation.

Visibility: Expert

#### 14.173.4.23 TLVendorName

`GenApi::IString& TLVendorName`

Description: Name of the GenTL Producer vendor.

Visibility: Beginner

#### 14.173.4.24 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](#)

## 14.174 U3V\_CHUNK\_TRAILER Struct Reference

header of a GVCP request packet

### Public Attributes

- `uint32_t ChunkID`
- `uint32_t ChunkLength`

#### 14.174.1 Detailed Description

header of a GVCP request packet

#### 14.174.2 Member Data Documentation

##### 14.174.2.1 ChunkID

`uint32_t ChunkID`

##### 14.174.2.2 ChunkLength

`uint32_t ChunkLength`

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

- `include/SpinGenApi/ChunkAdapterU3V.h`

## 14.175 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`

### 14.175.1 Detailed Description

U3V/GenCP command header.

### 14.175.2 Member Data Documentation

#### 14.175.2.1 CommandId

```
uint16_t CommandId
```

#### 14.175.2.2 Flags

```
uint16_t Flags
```

#### 14.175.2.3 Length

```
uint16_t Length
```

#### 14.175.2.4 Prefix

```
uint32_t Prefix
```

#### 14.175.2.5 ReqId

```
uint16_t ReqId
```

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

- include/SpinGenApi/[EventAdapterU3V.h](#)

## 14.176 U3V\_EVENT\_DATA Struct Reference

U3V/GenCP EVENT\_CMD specific command data.

## Public Attributes

- `uint16_t Reserved`
- `uint16_t EventId`
- `uint64_t Timestamp`

### 14.176.1 Detailed Description

U3V/GenCP EVENT\_CMD specific command data.

### 14.176.2 Member Data Documentation

#### 14.176.2.1 EventId

```
uint16_t EventId
```

#### 14.176.2.2 Reserved

```
uint16_t Reserved
```

#### 14.176.2.3 Timestamp

```
uint64_t Timestamp
```

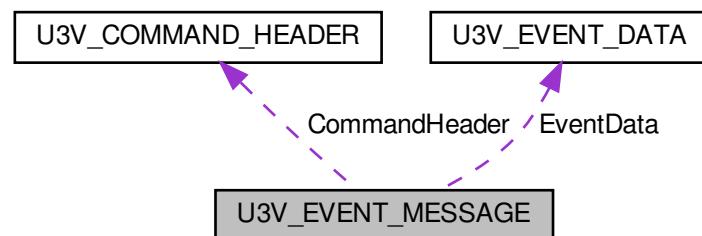
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

## 14.177 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

### 14.177.1 Detailed Description

Entire event data message (without the variable-sized data field)

### 14.177.2 Member Data Documentation

#### 14.177.2.1 CommandHeader

`U3V_COMMAND_HEADER` CommandHeader

#### 14.177.2.2 EventData

`U3V_EVENT_DATA` EventData

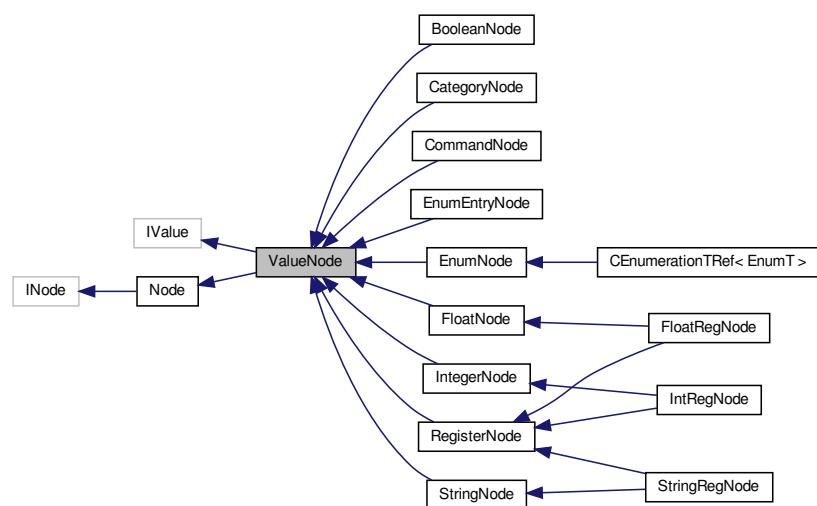
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

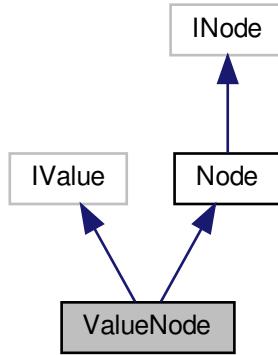
## 14.178 ValueNode Class Reference

[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

### 14.178.1 Detailed Description

[Interface](#) for value properties.

### 14.178.2 Constructor & Destructor Documentation

**14.178.2.1 ValueNode() [1/2]**

```
ValueNode ( )
```

Constructor.

**14.178.2.2 ValueNode() [2/2]**

```
ValueNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam IValue](#)

**14.178.2.3 ~ValueNode()**

```
~ValueNode ( )
```

Destructor.

**14.178.3 Member Function Documentation****14.178.3.1 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) |

**14.178.3.2 GetNode()**

```
virtual INode* GetNode ( ) [virtual]
```

### 14.178.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid ( ) const [virtual]
```

Checks if the value comes from cache or is requested from another node.

### 14.178.3.4 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](#).

### 14.178.3.5 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](#)

## 14.179 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*

### 14.179.1 Detailed Description

Version.

### 14.179.2 Member Data Documentation

#### 14.179.2.1 Major

`uint16_t Major`

#### 14.179.2.2 Minor

`uint16_t Minor`

*a is incompatible with b if a != b*

#### 14.179.2.3 SubMinor

`uint16_t SubMinor`

*a is incompatible b a > b*

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

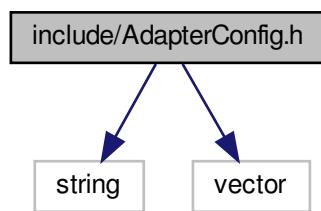
- include/SpinGenApi/[GCTypes.h](#)

# **Chapter 15**

## **File Documentation**

- 15.1 doc/spindocs/C++/GettingStarted.dox File Reference**
- 15.2 doc/spindocs/C++/ProgrammerGuide.dox File Reference**
- 15.3 doc/spindocs/shared/Benefits.dox File Reference**
- 15.4 doc/spindocs/shared/FlyCapture2Comparison.dox File Reference**
- 15.5 doc/spindocs/shared/GenICamGenTL.dox File Reference**
- 15.6 doc/spindocs/shared/Licensing.dox File Reference**
- 15.7 include/AdapterConfig.h File Reference**

Include dependency graph for AdapterConfig.h:



## Classes

- struct `IplInfo`
- struct `AdapterInfo`
- class `AdapterConfigException`

## Namespaces

- `AdapterConfig`

## Macros

- `#define ADAPTERCONFIG_API __declspec(dllexport)`

## Enumerations

- enum `AdapterConfigErr` {  
    `IP_ADDRESS_INVALID`,  
    `IP_ADDRESS_IS_NOT_V4`,  
    `IP_ADDRESS_TOO_LARGE`,  
    `IP_ADDRESS_TOO_SMALL`,  
    `HOST_ADDRESS_ZERO`,  
    `SUBNET_MASK_INVALID`,  
    `VALID_SUBNET_NOT_FOUND` }

## Functions

- `ADAPTERCONFIG_API std::vector< AdapterInfo > RetrieveAllAdapters ()`
- `ADAPTERCONFIG_API void AutoPopulateAdapterInfo (std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters)`
- `ADAPTERCONFIG_API void AutoPopulateAdvancedProperties (std::vector< AdapterInfo > &adaptersToConfigure)`
- `ADAPTERCONFIG_API void PopulateAdapterIplInfo (IplInfo startingIplInfo, std::vector< AdapterInfo > &adaptersToConfigure, const std::vector< AdapterInfo > &allAdapters)`
- `ADAPTERCONFIG_API void ValidateIpAddress (const std::string &ipAddr, unsigned int subnetMaskLength)`
- `ADAPTERCONFIG_API bool IsValidIpAddress (const std::string &ipAddr)`
- `ADAPTERCONFIG_API bool IsValidSubnetMask (const std::string &subnetMask)`
- `ADAPTERCONFIG_API bool IsOnSameSubnet (const std::string &ipAddrStr1, const std::string &ipAddrStr2, const unsigned int subnetMaskLength)`
- `ADAPTERCONFIG_API unsigned int GetSubnetMaskLength (const std::string &subnetMask)`
- `ADAPTERCONFIG_API std::string GetEnumerationLogFileName ()`
- `ADAPTERCONFIG_API std::string GetConfigLogFileName ()`
- `ADAPTERCONFIG_API void ConfigureAdapter (AdapterInfo &adapter, bool configureIP, bool configureAdvancedProperties)`
- `ADAPTERCONFIG_API unsigned int GetAutoSubnetMaskLength ()`
- `ADAPTERCONFIG_API std::string GetAutoSubnetMask ()`
- `ADAPTERCONFIG_API std::string GetMaxIpAddress ()`
- `ADAPTERCONFIG_API std::string GetMinIpAddress ()`
- `ADAPTERCONFIG_API std::string GetAutoGigabitDesc ()`
- `ADAPTERCONFIG_API std::string GetAuto10GDesc ()`
- `ADAPTERCONFIG_API std::string GetAutoStartIp ()`

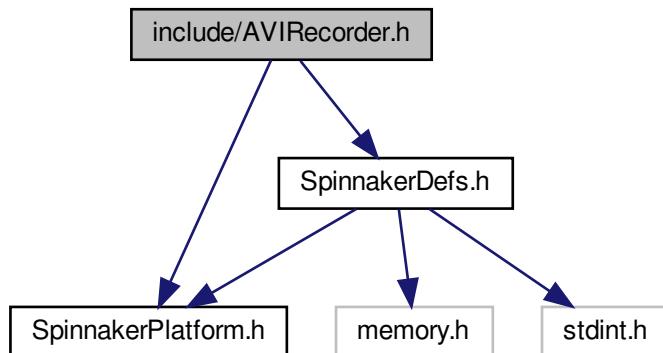
### 15.7.1 Macro Definition Documentation

#### 15.7.1.1 ADAPTERCONFIG\_API

```
#define ADAPTERCONFIG_API __declspec(dllimport)
```

## 15.8 include/AVIRecorder.h File Reference

Include dependency graph for AVIRecorder.h:



## Namespaces

- [Spinnaker](#)

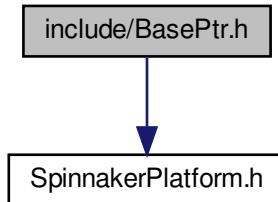
## Functions

- class [DEPRECATED\\_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER\_API A←  
VIRecorder

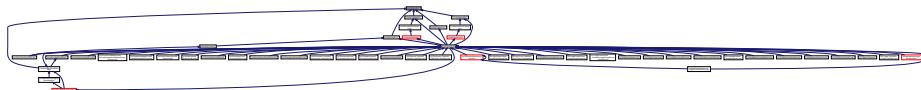
*Provides the functionality for the user to record images to an AVI file.*

## 15.9 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](#)

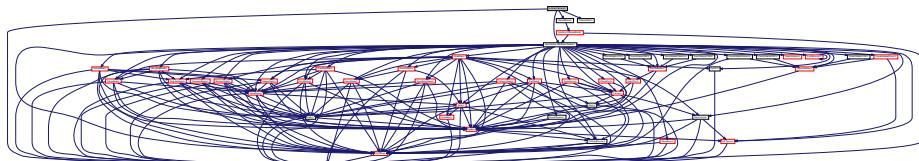
### Functions

- template<class T , class B >  
bool [operator==](#) (const std::nullptr\_t, const BasePtr< T, B > &rhs)

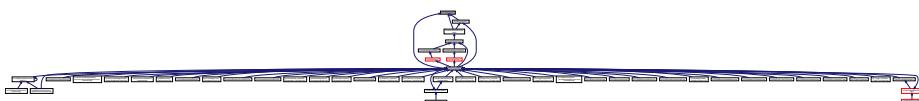
*Pointer equal.*

## 15.10 include/Camera.h File Reference

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



### Classes

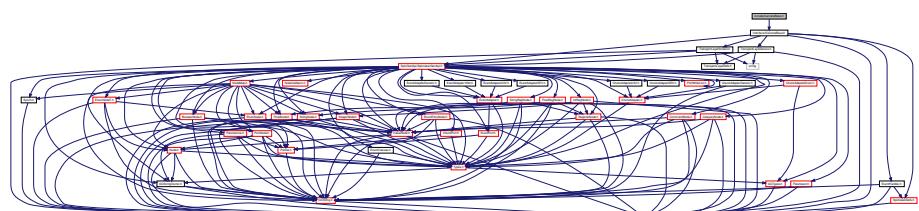
- class [Camera](#)  
*The camera object class.*

### Namespaces

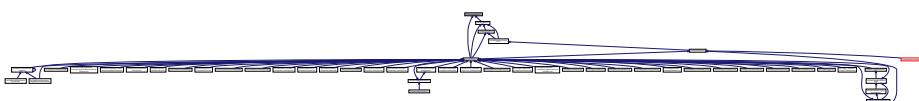
- [Spinnaker](#)

## 15.11 include/CameraBase.h File Reference

Include dependency graph for CameraBase.h:



This graph shows which files directly or indirectly include this file:



## Classes

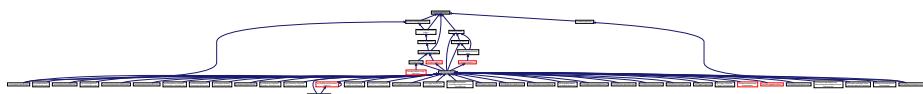
- class [CameraBase](#)  
*The base class for the camera object.*

## Namespaces

- [Spinnaker](#)

## 15.12 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_DEVICEREGISTERSENDIANNESS` }
- 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_DEVICEPOWERSUPPLYSELECTION }

• enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICETEMPERATURESELECTION }

• 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_AUTOALGORITHMSELECTION }

• 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_GEVSUPPORTEOPTIONSELECTOR }

• 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 FileModeEnums {
    FileMode_Read,
    FileMode_Write,
    FileMode_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 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,
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_Deglitch,
    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_EXPOSUREACTIVE MODE }

• 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_DEVICESTREAMCHANNELENDIANCESS }

• 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 GevGVSPExtendedIDModeEnums {
    GevGVSPExtendedIDMode_Off,
    GevGVSPExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDDIDMODE }

• 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 }

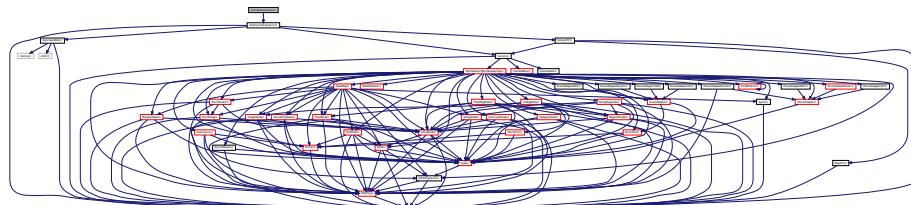
• enum CxpConnectionTestModeEnums {
  CxpConnectionTestMode_Off,
  CxpConnectionTestMode_Mode1,
  NUM_CXP CONNECTIONTESTMODE }

• enum CxpPoCxpStatusEnums {
  CxpPoCxpStatus_Auto,
  CxpPoCxpStatus_Off,
  CxpPoCxpStatus_Tripped,
  NUM_CXPPOCXPSTATUS }

```

## 15.13 include/CameraList.h File Reference

Include dependency graph for CameraList.h:



This graph shows which files directly or indirectly include this file:



## Classes

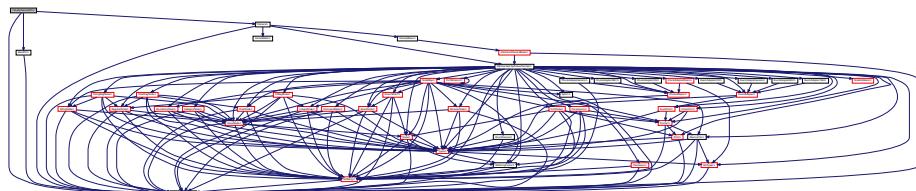
- class [CameraList](#)  
*Used to hold a list of camera objects.*

## Namespaces

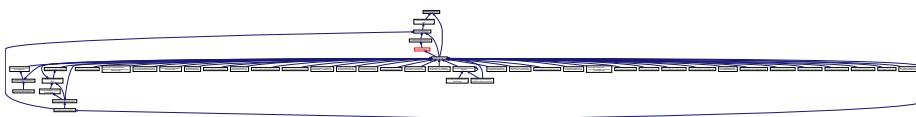
- [Spinnaker](#)

## 15.14 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](#)

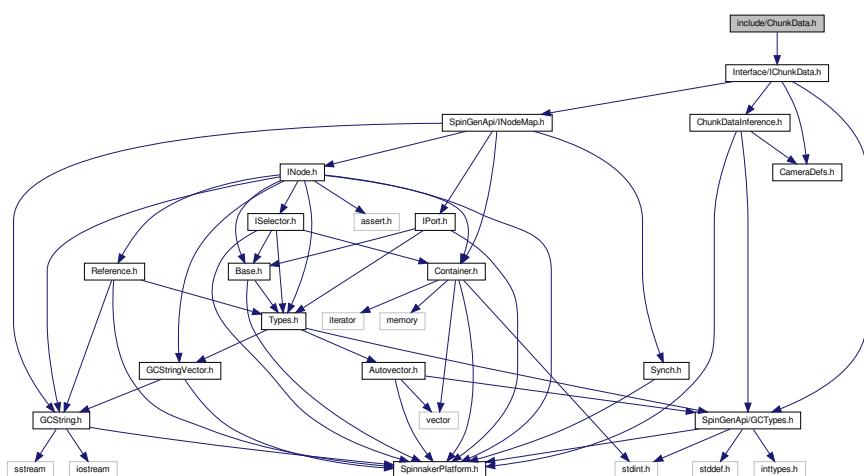
*A reference tracked pointer to a camera object.*

### Namespaces

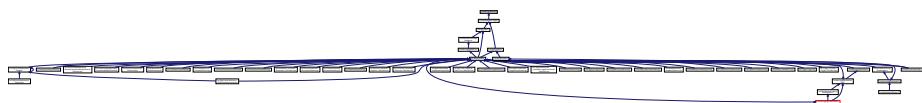
- [Spinnaker](#)

## 15.15 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](#)

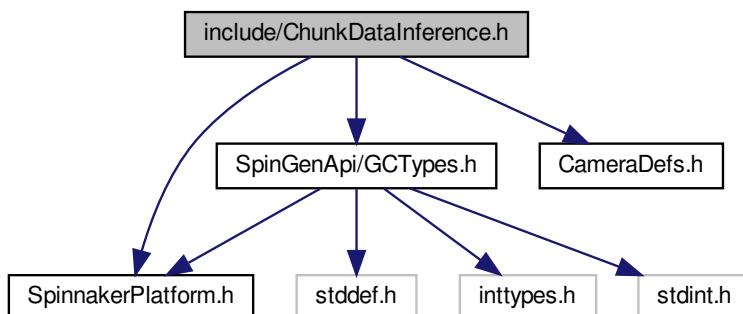
*The chunk data which contains additional information about an image.*

## Namespaces

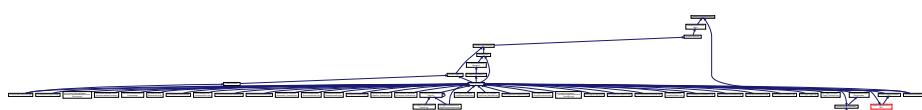
- [Spinnaker](#)

## 15.16 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

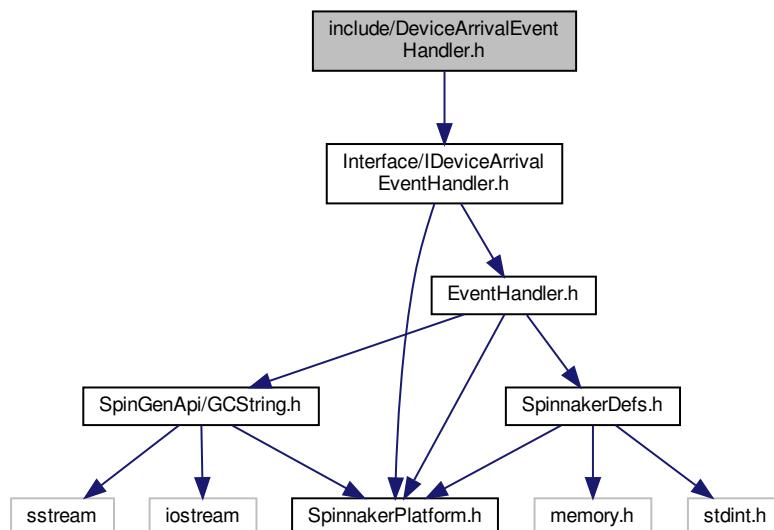
- Spinnaker

## Enumerations

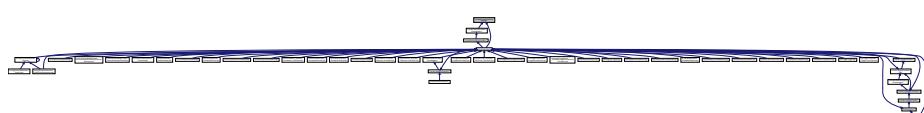
- enum `InferenceBoxType` {
   
 `INFERENCE_BOX_TYPE_RECTANGLE` = 0,
   
 `INFERENCE_BOX_TYPE_CIRCLE` = 1,
   
 `INFERENCE_BOX_TYPE_ROTATED_RECTANGLE` = 2
 }
   
*Inference Bounding Box Type.*

## 15.17 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`

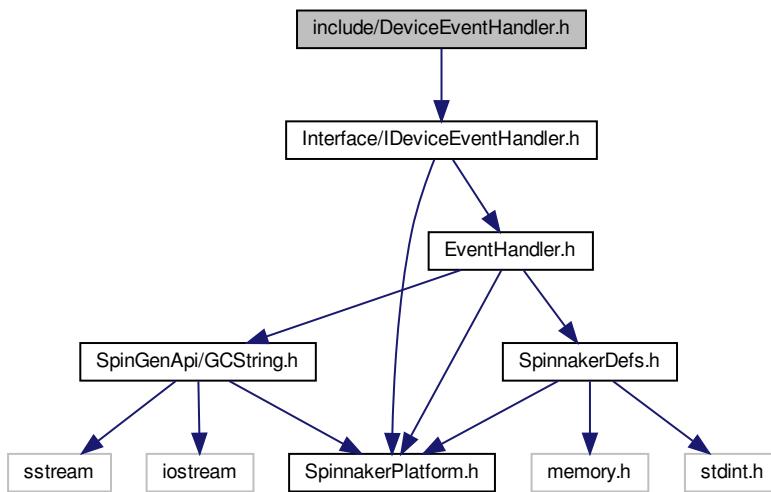
*An event handler for capturing the device arrival event.*

## Namespaces

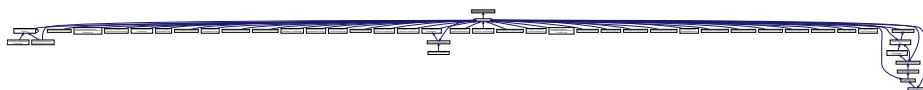
- [Spinnaker](#)

## 15.18 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](#)

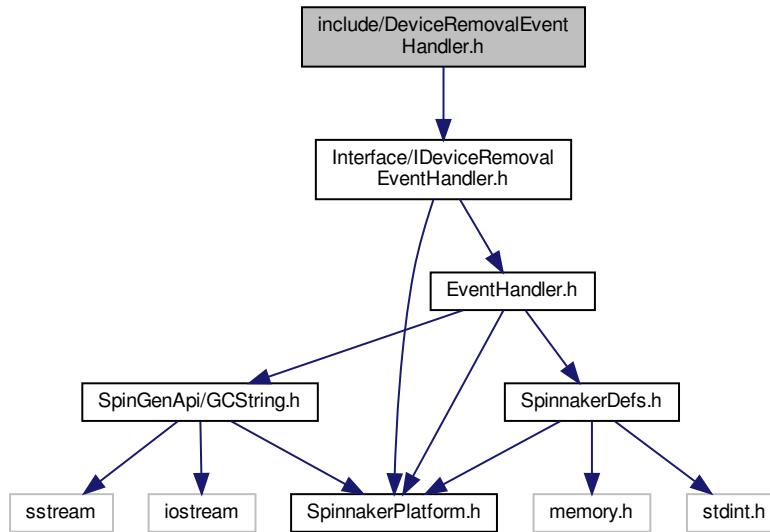
*A handler to device events.*

## Namespaces

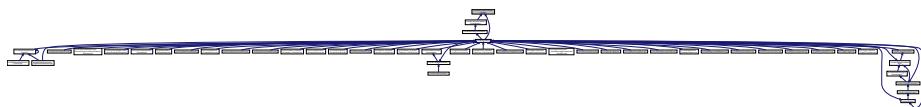
- [Spinnaker](#)

## 15.19 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](#)

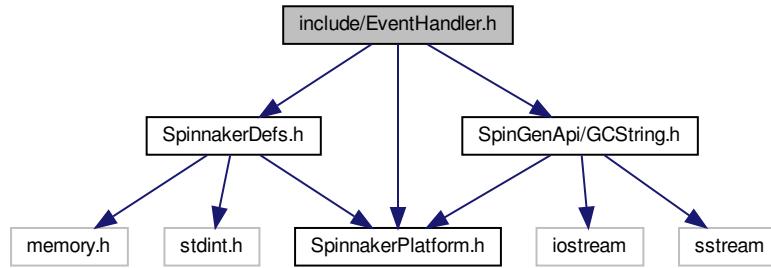
*An event handler for capturing the device removal event.*

### Namespaces

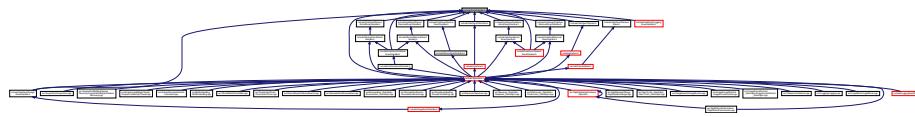
- [Spinnaker](#)

## 15.20 include/EventHandler.h File Reference

Include dependency graph for EventHandler.h:



This graph shows which files directly or indirectly include this file:



### Classes

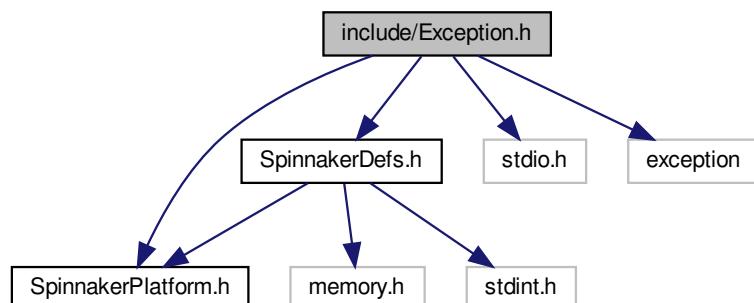
- class [EventHandler](#)  
*The base class for all event handler types.*

### Namespaces

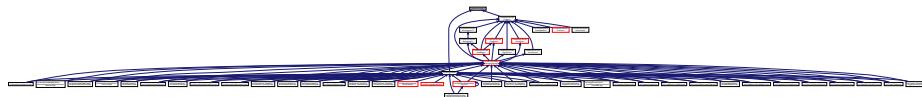
- [Spinnaker](#)

## 15.21 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](#)

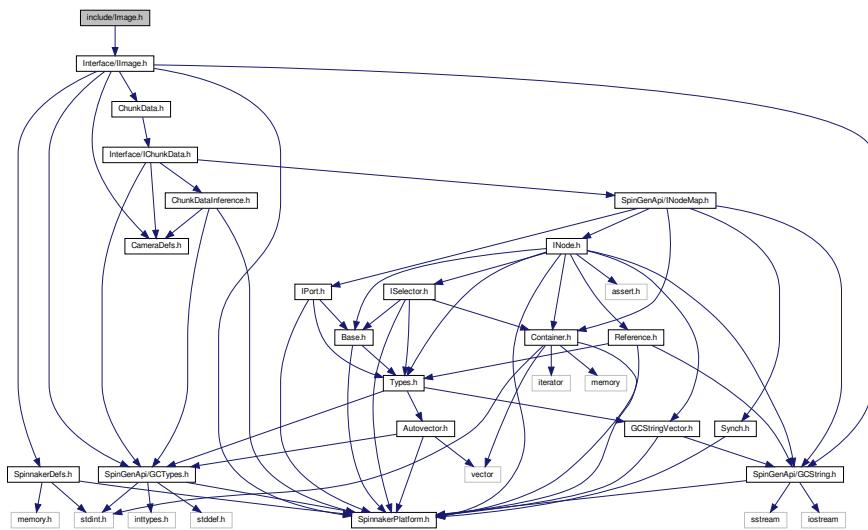
The [Exception](#) object represents an error that is returned from the library.

## Namespaces

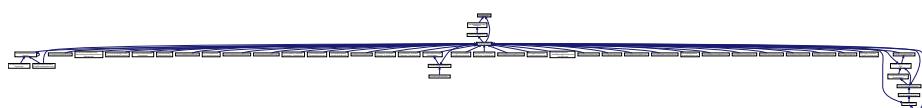
- [Spinnaker](#)

## 15.22 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](#)

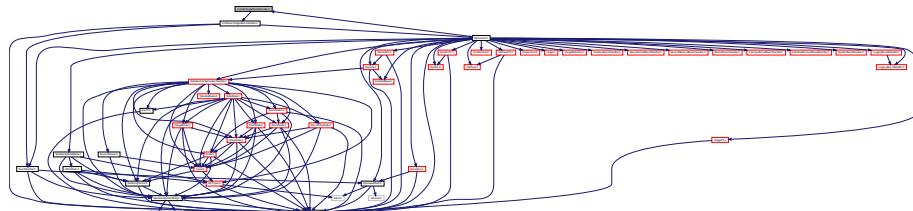
The [Image](#) object class.

## Namespaces

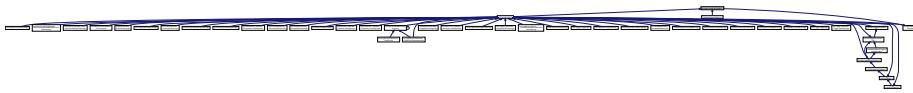
- Spinnaker

## 15.23 include/ImageEventHandler.h File Reference

Include dependency graph for ImageEventHandler.h:



This graph shows which files directly or indirectly include this file:



## Classes

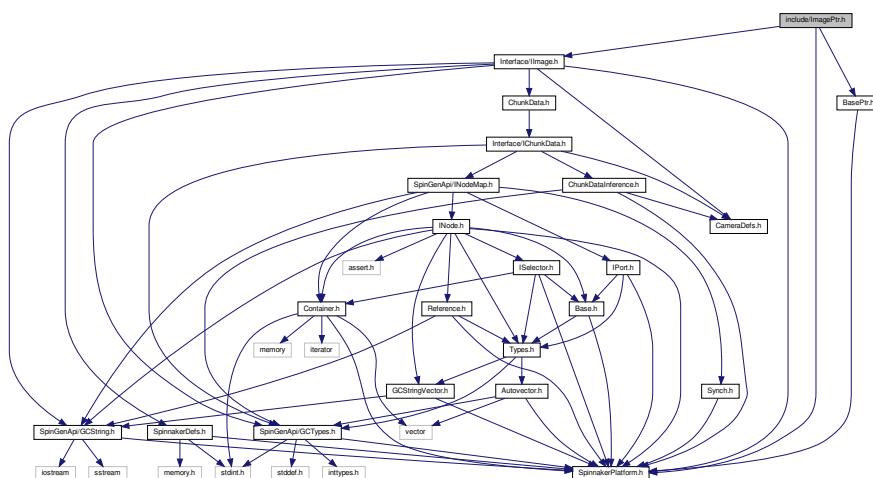
- class [ImageEventHandler](#)  
*A handler for capturing image arrival events.*

## Namespaces

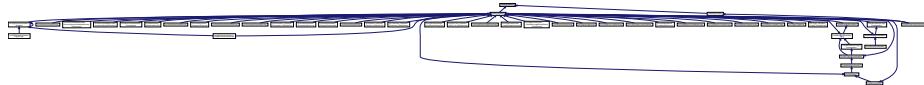
- Spinnaker

## 15.24 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](#)

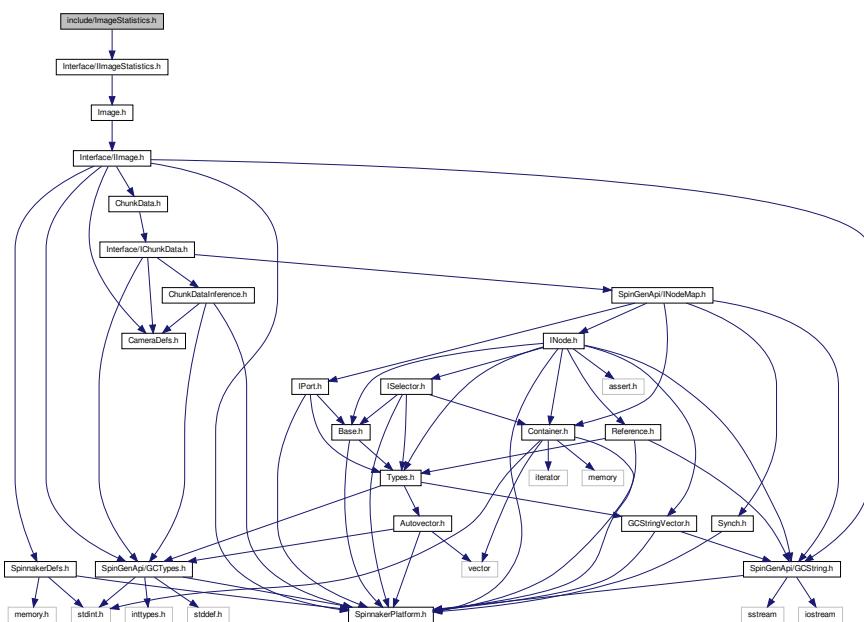
*A reference tracked pointer to an image object.*

## Namespaces

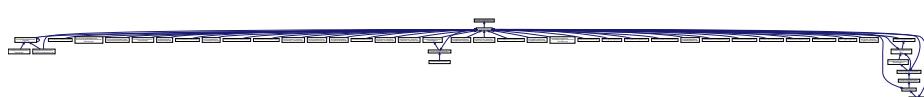
- [Spinnaker](#)

## 15.25 include/ImageStatistics.h File Reference

Include dependency graph for ImageStatistics.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ImageStatistics](#)

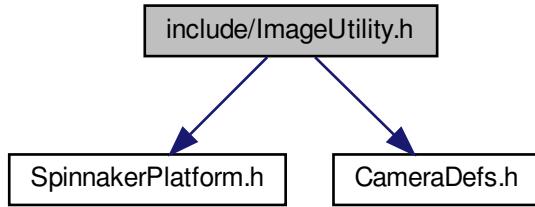
*Represents image statistics for an image.*

## Namespaces

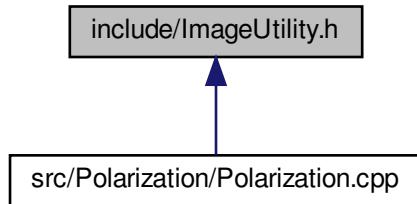
- [Spinnaker](#)

## 15.26 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](#)

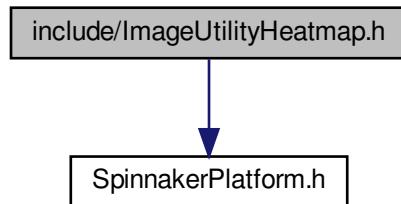
*Static helper functions for the image object class.*

## Namespaces

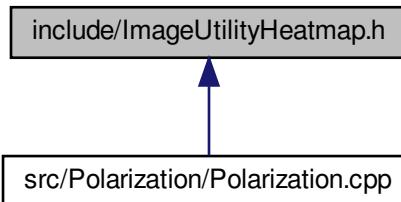
- [Spinnaker](#)

## 15.27 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](#)

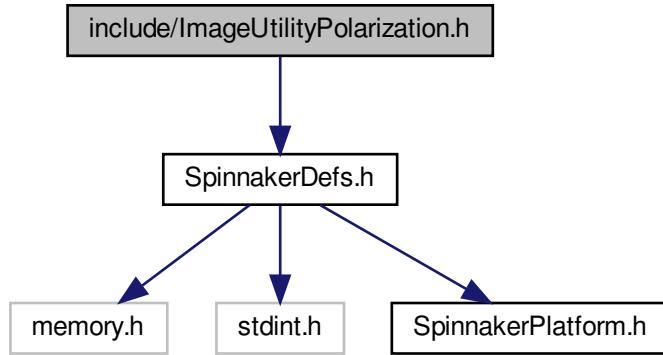
*Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.*

## Namespaces

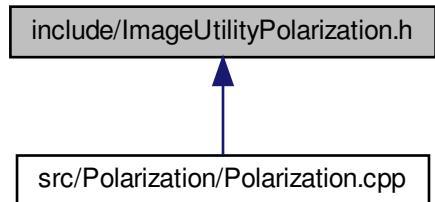
- [Spinnaker](#)

## 15.28 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](#)

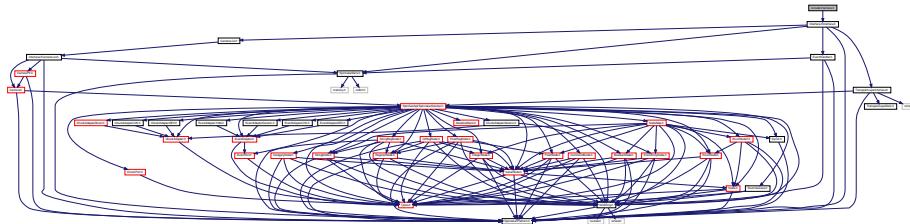
*Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.*

### Namespaces

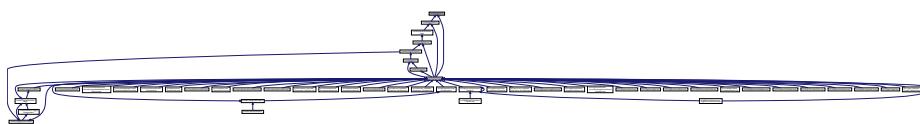
- [Spinnaker](#)

## 15.29 include/Interface.h File Reference

Include dependency graph for Interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

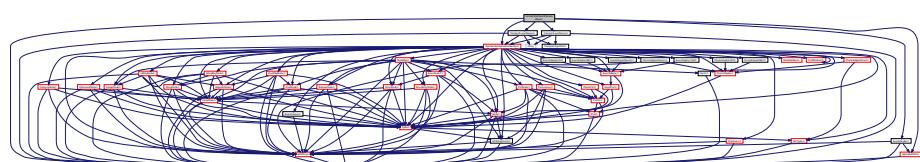
- class [Interface](#)  
*An interface object which holds a list of cameras.*

### Namespaces

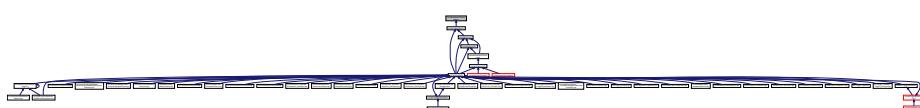
- [Spinnaker](#)

## 15.30 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](#)

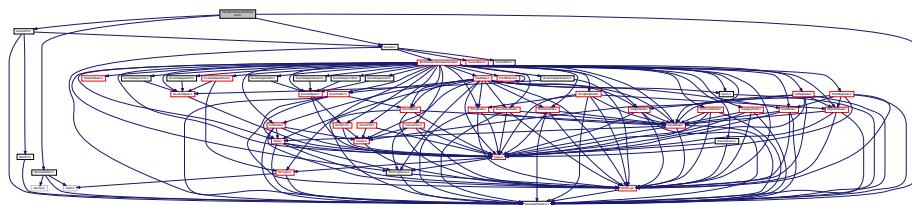
*The interface file for base class for the camera object.*

## Namespaces

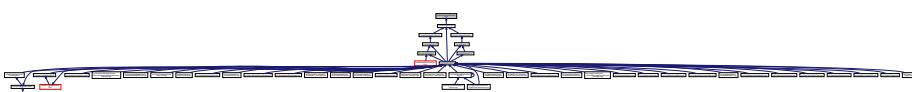
- [Spinnaker](#)

## 15.31 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](#)

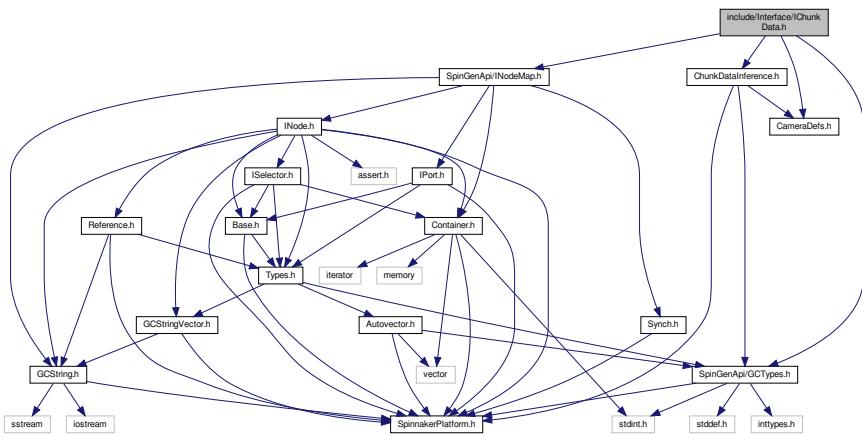
*Used to hold a list of camera objects.*

## Namespaces

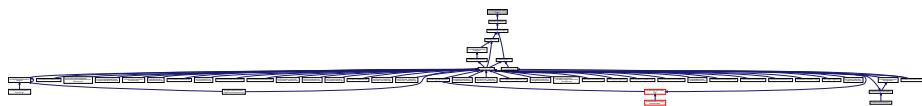
- [Spinnaker](#)

## 15.32 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](#)

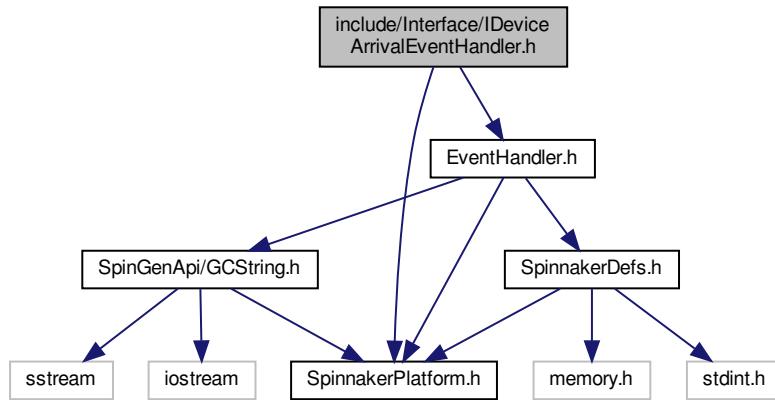
*The Interface file for [ChunkData](#).*

## Namespaces

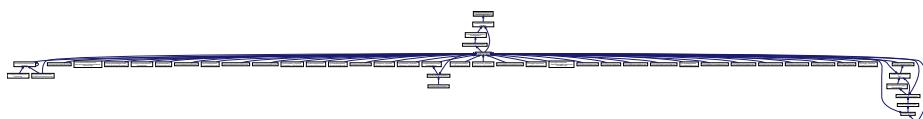
- [Spinnaker](#)

### 15.33 include/Interface/IDeviceArrivalEventHandler.h File Reference

Include dependency graph for IDeviceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



#### Classes

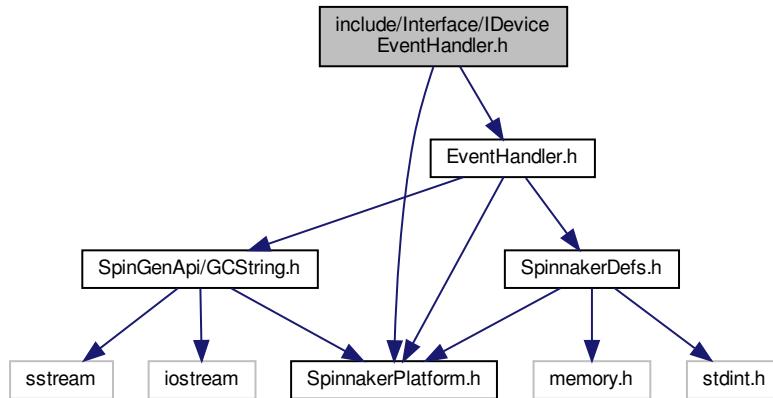
- class [IDeviceArrivalEventHandler](#)

#### Namespaces

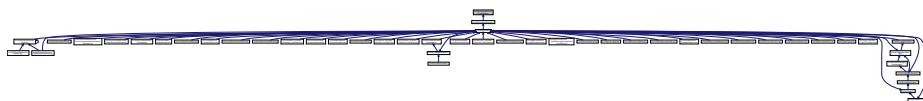
- [Spinnaker](#)

## 15.34 include/Interface/IDeviceEventHandler.h File Reference

Include dependency graph for IDeviceEventHandler.h:



This graph shows which files directly or indirectly include this file:



## Classes

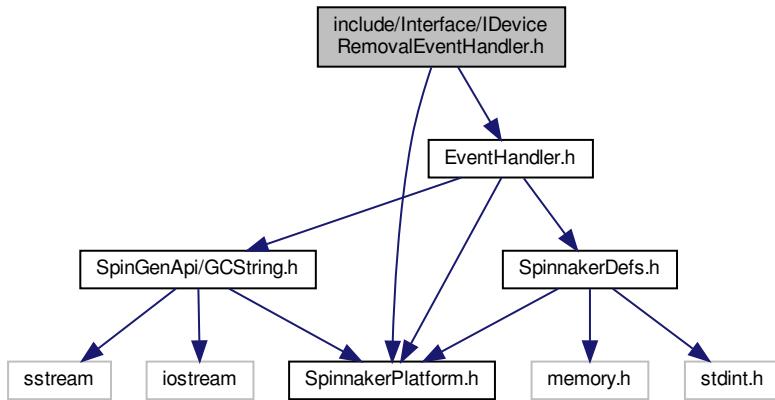
- class [IDeviceEventHandler](#)

## Namespaces

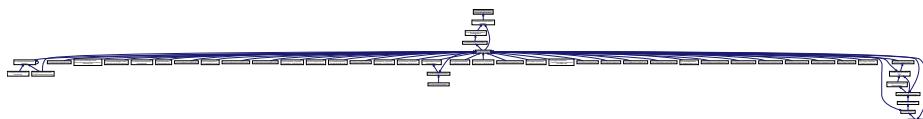
- [Spinnaker](#)

## 15.35 include/Interface/IDeviceRemovalEventHandler.h File Reference

Include dependency graph for IDeviceRemovalEventHandler.h:



This graph shows which files directly or indirectly include this file:



### Classes

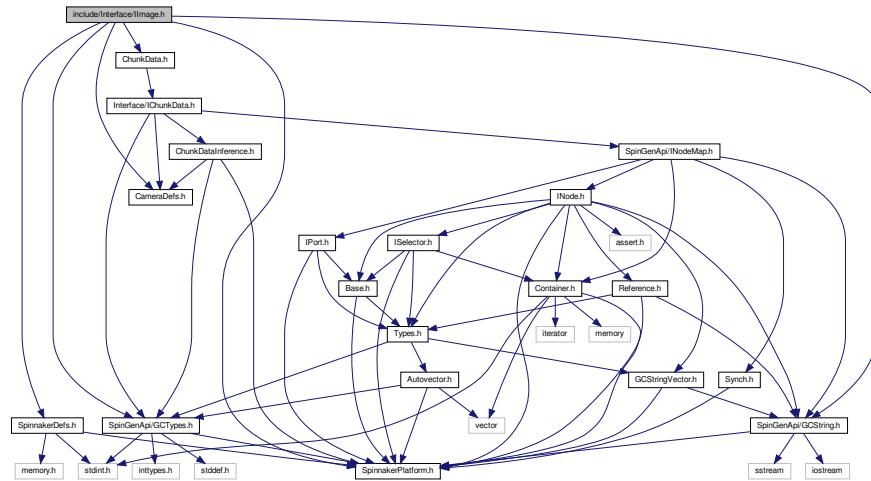
- class [IDeviceRemovalEventHandler](#)

### Namespaces

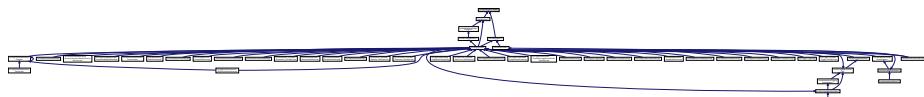
- [Spinnaker](#)

## 15.36 include/Interface/IImage.h File Reference

Include dependency graph for IImage.h:



This graph shows which files directly or indirectly include this file:



## Classes

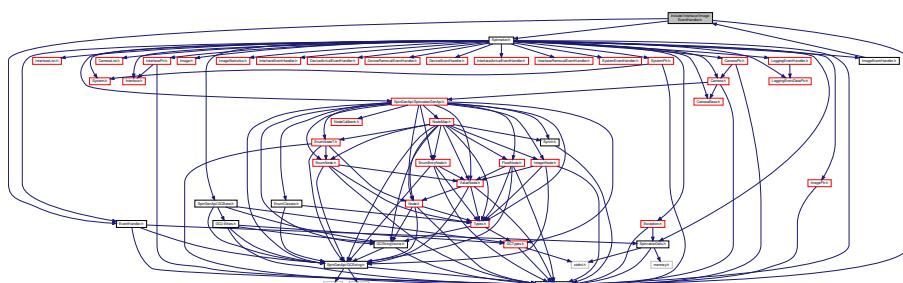
- class `IImage`  
*The interface file for `Image`.*

## Namespaces

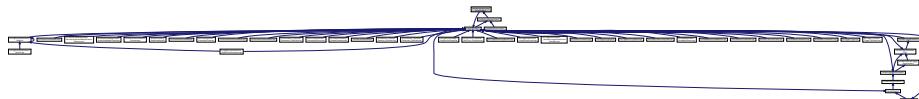
- Spinnaker

## 15.37 include/Interface/IImageEventHandler.h File Reference

Include dependency graph for `ILImageEventHandler.h`:



This graph shows which files directly or indirectly include this file:



## Classes

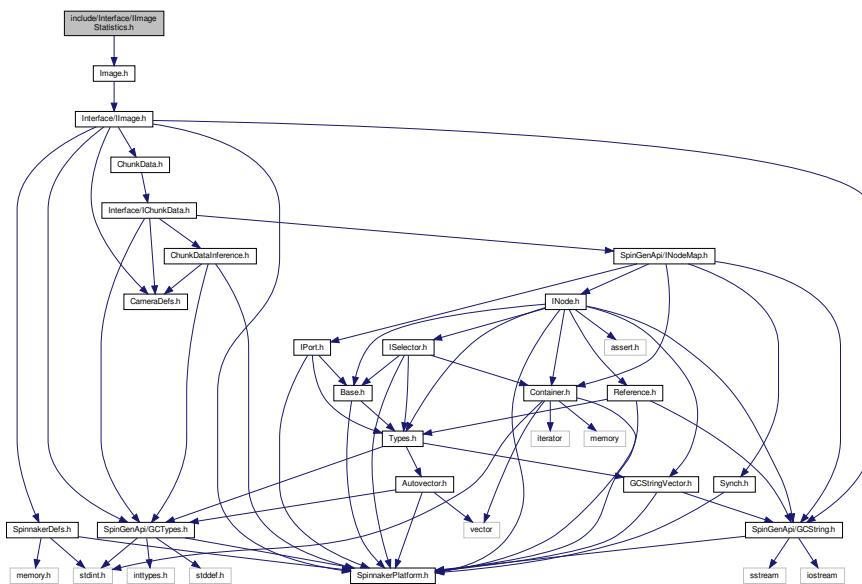
- class [IImageEventHandler](#)

## Namespaces

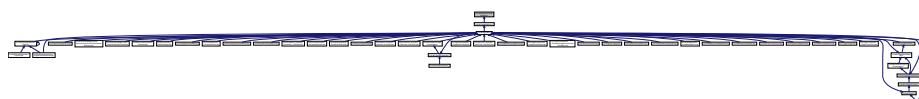
- [Spinnaker](#)

## 15.38 include/Interface/IImageStatistics.h File Reference

Include dependency graph for `IImageStatistics.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [IImageStatistics](#)

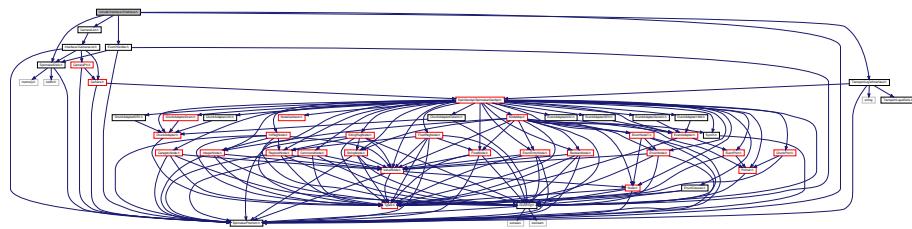
*The interface file for image statistics.*

## Namespaces

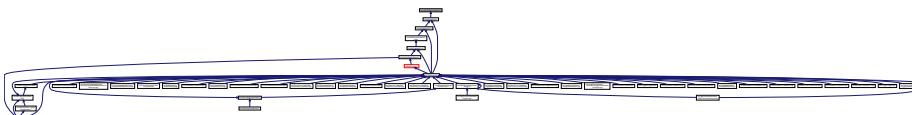
- Spinnaker

## 15.39 include/Interface/IInterface.h File Reference

Include dependency graph for IInterface.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class IInterface

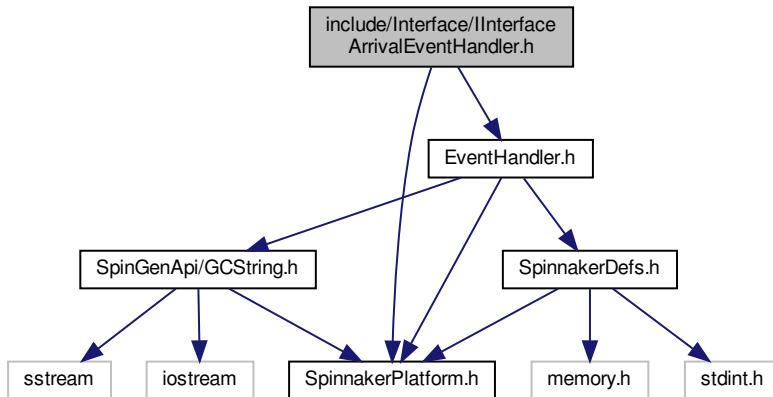
*The interface file for Interface.*

## Namespaces

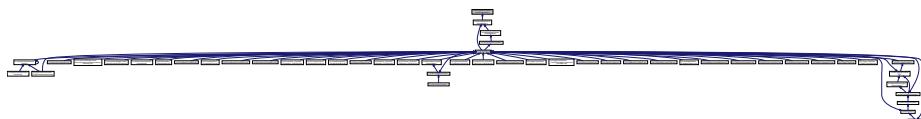
- Spinnaker

## 15.40 include/Interface/IInterfaceArrivalEventHandler.h File Reference

Include dependency graph for IInterfaceArrivalEventHandler.h:



This graph shows which files directly or indirectly include this file:



### Classes

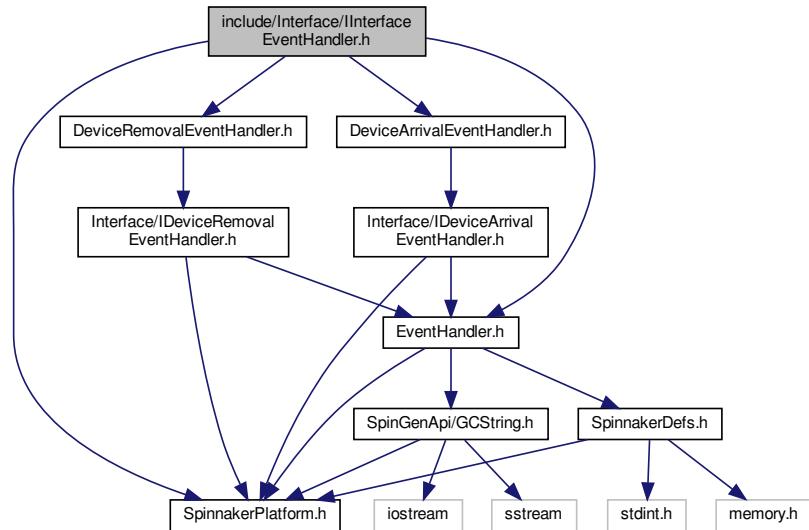
- class [IInterfaceArrivalEventHandler](#)

### Namespaces

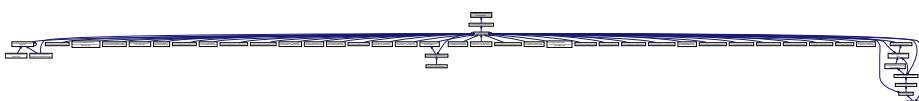
- [Spinnaker](#)

## 15.41 include/Interface/IInterfaceEventHandler.h File Reference

Include dependency graph for IInterfaceEventHandler.h:



This graph shows which files directly or indirectly include this file:



### Classes

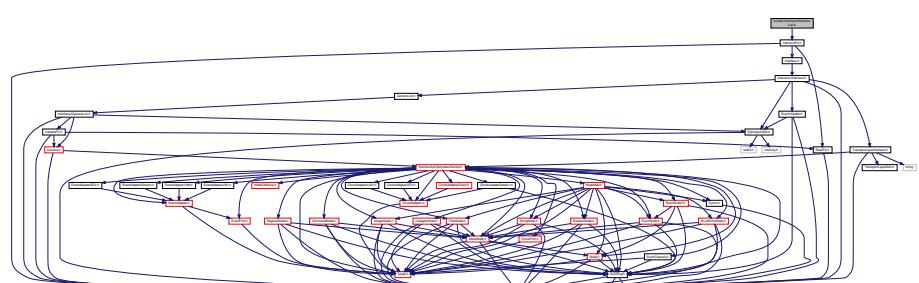
- class `IInterfaceEventHandler`

### Namespaces

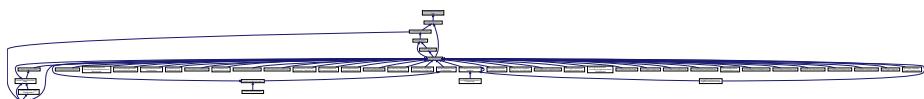
- `Spinnaker`

## 15.42 include/Interface/IInterfaceList.h File Reference

Include dependency graph for IInterfaceList.h:



This graph shows which files directly or indirectly include this file:



## Classes

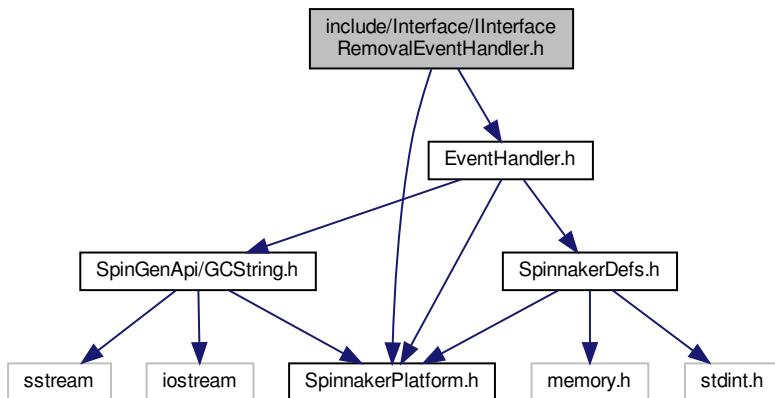
- class [IInterfaceList](#)  
*The interface file for `IInterfaceList` class.*

## Namespaces

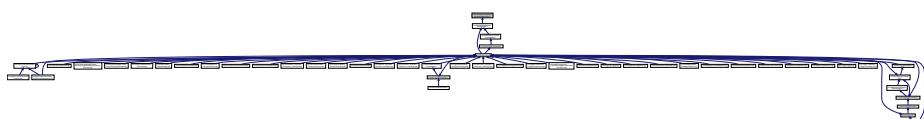
- [Spinnaker](#)

## 15.43 include/Interface/IInterfaceRemovalEventHandler.h File Reference

Include dependency graph for `IInterfaceRemovalEventHandler.h`:



This graph shows which files directly or indirectly include this file:



## Classes

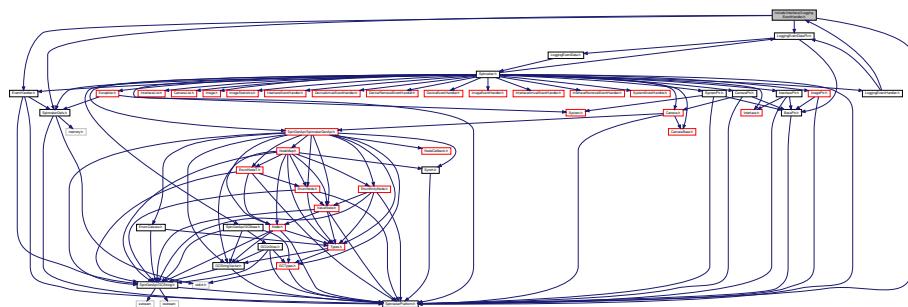
- class [IInterfaceRemovalEventHandler](#)

## Namespaces

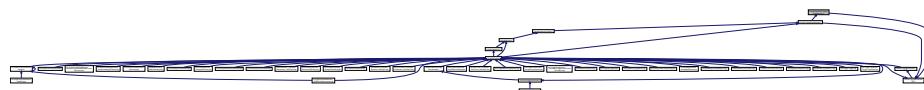
- Spinnaker

## 15.44 include/Interface/ILoggingEventHandler.h File Reference

Include dependency graph for ILoggingEventHandler.h:



This graph shows which files directly or indirectly include this file:



## Classes

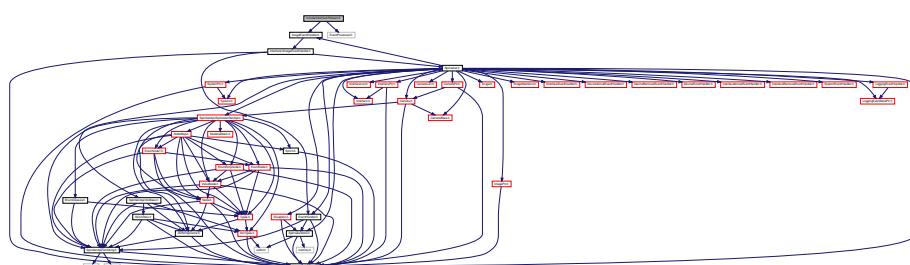
- class [ILoggingEventHandler](#)

## Namespaces

- Spinnaker

## 15.45 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:



## Classes

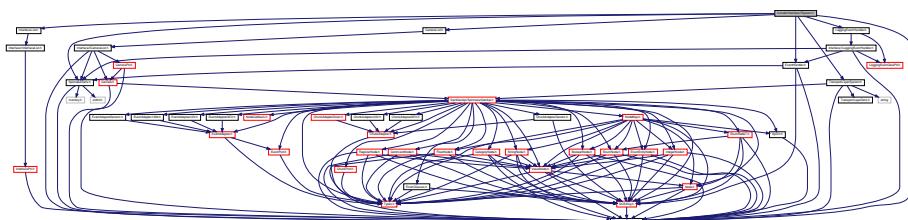
- class [IDataStream](#)

## Namespaces

- [Spinnaker](#)

## 15.46 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](#)

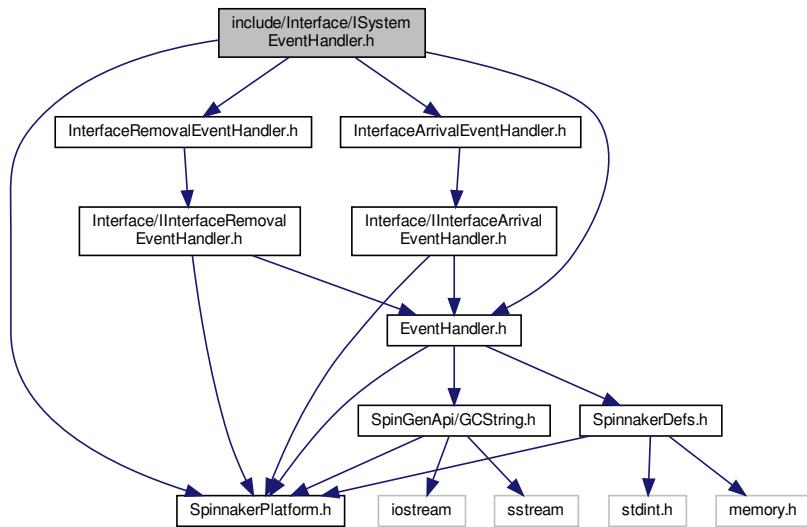
*The interface file for [System](#).*

## Namespaces

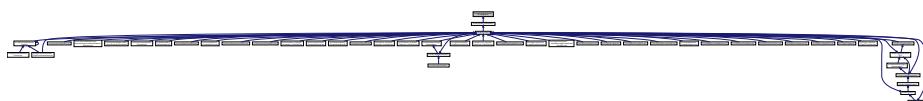
- [Spinnaker](#)

## 15.47 include/Interface/ISystemEventHandler.h File Reference

Include dependency graph for ISystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



### Classes

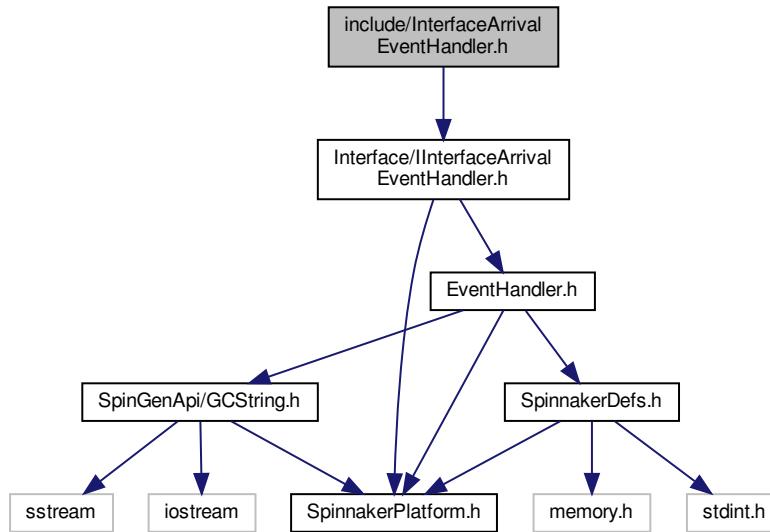
- class [ISystemEventHandler](#)

### Namespaces

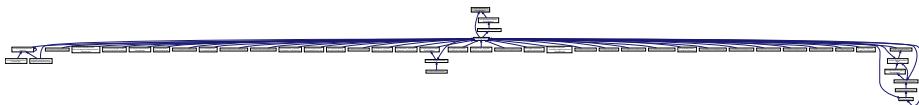
- [Spinnaker](#)

## 15.48 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](#)

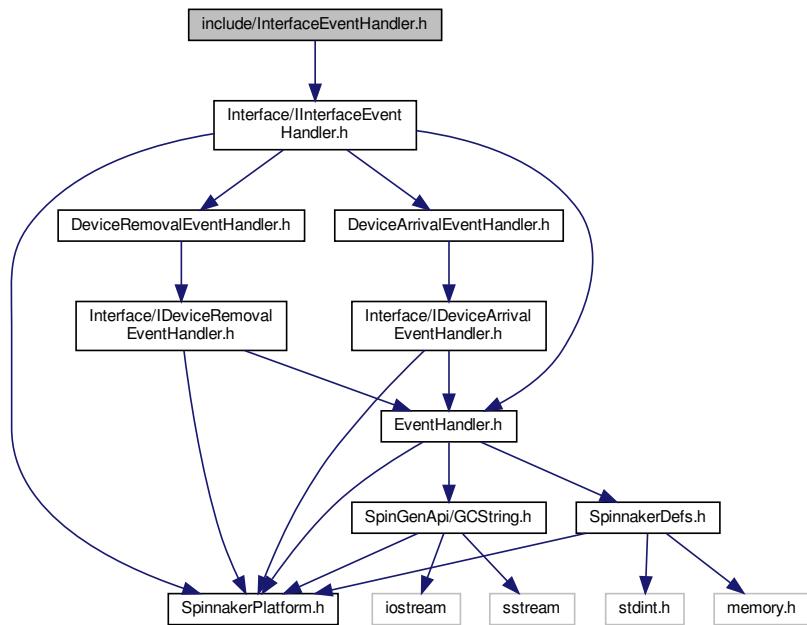
*An event handler for capturing the interface arrival event.*

### Namespaces

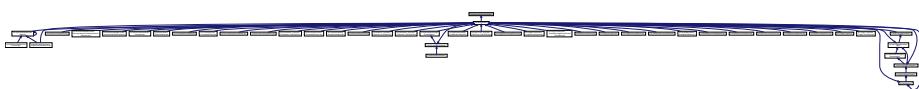
- [Spinnaker](#)

## 15.49 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](#)

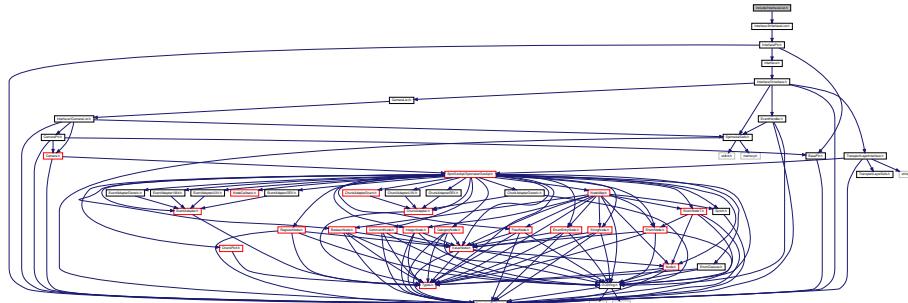
*A handler to device arrival and removal events on all interfaces.*

## Namespaces

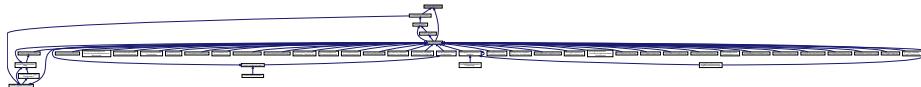
- [Spinnaker](#)

## 15.50 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](#)

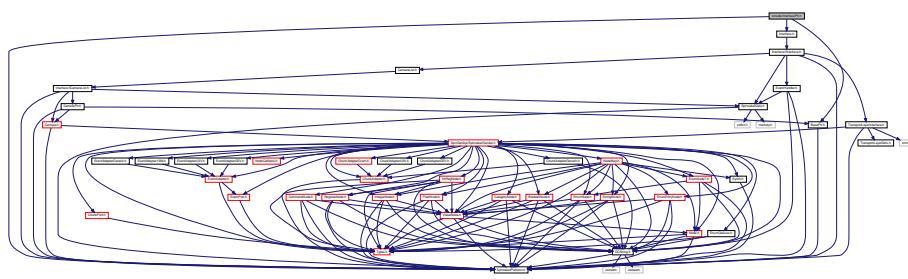
*A list of the available interfaces on the system.*

### Namespaces

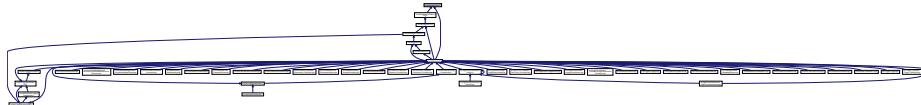
- [Spinnaker](#)

## 15.51 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](#)

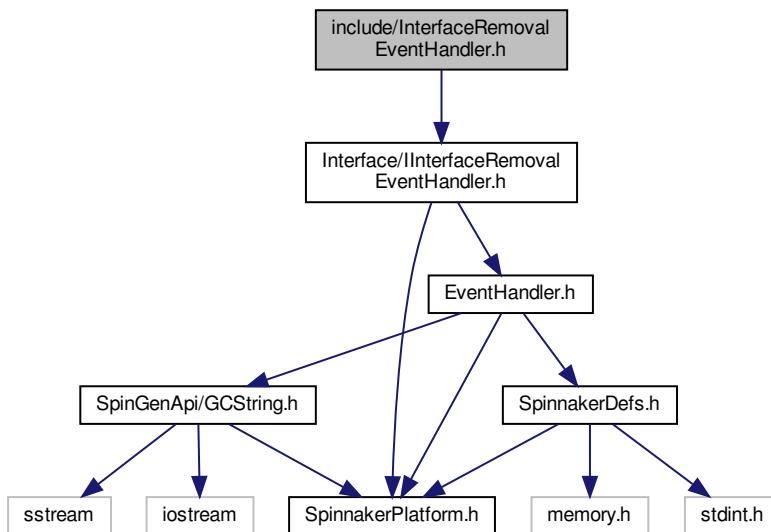
*A reference tracked pointer to the interface object.*

## Namespaces

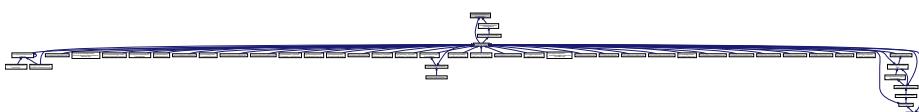
- [Spinnaker](#)

## 15.52 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](#)

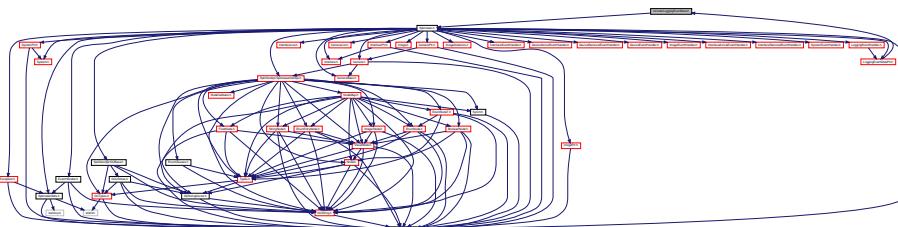
*An event handler for capturing the interface removal event.*

## Namespaces

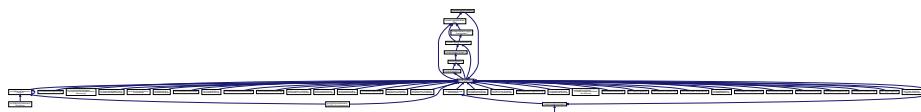
- [Spinnaker](#)

## 15.53 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:



This graph shows which files directly or indirectly include this file:



### Classes

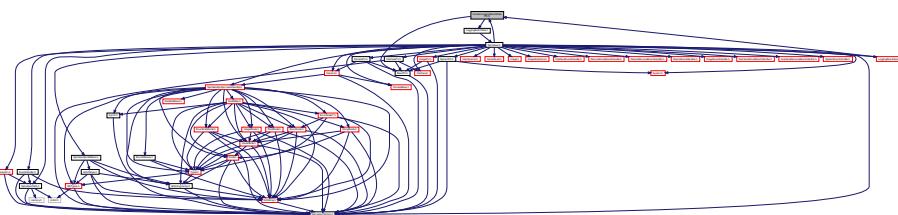
- class [LoggingEventData](#)  
*The LoggingEventData object.*

### Namespaces

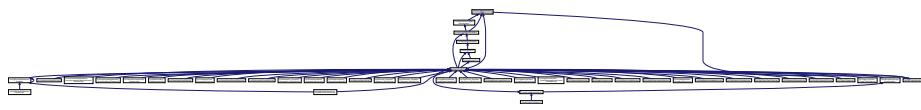
- [Spinnaker](#)

## 15.54 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](#)

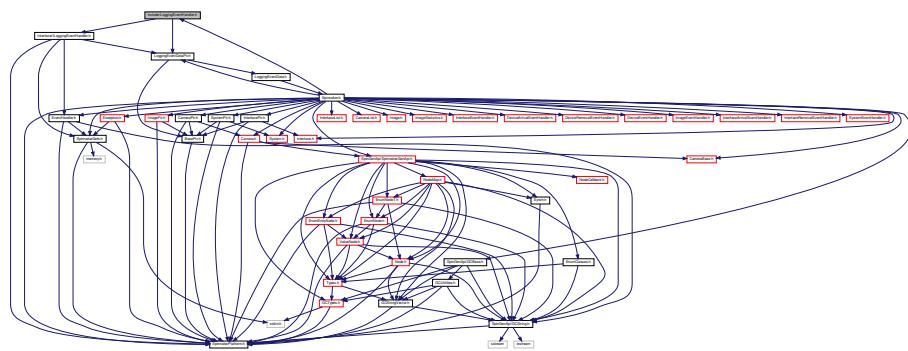
*A reference tracked pointer to the LoggingEvent object.*

## Namespaces

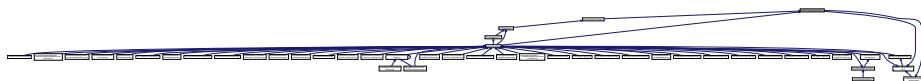
- [Spinnaker](#)

## 15.55 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](#)

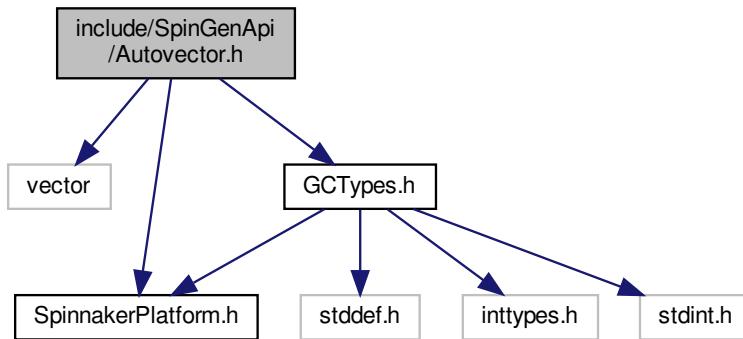
*An event handler for capturing the device logging event.*

## Namespaces

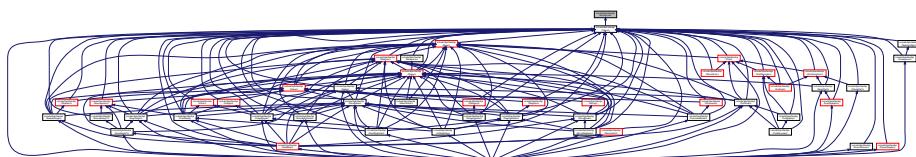
- [Spinnaker](#)

## 15.56 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



## Classes

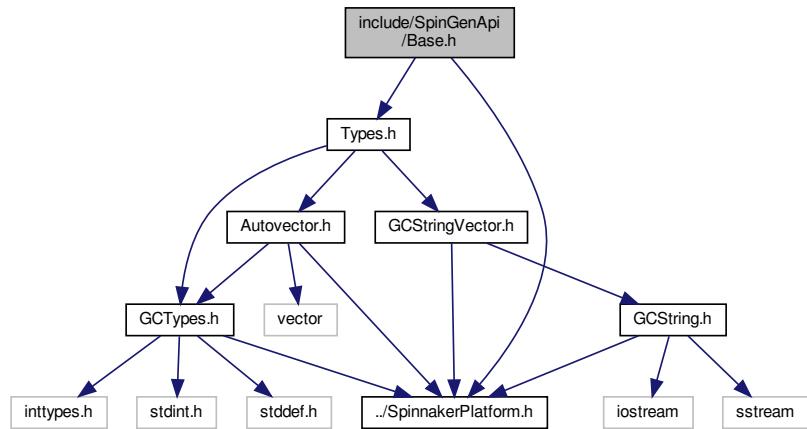
- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*

## Namespaces

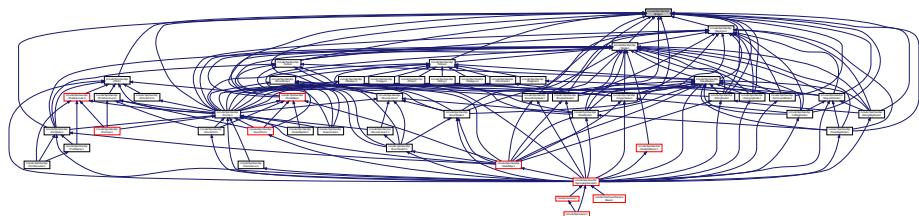
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.57 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

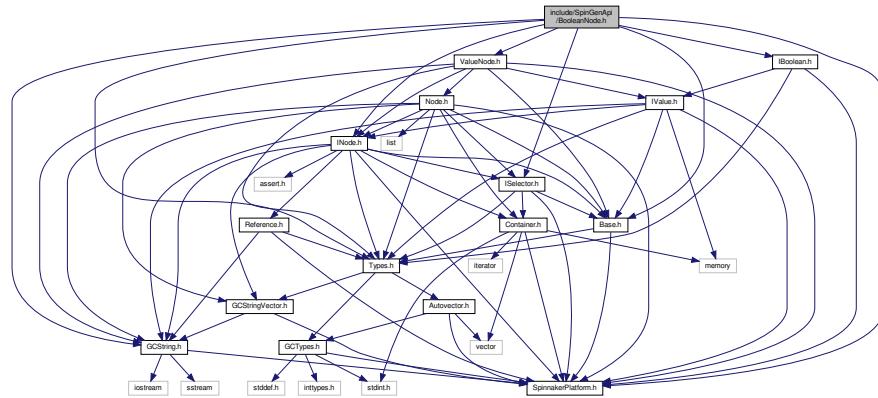
### Variables

- interface SPINNAKER\_API\_ABSTRACT IBase

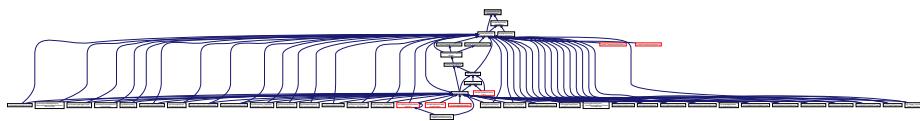
*Base interface common to all nodes.*

## 15.58 include/SpinGenApi/BooleanNode.h File Reference

Include dependency graph for BooleanNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [BooleanNode](#)  
*Interface for string properties.*

## Namespaces

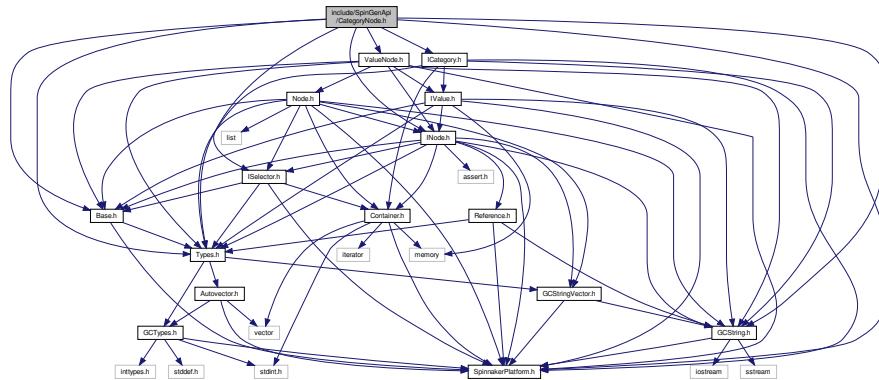
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

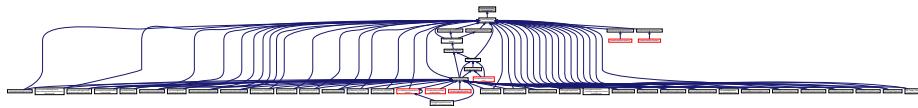
- typedef BooleanNode [CBooleanRef](#)

## 15.59 include/SpinGenApi/CategoryNode.h File Reference

Include dependency graph for CategoryNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CategoryNode](#)  
*Interface for string properties.*

## Namespaces

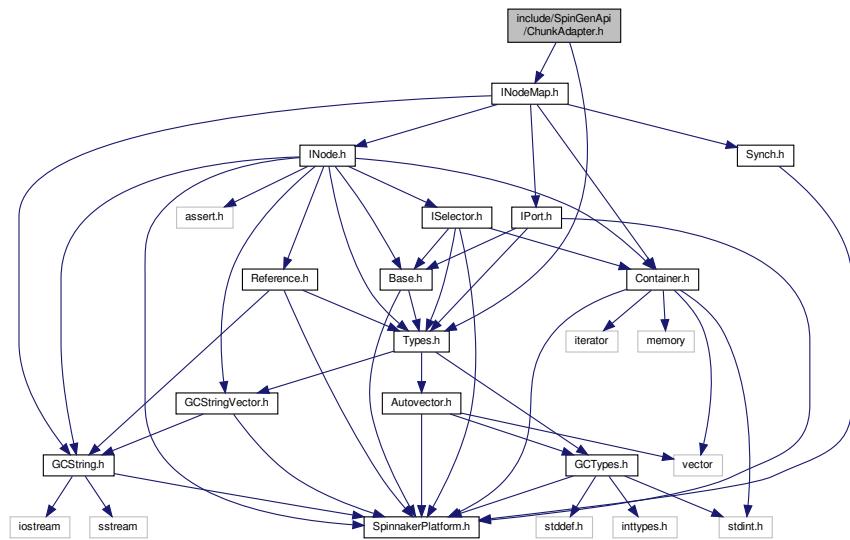
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

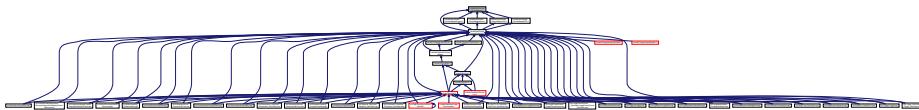
- typedef CategoryNode [CCategoryRef](#)

## 15.60 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



## Classes

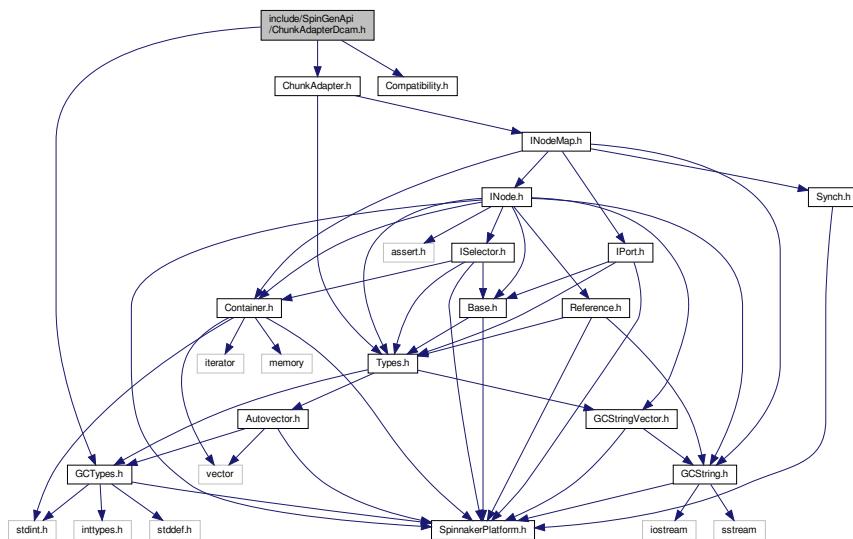
- struct [AttachStatistics\\_t](#)  
*Delivers information about the attached chunks and nodes.*
- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*

## Namespaces

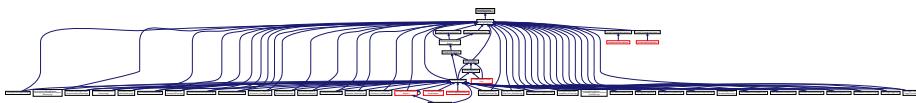
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.61 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

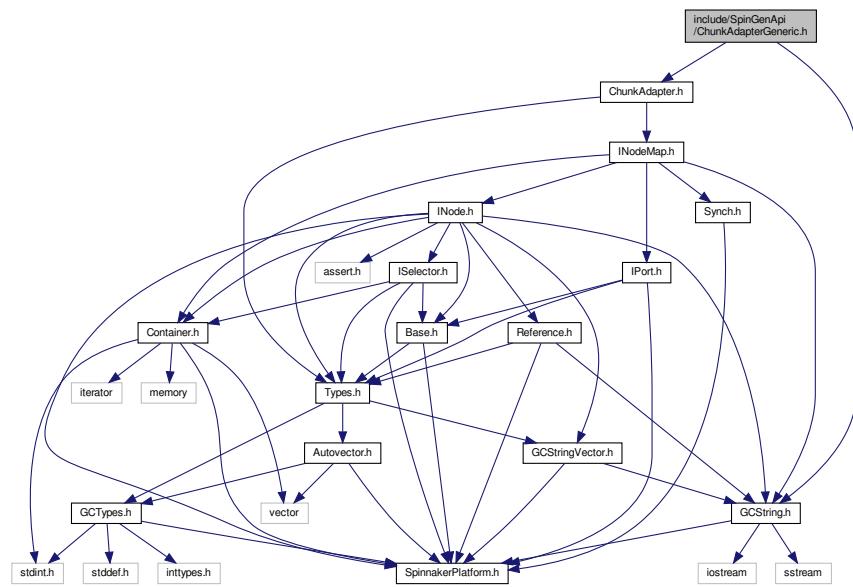
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

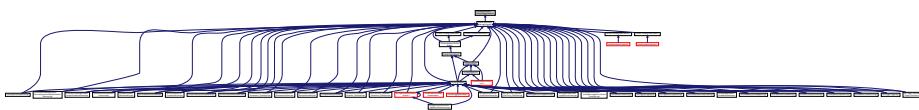
- void [SPINNAKER\\_API\\_SET\\_GUID](#) (SPIN\_GUID &name, uint32\_t l, uint16\_t w1, uint16\_t w2, uint8\_t b1, uint8\_t b2, uint8\_t b3, uint8\_t b4, uint8\_t b5, uint8\_t b6, uint8\_t b7, uint8\_t b8)

## 15.62 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



## Classes

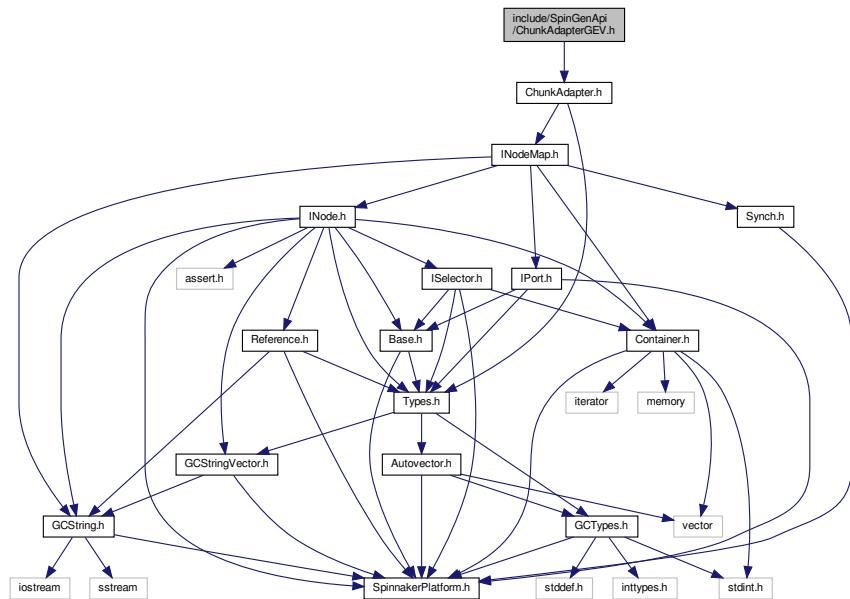
- struct [SingleChunkData\\_t](#)
  - struct [SingleChunkDataStr\\_t](#)
  - class [CChunkAdapterGeneric](#)

## Namespaces

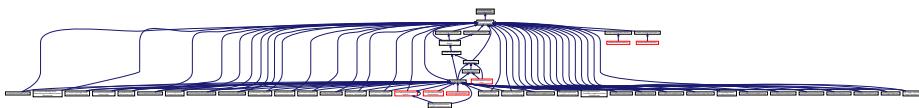
- Spinnaker
  - Spinnaker::GenApi

## 15.63 include/SpinGenApi/ChunkAdapterGEV.h File Reference

Include dependency graph for ChunkAdapterGEV.h:



This graph shows which files directly or indirectly include this file:



## Classes

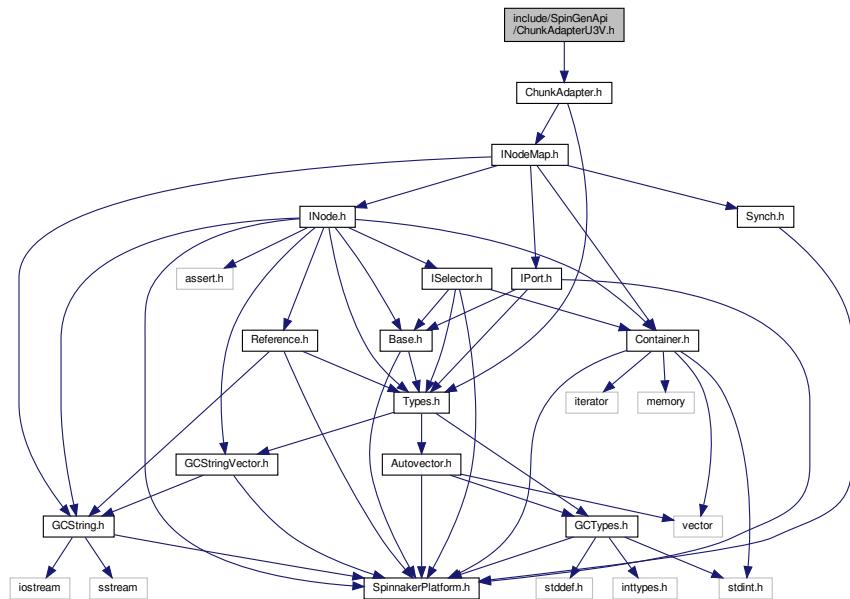
- struct [GVCP\\_CHUNK\\_TRAILER](#)  
*header of a GVCP request packet*
- class [CChunkAdapterGEV](#)  
*Connects a chunked DCAM buffer to a node map.*

## Namespaces

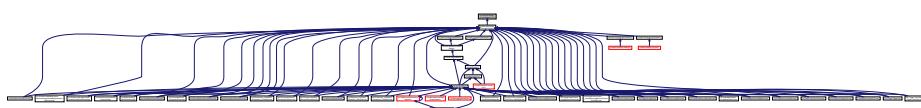
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.64 include/SpinGenApi/ChunkAdapterU3V.h File Reference

Include dependency graph for ChunkAdapterU3V.h:



This graph shows which files directly or indirectly include this file:



## Classes

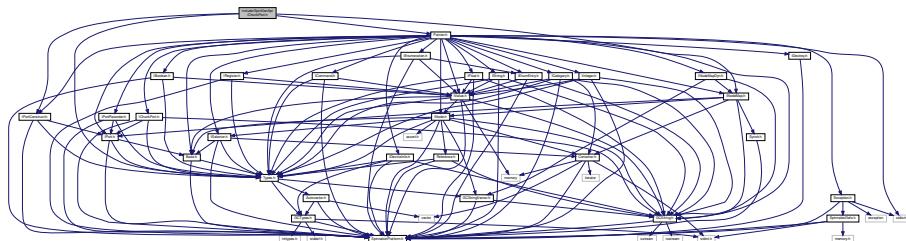
- struct [U3V\\_CHUNK\\_TRAILER](#)  
*header of a GVCP request packet*
- class [CChunkAdapterU3V](#)  
*Connects a chunked U3V buffer to a node map.*

## Namespaces

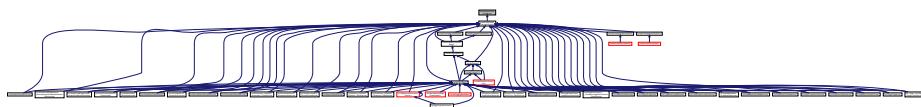
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.65 include/SpinGenApi/ChunkPort.h File Reference

Include dependency graph for ChunkPort.h:



This graph shows which files directly or indirectly include this file:



### Classes

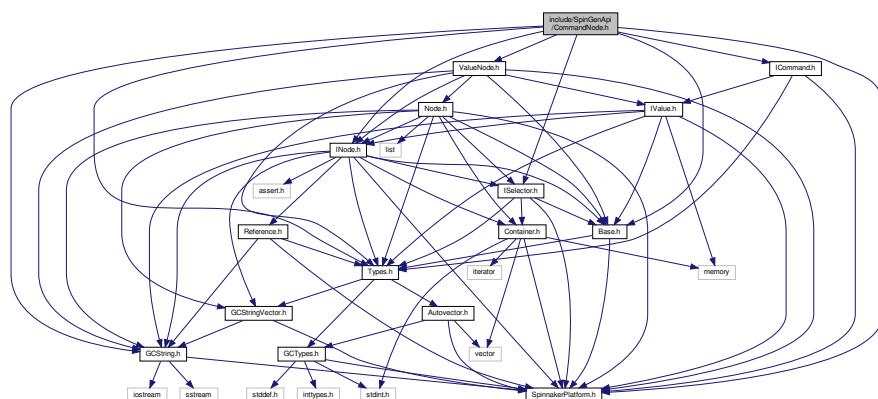
- class [CChunkPort](#)  
*Port attachable to a chunk in a buffer.*

### Namespaces

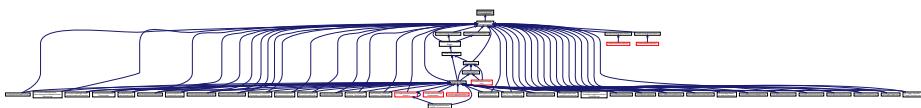
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.66 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

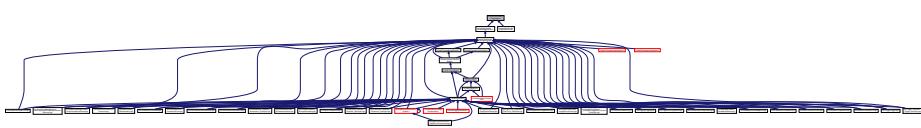
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

- typedef CommandNode [CCommandRef](#)

## 15.67 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



## Macros

- `#define FMT_I64 "ll"`

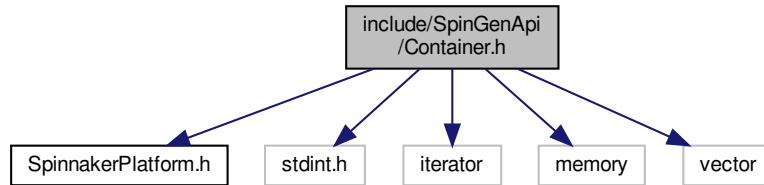
### 15.67.1 Macro Definition Documentation

#### 15.67.1.1 FMT\_I64

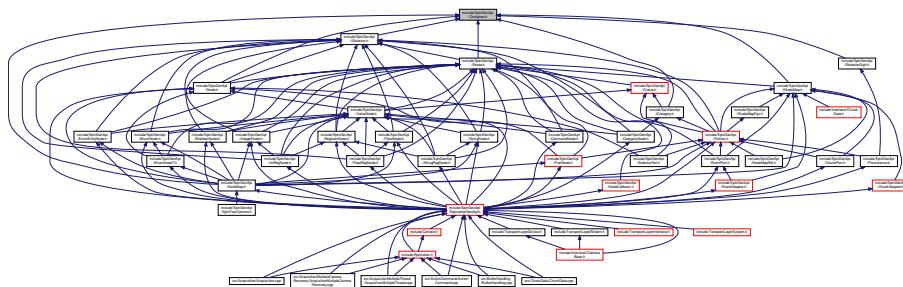
```
#define FMT_I64 "ll"
```

## 15.68 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:



This graph shows which files directly or indirectly include this file:



## 15.69 include/SpinGenApi/Counter.h File Reference

### Classes

- class [Counter](#)

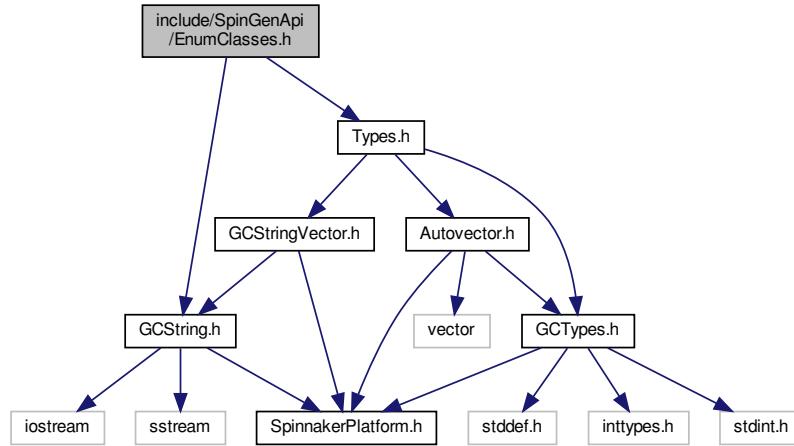
*Definition of a simple [Counter](#) class.*

### Namespaces

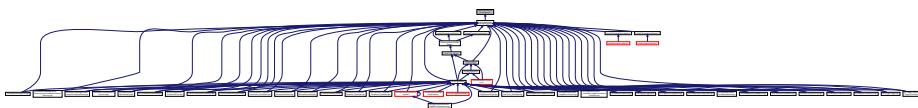
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.70 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 [EEndianessClass](#)  
*Holds conversion methods for the endianess 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](#)

*Holds conversion methods for the converter formulas.*

- class [EDisplayNotationClass](#)

*Holds conversion methods for the notation type of floats.*

- class [EInputDirectionClass](#)

*Holds conversion methods for the notation type of floats.*

- class [EGenApiSchemaVersionClass](#)

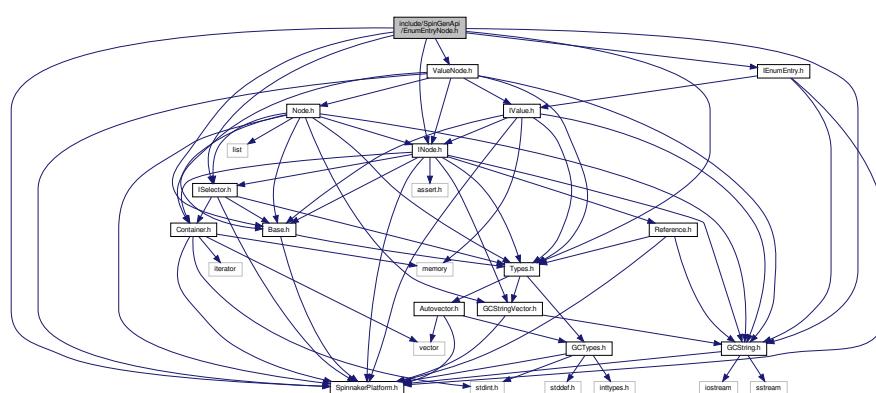
*helper class converting EGenApiSchemaVersion from and to string*

## Namespaces

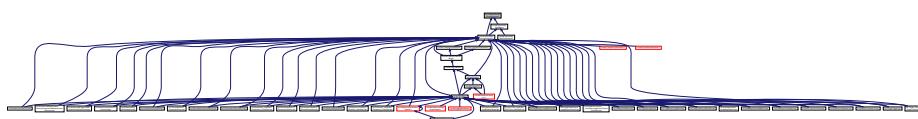
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.71 include/SpinGenApi/EnumEntryNode.h File Reference

Include dependency graph for EnumEntryNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [EnumEntryNode](#)

*Interface for string properties.*

## Namespaces

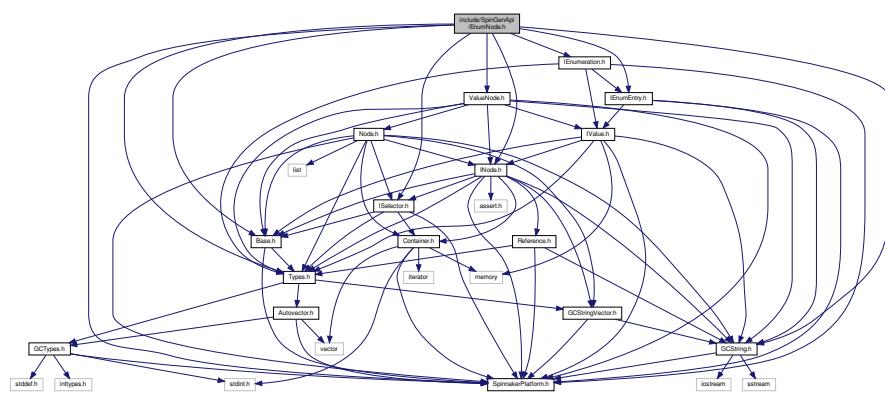
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

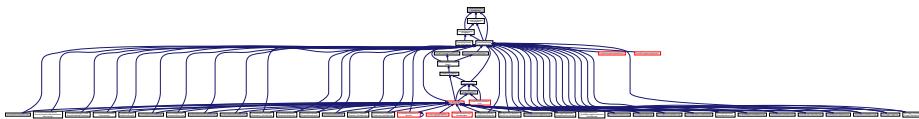
- `typedef EnumEntryNode CEnumEntryRef`

## 15.72 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

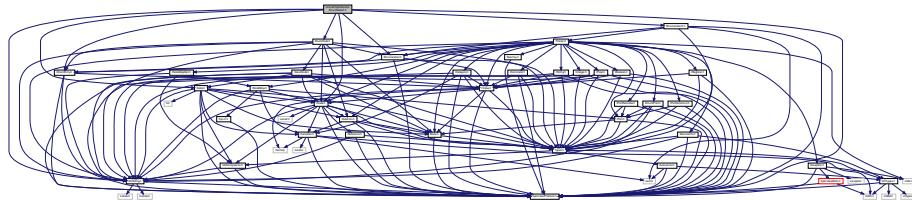
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

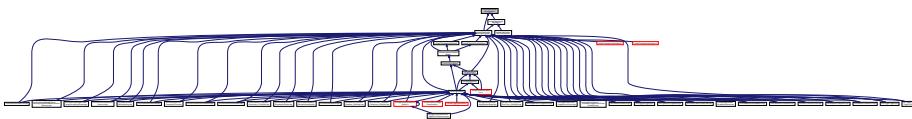
- `typedef EnumNode CEnumerationRef`

## 15.73 include/SpinGenApi/EnumNodeT.h File Reference

Include dependency graph for EnumNodeT.h:



This graph shows which files directly or indirectly include this file:



### Classes

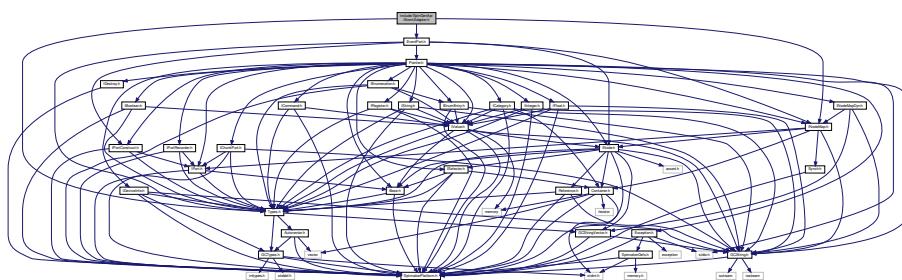
- class [CEnumerationTRef< EnumT >](#)  
*Interface for string properties.*

### Namespaces

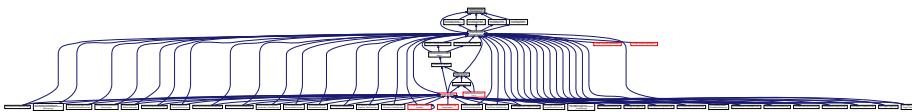
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.74 include/SpinGenApi/EventAdapter.h File Reference

Include dependency graph for EventAdapter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CEventAdapter](#)

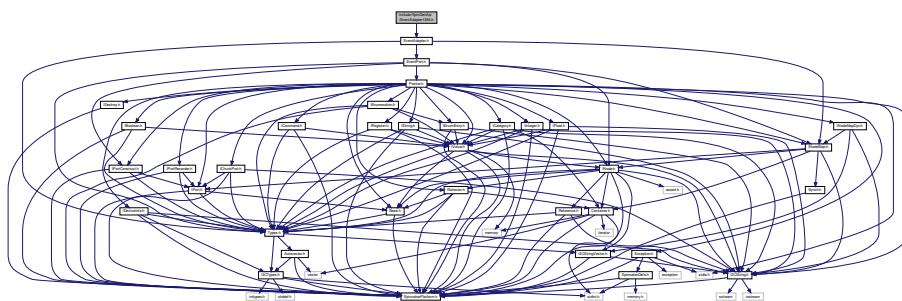
*Delivers Events to ports.*

## Namespaces

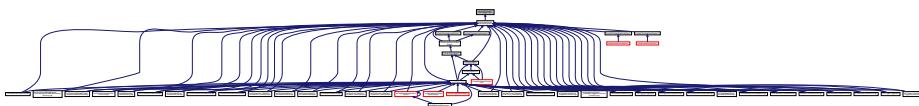
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.75 include/SpinGenApi/EventAdapter1394.h File Reference

Include dependency graph for EventAdapter1394.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [CEventAdapter1394](#)

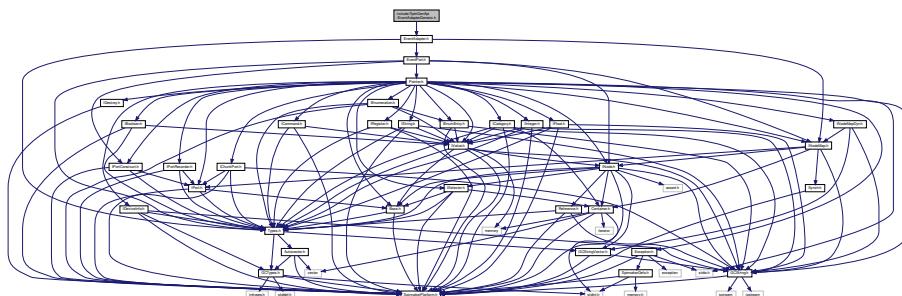
*Distribute the events to the node map.*

## Namespaces

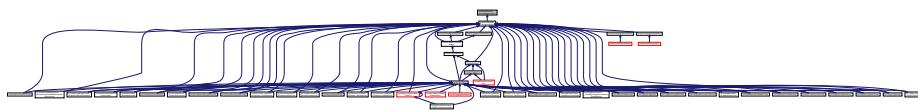
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.76 include/SpinGenApi/EventAdapterGeneric.h File Reference

Include dependency graph for EventAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



### Classes

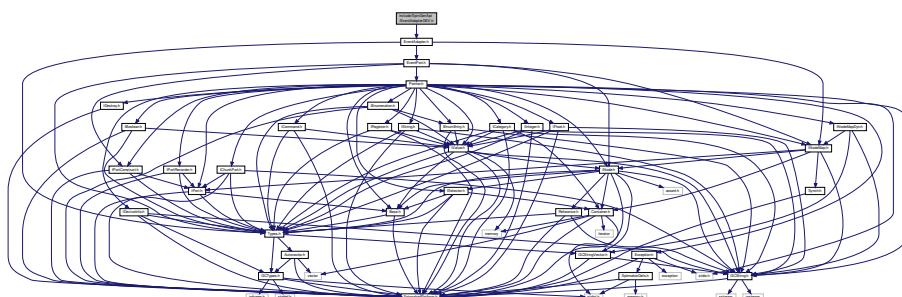
- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*

### Namespaces

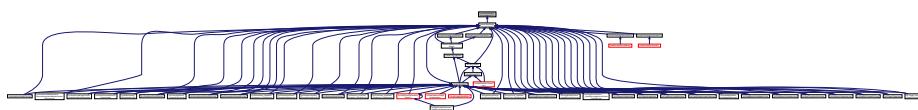
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.77 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

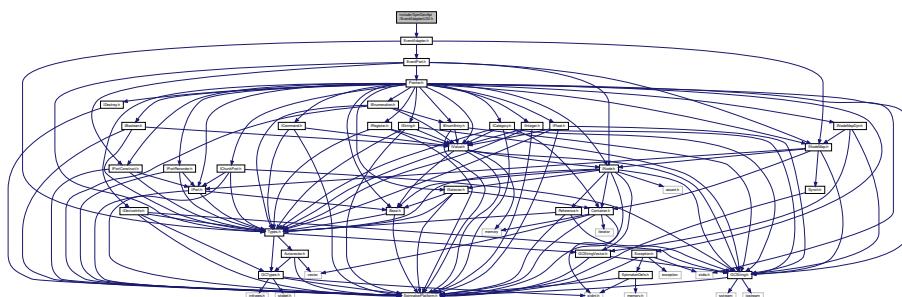
- enum [GVCP\\_MESSAGE\\_TAGS](#) {
   
*TAG\_EVENT\_CMD = 0xc0,*
  
*TAG\_EVENTDATA\_CMD = 0xc2 }*

## Variables

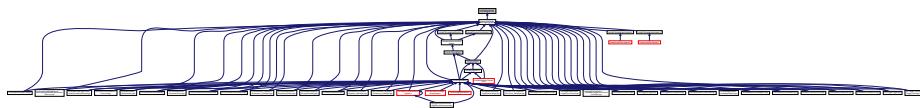
- const uint8\_t [COMMAND\\_MAGIC](#) = 0x42

## 15.78 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

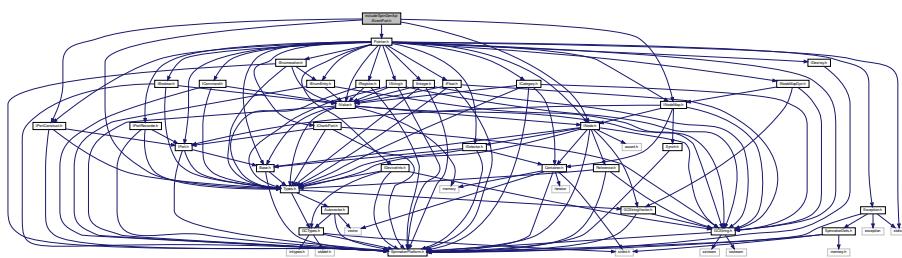
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Variables

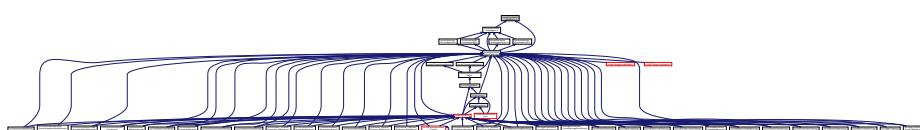
- const uint32\_t [U3V\\_EVENT\\_PREFIX](#) = 0x45563355
- const uint16\_t [GENCP\\_EVENT\\_CMD\\_ID](#) = 0x0C00
- const size\_t [GENCP\\_COMMAND\\_HEADER\\_SIZE](#) = sizeof(U3V\_COMMAND\_HEADER)
- const size\_t [GENCP\\_EVENT\\_BASIC\\_SIZE](#) = sizeof(U3V\_EVENT\_MESSAGE)

## 15.79 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](#)

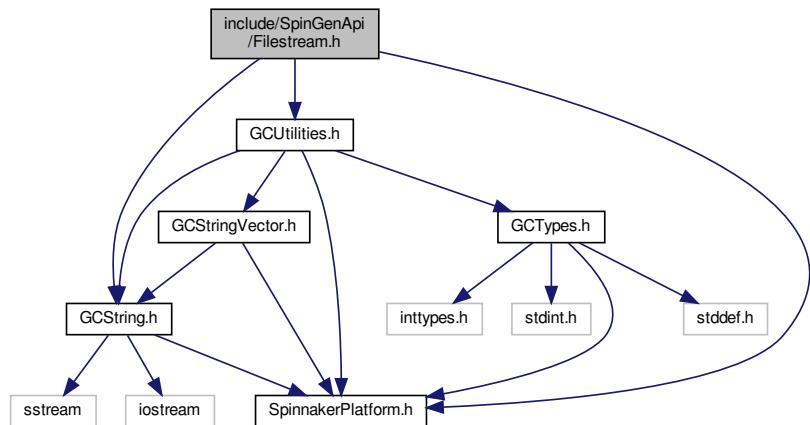
*Port attachable to an event.*

## Namespaces

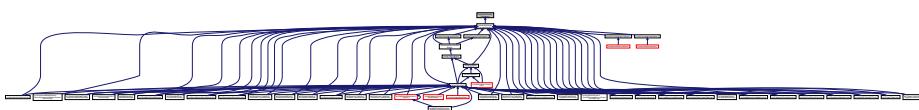
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.80 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 >](#)

# Namespaces

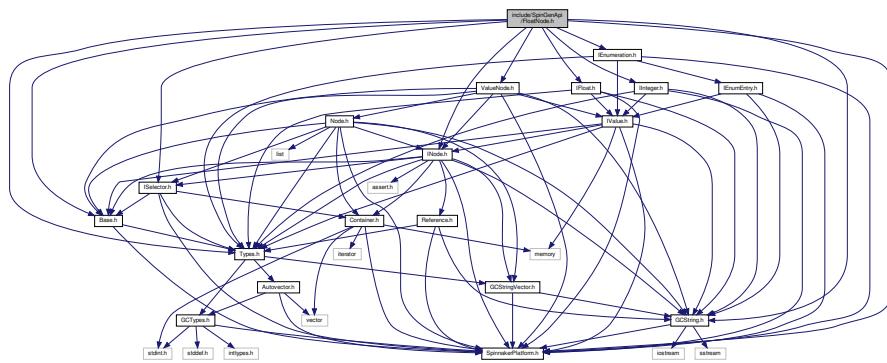
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

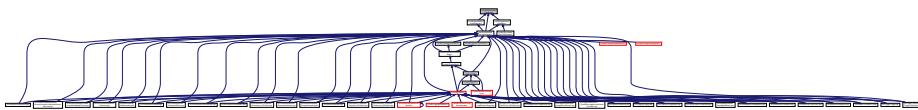
- `typedef ODevFileStreamBase< char, std::char_traits< char > > ODevFileStream`
  - `typedef IDevFileStreamBase< char, std::char_traits< char > > IDevFileStream`

## 15.81 include/SpinGenApi/FloatNode.h File Reference

Include dependency graph for `FloatNode.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `FloatNode`  
*Interface for string properties.*

## Namespaces

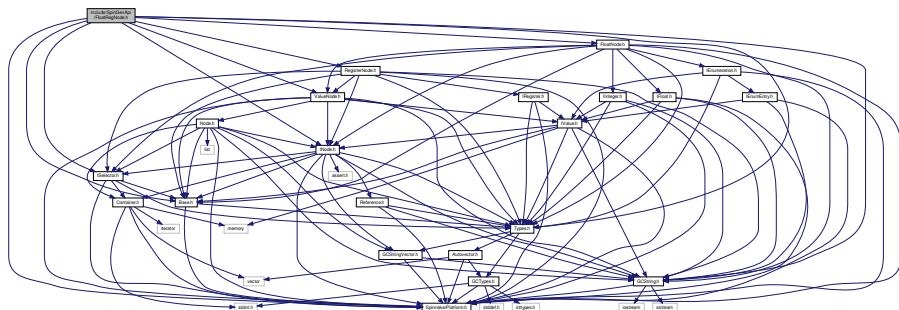
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

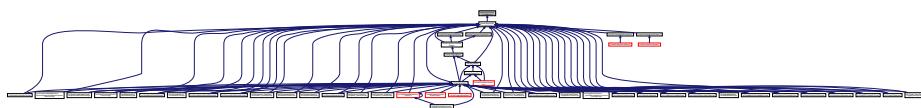
- `typedef FloatNode CFloatRef`

## 15.82 include/SpinGenApi/FloatRegNode.h File Reference

Include dependency graph for FloatRegNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

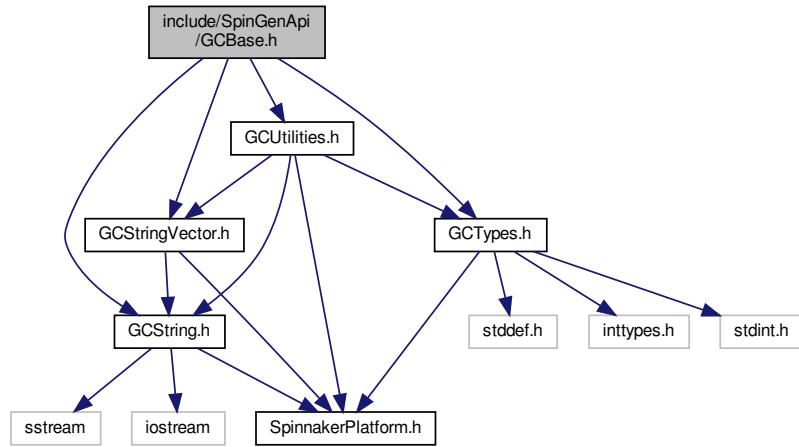
- class `FloatRegNode`

## Namespaces

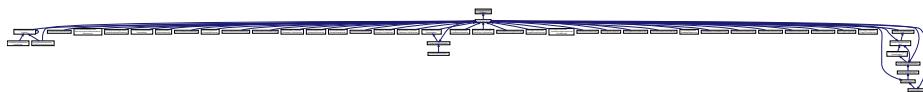
- Spinnaker
  - Spinnaker::GenApi

## 15.83 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

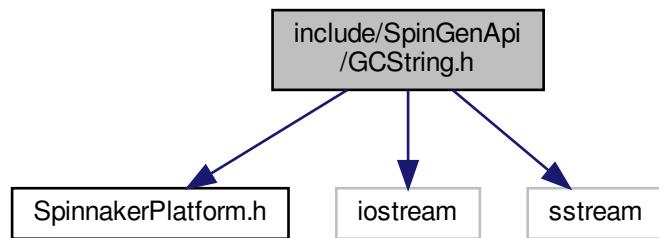


This graph shows which files directly or indirectly include this file:

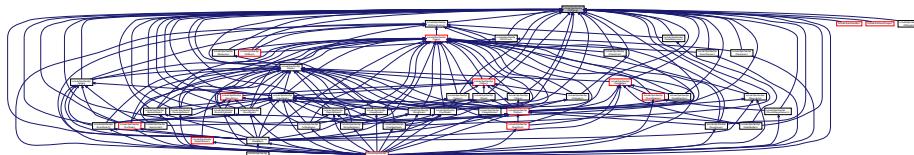


## 15.84 include/SpinGenApi/GCString.h File Reference

Include dependency graph for GCString.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gcstring](#)

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## Macros

- `#define GCSTRING_NPOS size_t(-1)`

## 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.*
- `std::ostream & operator<< (std::ostream &ostr, const Spinnaker::GenICam::gcstring &str)`  
*STL operator out.*
- `std::istream & operator>> (std::istream &istr, Spinnaker::GenICam::gcstring &str)`  
*STL operator in.*

### 15.84.1 Macro Definition Documentation

#### 15.84.1.1 GCSTRING\_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

### 15.84.2 Function Documentation

### 15.84.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & ostr,
    const Spinnaker::GenICam::gcstring & str ) [inline]
```

STL operator out.

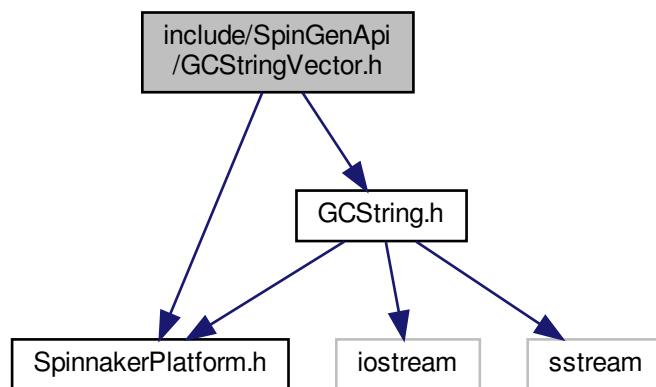
### 15.84.2.2 operator>>()

```
std::istream& operator>> (
    std::istream & istr,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

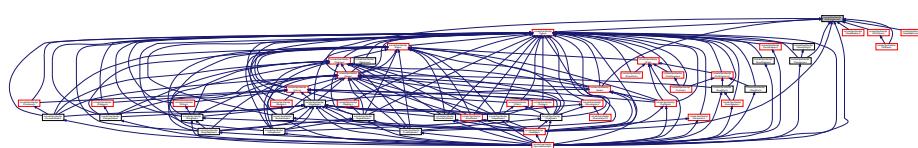
STL operator in.

## 15.85 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

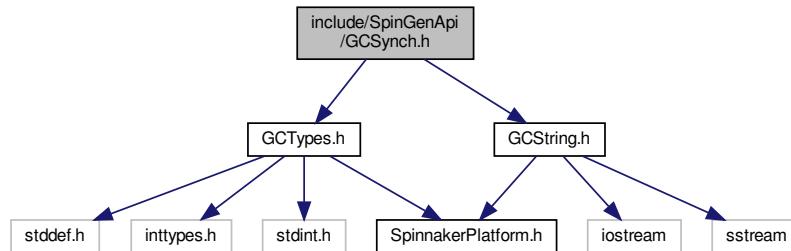


This graph shows which files directly or indirectly include this file:



## 15.86 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



## Classes

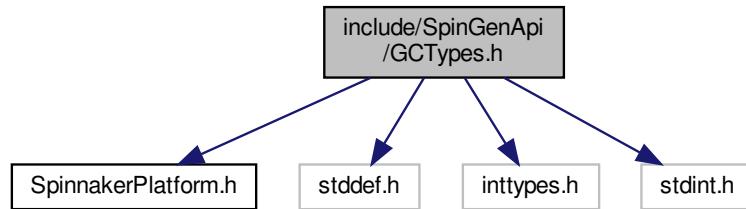
- class [CLOCK](#)  
*A lock class.*
- class [CLOCKEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)
- class [LockableObject< Object >](#)  
*Instance-Lock for an object.*
- class [LockableObject< Object >::Lock](#)  
*A scopelevel Lock class.*
- class [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*

## Namespaces

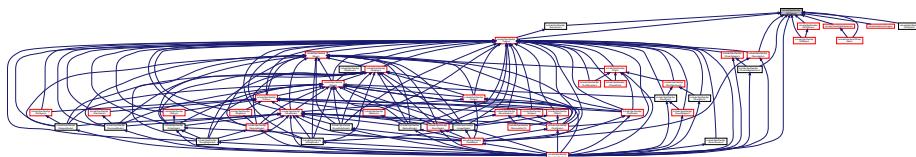
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

## 15.87 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>(0xfffffffffffffffULL) /* 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>(0x00000000ffffffffffULL) /* maximum unsigned int32 value */`
- `#define GC_INT8_MAX static_cast<int64_t>(0x0000000000000007fLL) /* 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*

### 15.87.1 Macro Definition Documentation

#### 15.87.1.1 \_\_STDC\_CONSTANT\_MACROS

```
#define __STDC_CONSTANT_MACROS
```

#### 15.87.1.2 \_\_STDC\_LIMIT\_MACROS

```
#define __STDC_LIMIT_MACROS
```

#### 15.87.1.3 GC\_INT32\_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */
```

#### 15.87.1.4 GC\_INT32\_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */
```

#### 15.87.1.5 GC\_INT64\_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffLL) /* maximum signed int64 value */
```

### 15.87.1.6 GC\_INT64\_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */
```

### 15.87.1.7 GC\_INT8\_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x0000000000000007fLL) /* maximum signed int8 value */
```

### 15.87.1.8 GC\_INT8\_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffff80LL) /* minimum signed int8 value */
```

### 15.87.1.9 GC\_UINT32\_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */
```

### 15.87.1.10 GC\_UINT64\_MAX

```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */
```

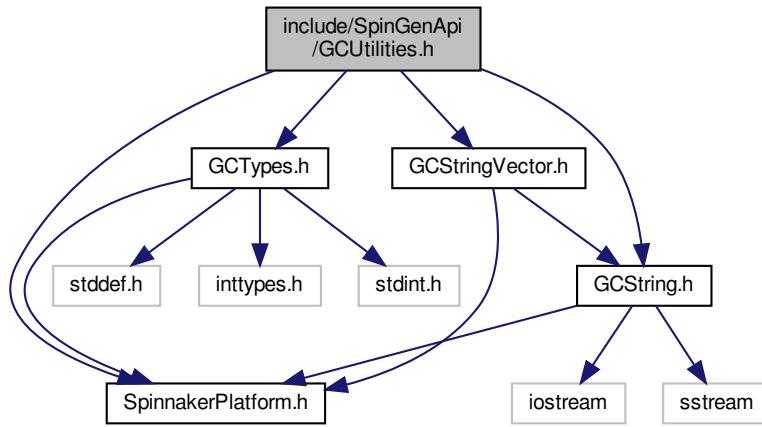
### 15.87.1.11 GC\_UINT8\_MAX

```
#define GC_UINT8_MAX static_cast<uint64_t>(0x000000000000ffULL) /* maximum unsigned int8 value */
```

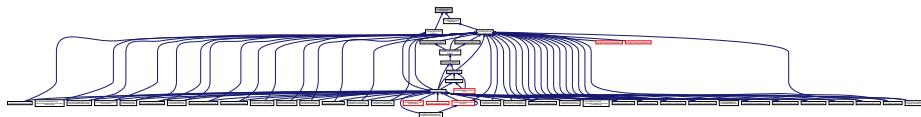
---

## 15.88 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 **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.*

### 15.88.1 Macro Definition Documentation

**15.88.1.1 \_\_ERR\_\_**

```
#define __ERR__ __OUTPUT_FORMATER__("ERROR")
```

**15.88.1.2 \_\_LINE\_STR\_\_**

```
#define __LINE_STR__ EXPAND_TO_STRINGISE(__LINE__)
```

**15.88.1.3 \_\_LOCATION\_\_**

```
#define __LOCATION__ __FILE__ "(" __LINE_STR__ ") "
```

**15.88.1.4 \_\_OUTPUT\_FORMATER\_\_**

```
#define __OUTPUT_FORMATER__(  
    _type ) __LOCATION__ " : " _type " : "
```

**15.88.1.5 \_\_TODO\_\_**

```
#define __TODO__ __OUTPUT_FORMATER__("TBD")
```

**15.88.1.6 \_\_WARN\_\_**

```
#define __WARN__ __OUTPUT_FORMATER__("WARNING")
```

**15.88.1.7 \_TO\_STRING**

```
#define _TO_STRING(  
    __stN ) #__stN
```

**15.88.1.8 EXPAND\_TO\_STRINGISE**

```
#define EXPAND_TO_STRINGISE(  
    __stN ) _TO_STRING(__stN)
```

**15.88.1.9 GC\_COUNTOF**

```
#define GC_COUNTOF(  
    arr ) (sizeof(arr) / sizeof(arr)[0])
```

**15.88.1.10 GENICAM\_DEPRECATED**

```
#define GENICAM_DEPRECATED(  
    FUNCTION ) FUNCTION
```

**15.88.1.11 GENICAM\_UNUSED**

```
#define GENICAM_UNUSED(  
    unused_var ) ((void)(unused_var))
```

**15.88.1.12 USE\_TEMP\_CACHE\_FILE [1/2]**

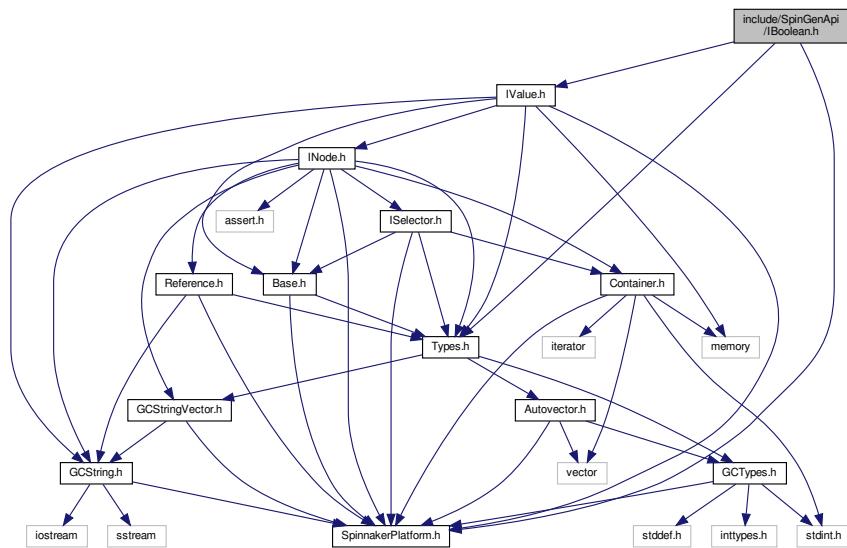
```
#define USE_TEMP_CACHE_FILE 1
```

**15.88.1.13 USE\_TEMP\_CACHE\_FILE [2/2]**

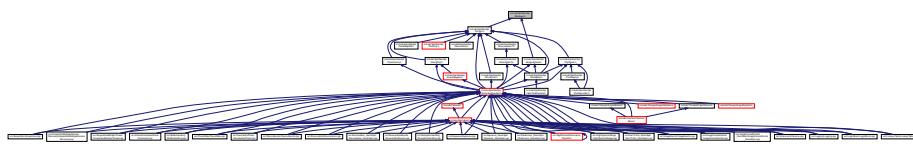
```
#define USE_TEMP_CACHE_FILE 1
```

## 15.89 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

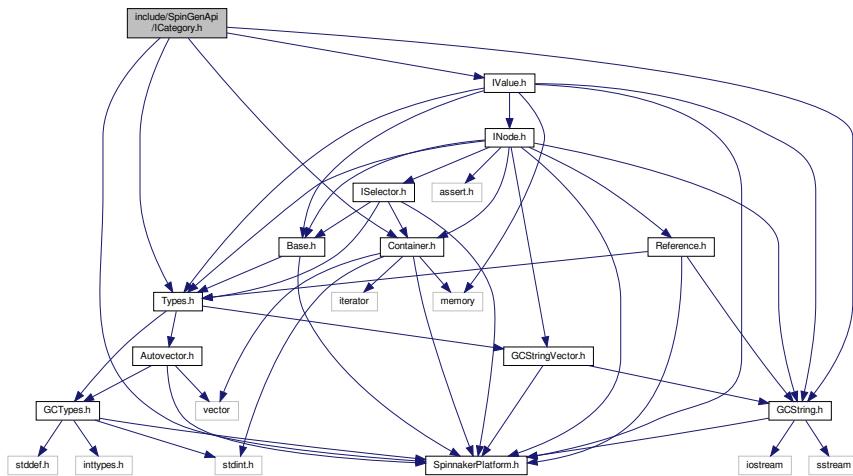
- virtual void `operator=` (bool Value)  
*Set node value.*
- virtual bool `GetValue` (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool `operator()` () const  
*Get node value.*

### Variables

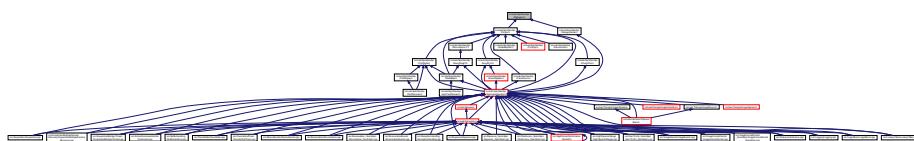
- interface `SPINNAKER_API_ABSTRACT IBoolean`  
*Interface for Boolean properties.*
- interface `SPINNAKER_API_ABSTRACT bool Verify = true) = 0`

## 15.90 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

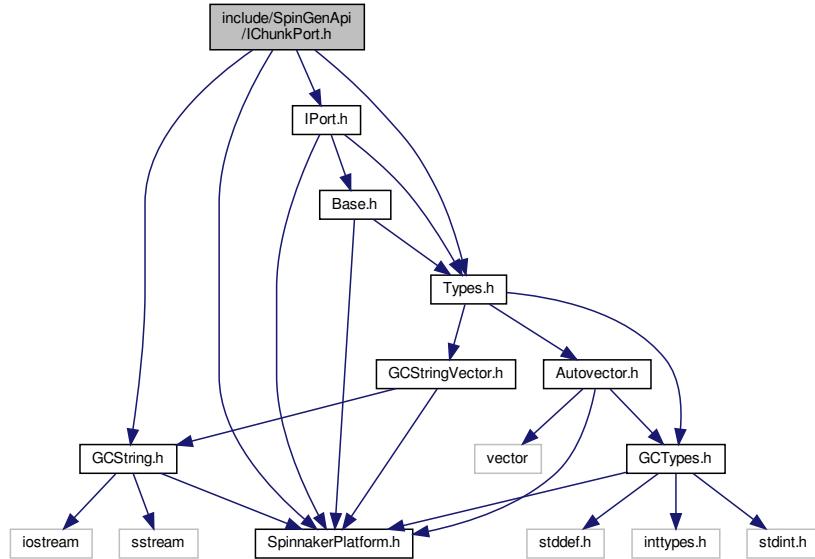
## Variables

- interface SPINNAKER\_API\_ABSTRACT ICategory

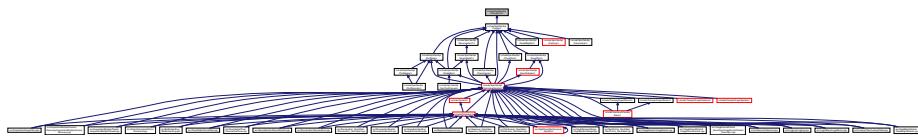
*Gives access to a category node.*

## 15.91 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.*

### 15.91.1 Macro Definition Documentation

#### 15.91.1.1 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)

#### 15.91.1.2 CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Lenght of the CHUNK\_BASE\_ADDRESS\_REGISTER pseudo register.

#### 15.91.1.3 CHUNK\_LENGTH\_REGISTER

```
#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX - 15)
```

Address of a int64\_t pseudo register containing the length of the chunk.

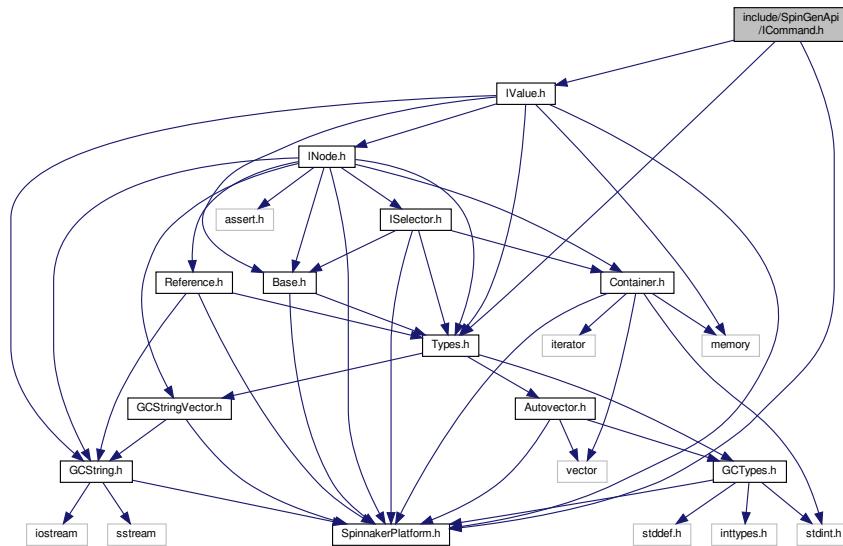
#### 15.91.1.4 CHUNK\_LENGTH\_REGISTER\_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

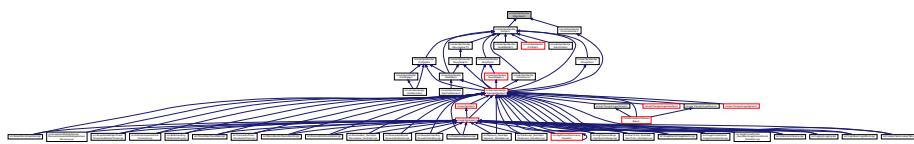
Lenght of the CHUNK\_LENGTH\_REGISTER pseudo register.

## 15.92 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

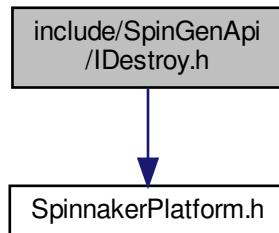
- virtual bool [`operator\(\)`](#) () const  
*Get node value.*
- virtual bool [`IsDone`](#) (bool Verify=true)=0  
*Query whether the command is executed.*

## Variables

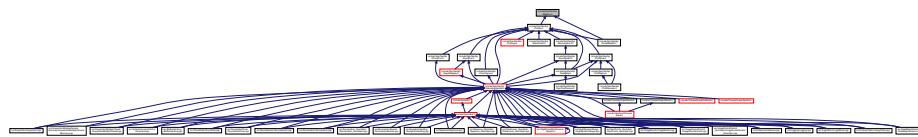
- interface [`SPINNAKER\_API\_ABSTRACT ICommand`](#)  
*Interface for command like properties.*

## 15.93 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](#)

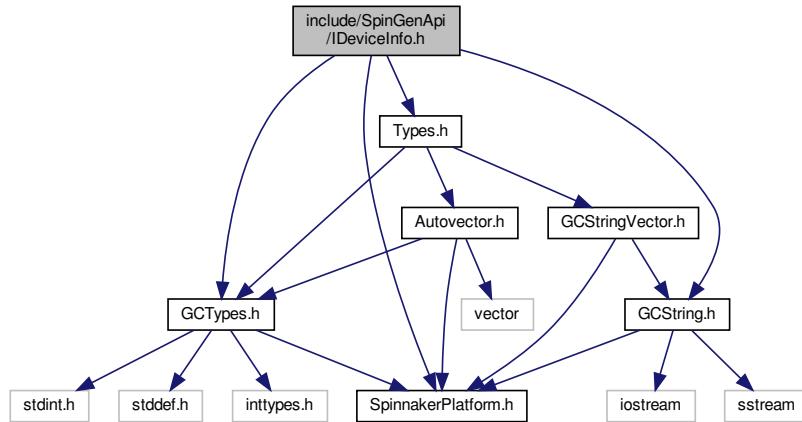
### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IDestroy](#)

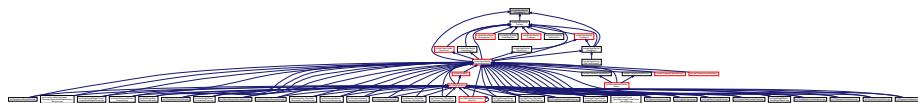
*Interface to destroy an object.*

## 15.94 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

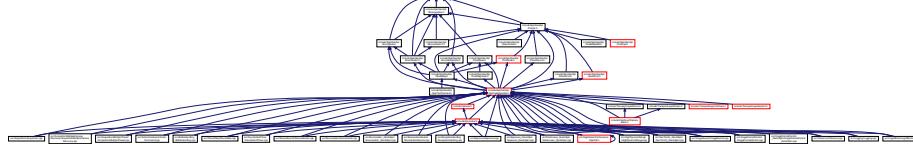
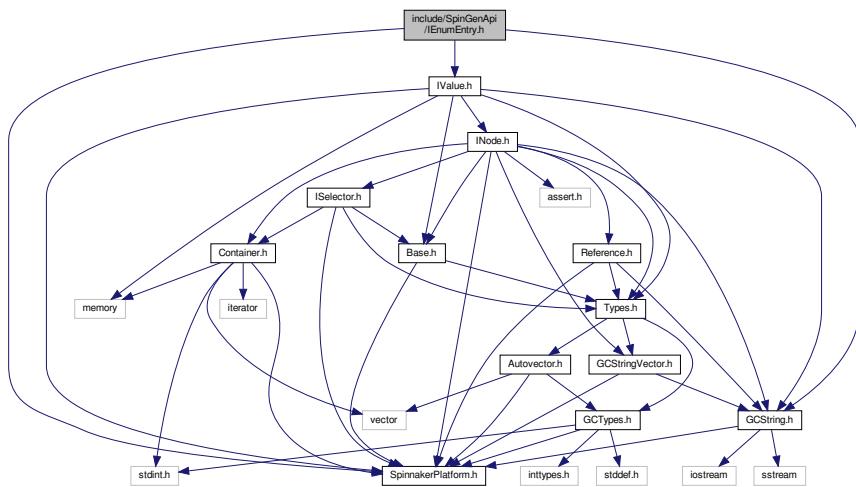
- virtual `GenICam::gcstring GetVendorName ()=0`  
`Get the vendor name.`
- virtual `GenICam::gcstring GetToolTip ()=0`  
`Get tool tip.`
- virtual `GenICam::gcstring GetStandardNameSpace ()=0`  
`Get the standard name space.`
- virtual void `GetGenApiVersion (GenICam::Version_t &Version, uint16_t &Build)=0`  
`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 IDeviceInfo  
*Interface to get information about the device (= nodemap)*

## 15.95 include/SpinGenApi/IEnumEntry.h File Reference

Include dependency graph for IEnumEntry.h:



## Namespaces

- Spinnaker
- Spinnaker::GenApi

## Functions

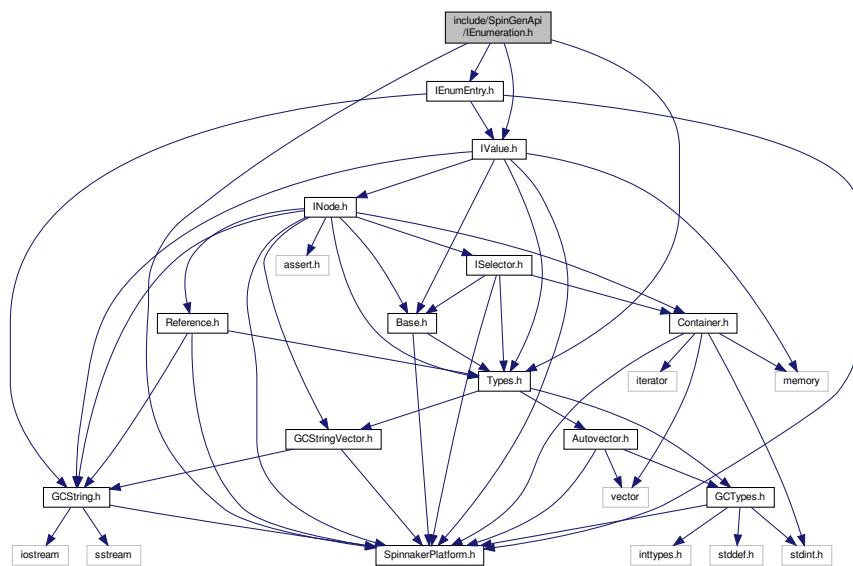
- virtual GenICam::gcstring GetSymbolic () const =0  
*Get symbolic enum value.*
- virtual double GetNumericValue ()=0  
*Get double number associated with the entry.*
- virtual bool IsSelfClearing ()=0  
*Indicates if the corresponding EnumEntry is self clearing.*

## Variables

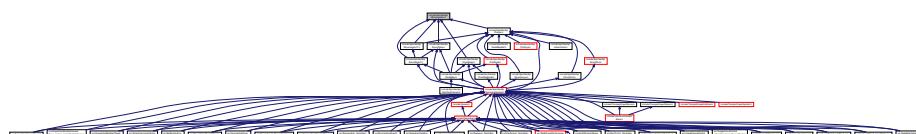
- interface SPINNAKER\_API\_ABSTRACT IEnumEntry  
*Interface of single enum value.*

## 15.96 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

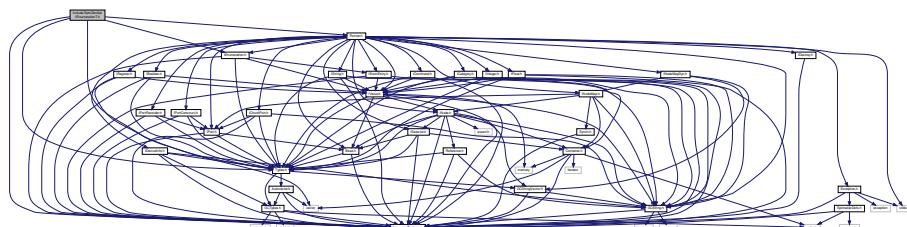
- *Set integer node value.*
- virtual GenICam::gcstring [operator\\*](#) ()=0  
*Get string node value.*
- virtual int64\_t [GetIntValue](#) (bool Verify=false, bool IgnoreCache=false)=0  
*Get integer node value.*
- virtual IEnumEntry \* [GetEntryByName](#) (const GenICam::gcstring &Symbolic)=0  
*Get 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

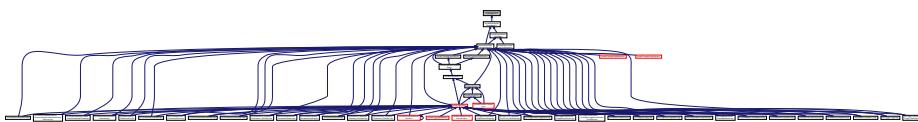
- interface SPINNAKER\_API\_ABSTRACT IEnumeration  
*Interface for enumeration properties.*

## 15.97 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

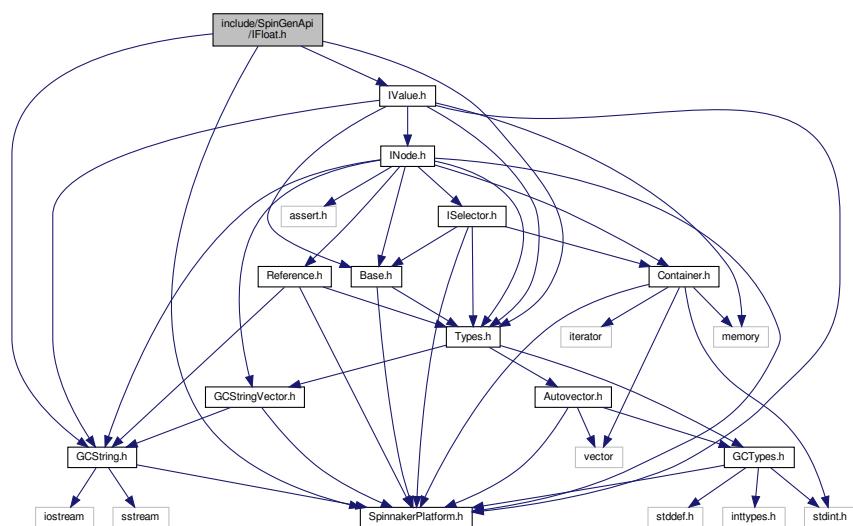
- virtual IEnumeration & **operator=** (EnumT Value)=0  
*Set node value.*
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool **operator()** () const  
*Get node value.*
- virtual IEnumeration & **operator=** (const GenlCam::gcstring &ValueStr)=0  
*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

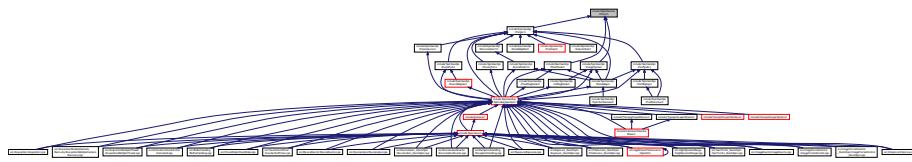
- template<typename EnumT >  
**interface SPINNAKER\_API\_ABSTRACT IEnumerationT**  
*Interface for enumeration properties.*
- **interface SPINNAKER\_API\_ABSTRACT** virtual public **IEnumReference**  
*Interface to construct an enum reference.*

## 15.98 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

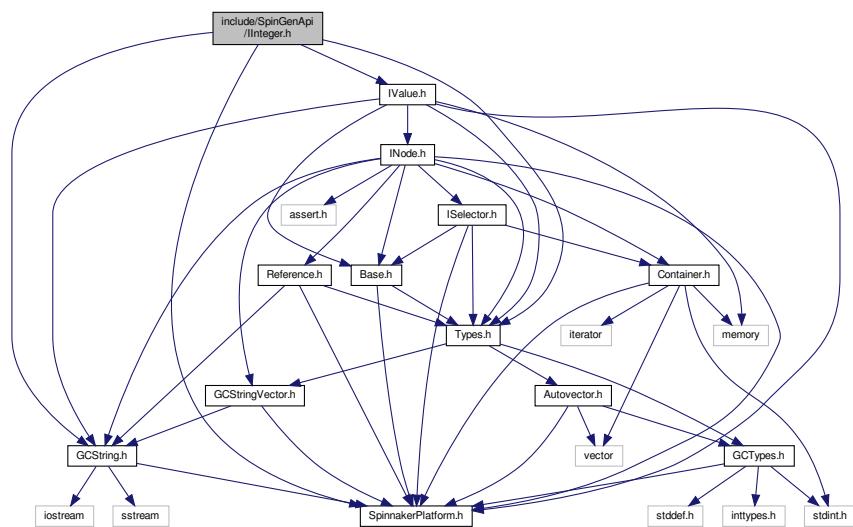
- virtual IFloat & [operator=](#) (double Value)=0  
*Set node value.*
- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool [operator\(\)](#) () const  
*Get node value.*
- virtual GenICam::gcstring [operator\\*](#) ()=0  
*Get string node 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

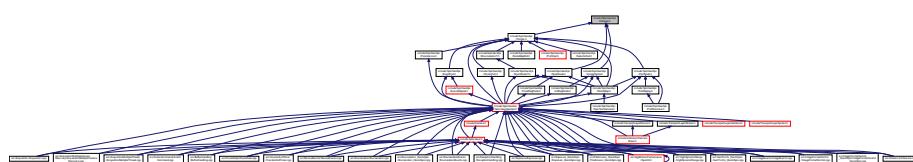
- interface SPINNAKER\_API\_ABSTRACT IFloat  
*Interface for float properties.*

## 15.99 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker
- Spinnaker::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

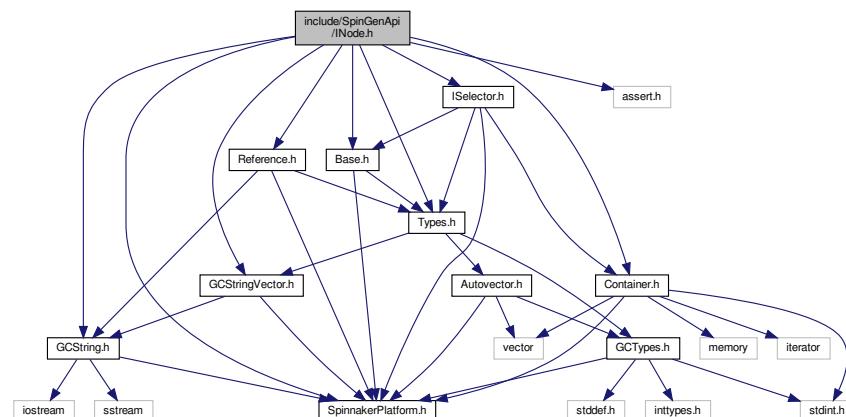
- *Get node value.*
- virtual GenICam::gcstring **operator\*** ()=0
  - Get string node value.*
- virtual double **GetMin** ()=0
  - Get minimum value allowed.*
- virtual double **GetMax** ()=0
  - Get maximum value allowed.*
- virtual EIncMode **GetIncMode** ()=0
  - Get increment mode.*
- virtual double **GetInc** ()=0
  - Get the constant increment if there is any.*
- virtual double<sub>\_autovector\_t</sub> **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

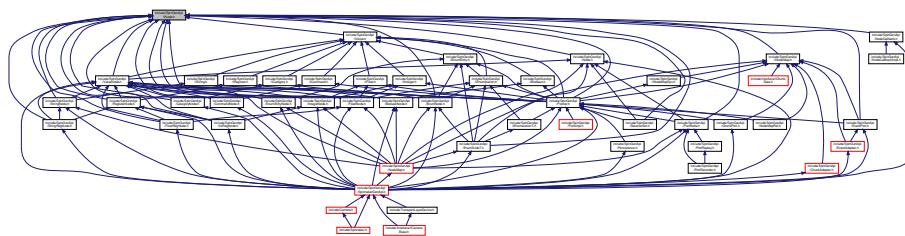
- interface SPINNAKER\_API\_ABSTRACT IInteger
  - Interface for integer properties.*

## 15.100 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 IsCachable () 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)

- bool **IsImplemented** (const IBase &r)
 

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

*Checks if a node is implemented.*
- bool **IsAvailable** (const IBase \*p)
 

*Tests if 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

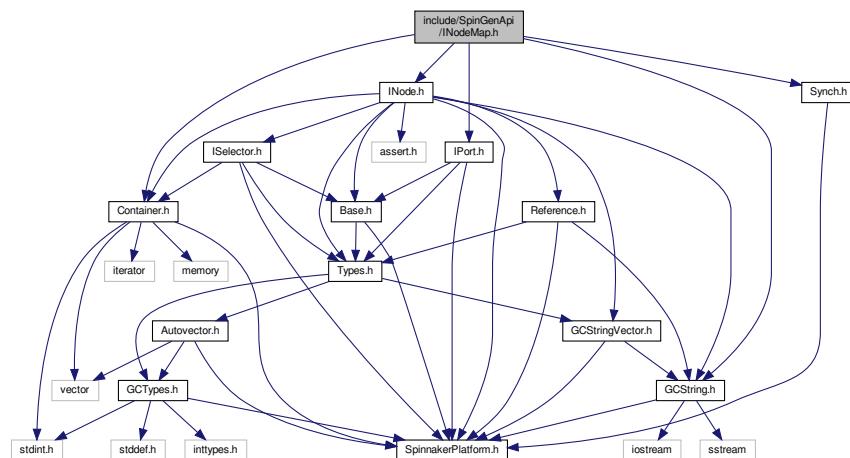
- interface SPINNAKER\_API\_ABSTRACT INode
 

*Interface common to all nodes.*
- interface SPINNAKER\_API\_ABSTRACT virtual public IReference
 

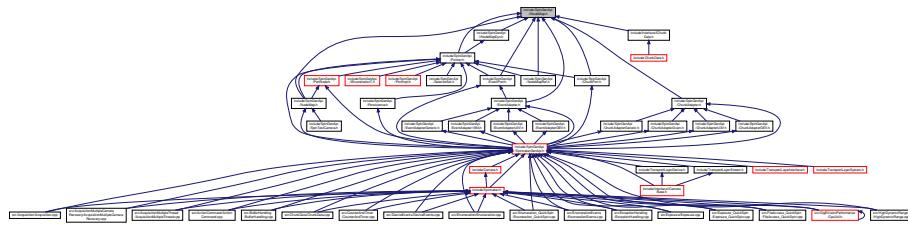
*Interface to construct a reference.*

## 15.101 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

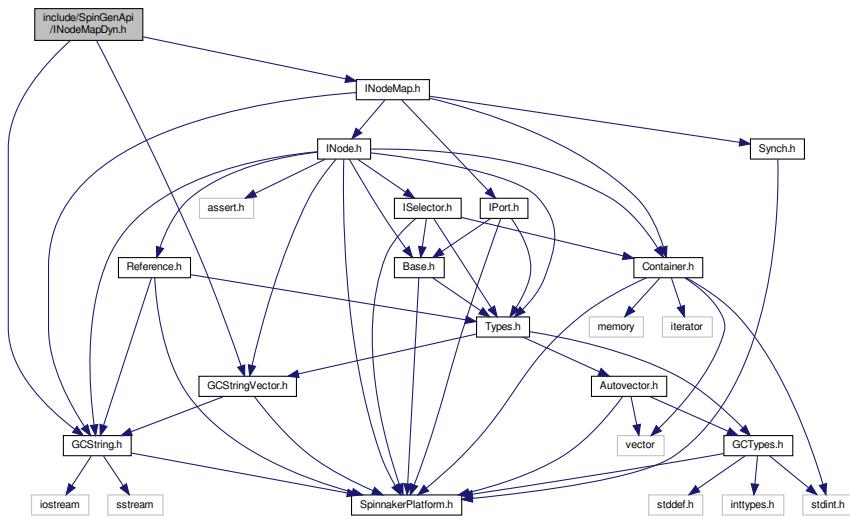
- virtual `INode *` [GetNode](#) (`const GenICam::gcstring &Name`) `const =0`  
*Retrieves the node from the central map by Name.*
- virtual void [InvalidateNodes](#) () `const =0`  
*Invalidates all nodes.*
- virtual bool [Connect](#) (`IPort *pPort, const GenICam::gcstring &PortName`) `const =0`  
*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.*

## 15.102 include/SpinGenApi/INodeMapDyn.h File Reference

Include dependency graph for INodeMapDyn.h:



This graph shows which files directly or indirectly include this file:

## Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Functions

- 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

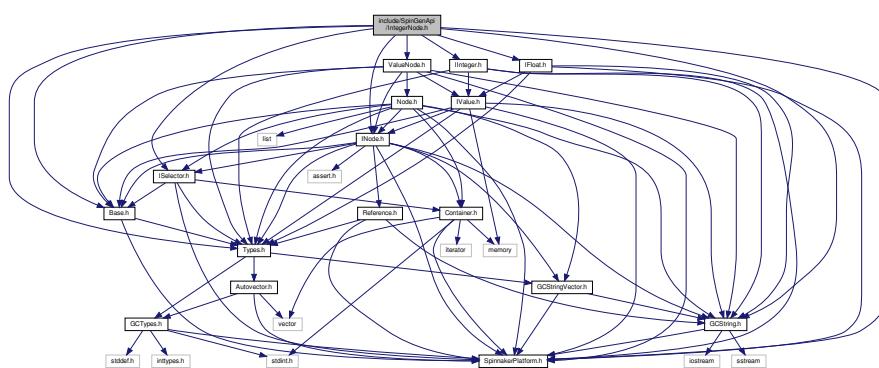
- Loads an XML, checks it for correctness, pre-processes it, caches it, and optionally applies a style sheet, and optionally writes it to a file.*
- virtual void [MergeXMLFiles](#) (const GenICam::gcstring &TargetFileName, const GenICam::gcstring &InjectedFileName, const GenICam::gcstring &OutputFileName)=0  
*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.*

## Variables

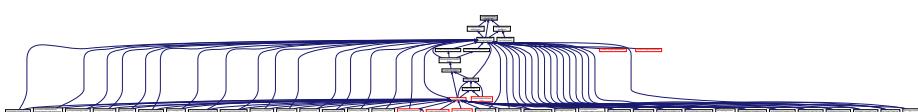
- interface [SPINNAKER\\_API\\_ABSTRACT INodeMapDyn](#)  
*Interface to access the node map.*

## 15.103 include/SpinGenApi/IntegerNode.h File Reference

Include dependency graph for IntegerNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class `IntegerNode`  
*Interface for string properties.*

# Namespaces

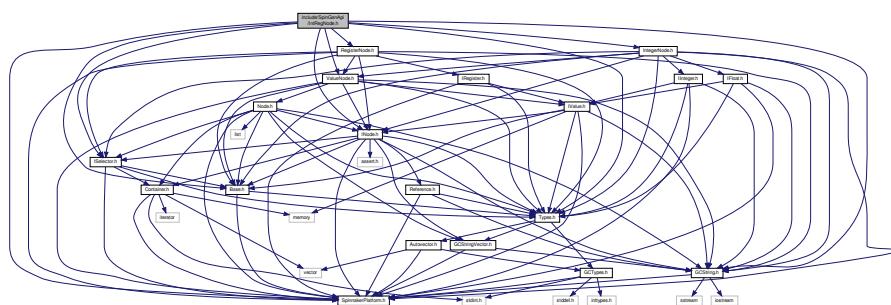
- Spinnaker
  - Spinnaker::GenApi

## TypeDefs

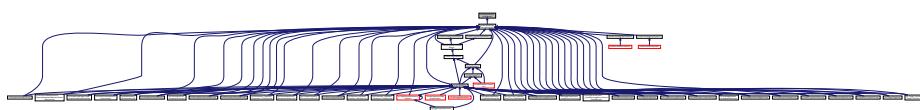
- `typedef IntegerNode CIntegerRef`

## 15.104 include/SpinGenApi/IntRegNode.h File Reference

Include dependency graph for IntRegNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

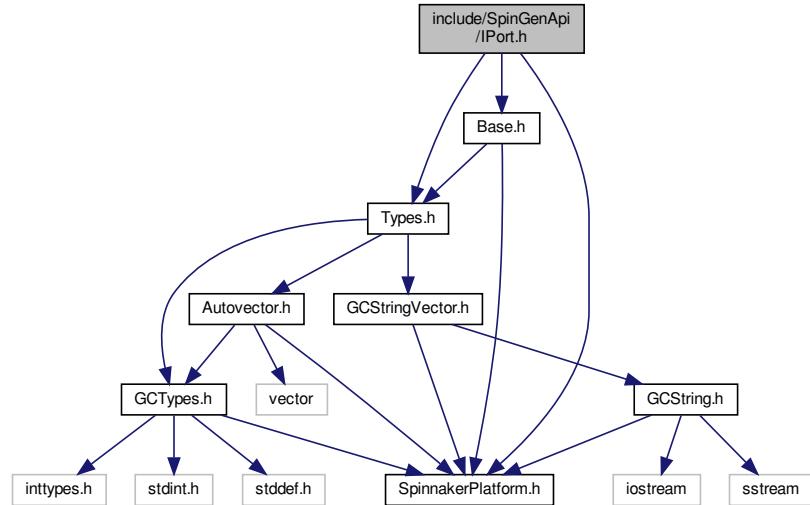
- class IntRegNode  
*Interface for string properties.*

## Namespaces

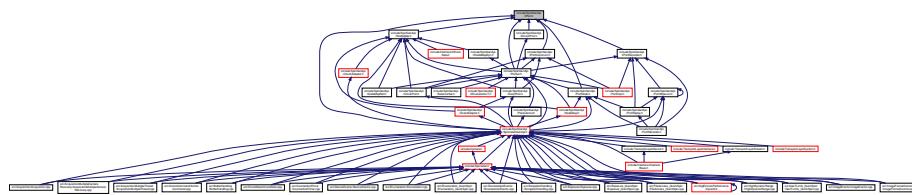
- Spinnaker
  - Spinnaker::GenApi

## 15.105 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

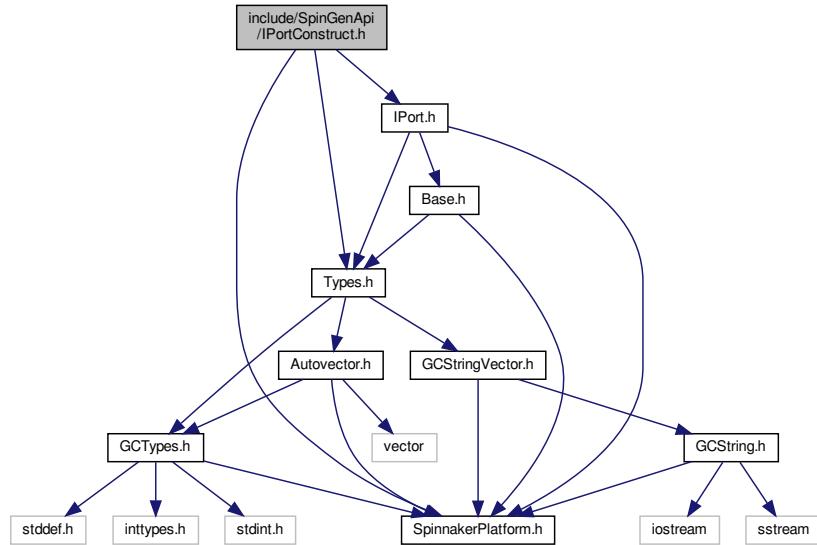
- virtual void `Write` (const void \*pBuffer, int64\_t Address, int64\_t Length)=0  
*Writes a chunk of bytes to the port.*

## Variables

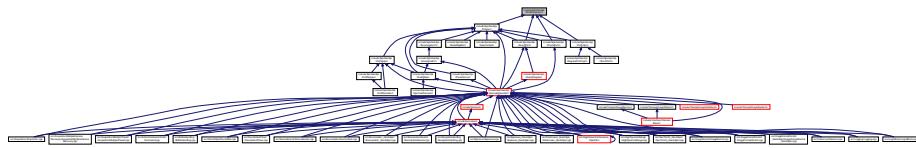
- interface SPINNAKER\_API\_ABSTRACT `IPort`  
*Interface for ports.*
- interface SPINNAKER\_API\_ABSTRACT int64\_t `Address`
- interface SPINNAKER\_API\_ABSTRACT int64\_t int64\_t `Length` = 0

## 15.106 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

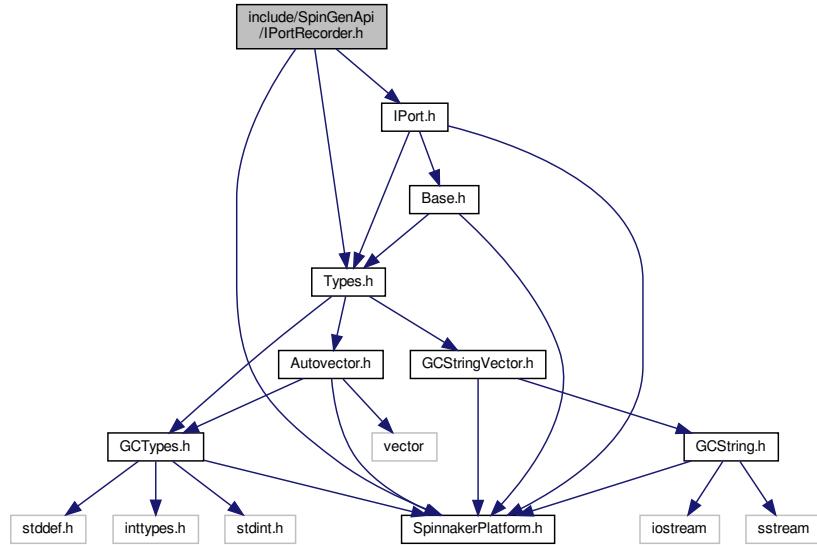
- virtual EYesNo [GetSwapEndianess \(\)=0](#)  
*Determines if the port adapter must perform an endianess swap.*

## Variables

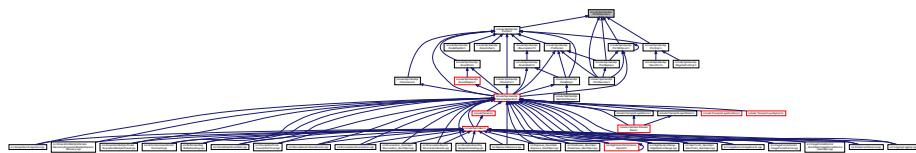
- interface SPINNAKER\_API IPortConstruct  
*Interface for ports.*

## 15.107 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

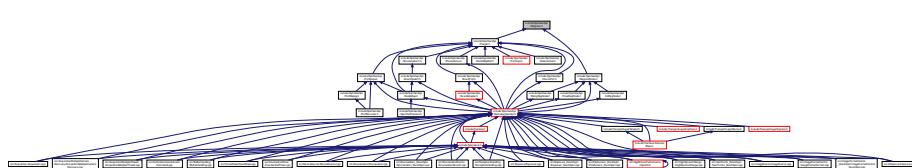
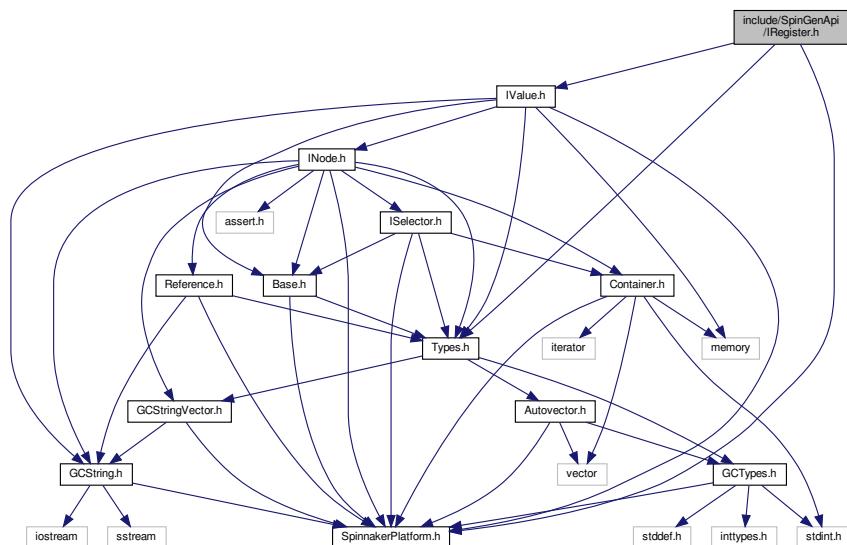
- virtual void [Replay](#) (IPort \*pPort)=0  
*Replays the write command to the given port interface.*
- virtual void [SetCookie](#) (const int64\_t Value)=0  
*Sets a cookie in case the port implementation want to cache a command list.*
- virtual int64\_t [GetCookie](#) ()=0  
*Gets the cookie a port 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.*

## 15.108 include/SpinGenApi/IRegister.h File Reference

Include dependency graph for IRegister.h:



## Namespaces

- Spinnaker
- Spinnaker::GenApi

## Functions

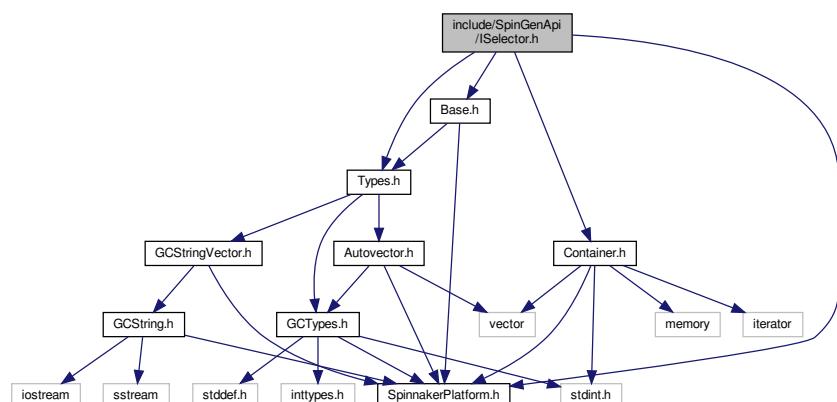
- virtual void [Get](#) (uint8\_t \*pBuffer, int64\_t Length, bool Verify=false, bool IgnoreCache=false)=0  
*Fills a buffer with the register's contents.*
- virtual int64\_t [GetLength](#) ()=0  
*Retrieves the Length of the register [Bytes].*
- virtual int64\_t [GetAddress](#) ()=0  
*Retrieves the Address of the register.*

## Variables

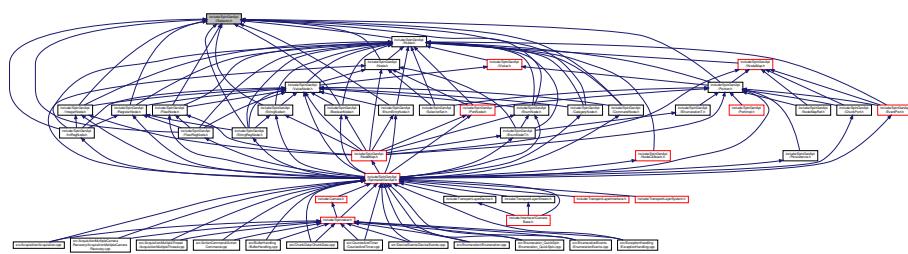
- [interface SPINNAKER\\_API\\_ABSTRACT IRegister](#)  
*Interface for registers.*

## 15.109 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

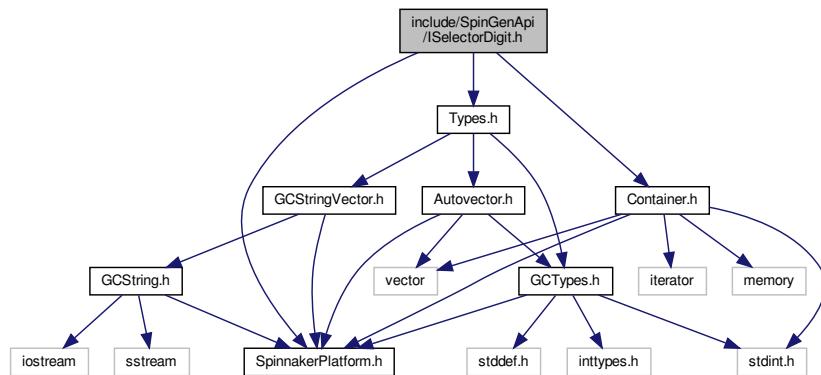
- virtual void [GetSelectedFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of selected features*
- virtual void [GetSelectingFeatures](#) (FeatureList\_t &) const =0  
*retrieve the group of features selecting this node*

## Variables

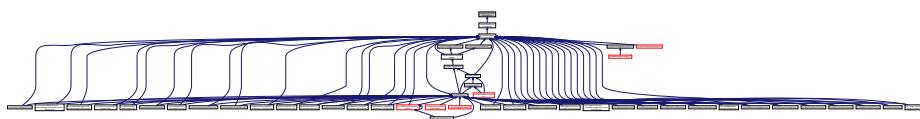
- interface [SPINNAKER\\_API\\_ABSTRACT ISelector](#)  
*Interface for groups of features selected by a single one.*

## 15.110 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

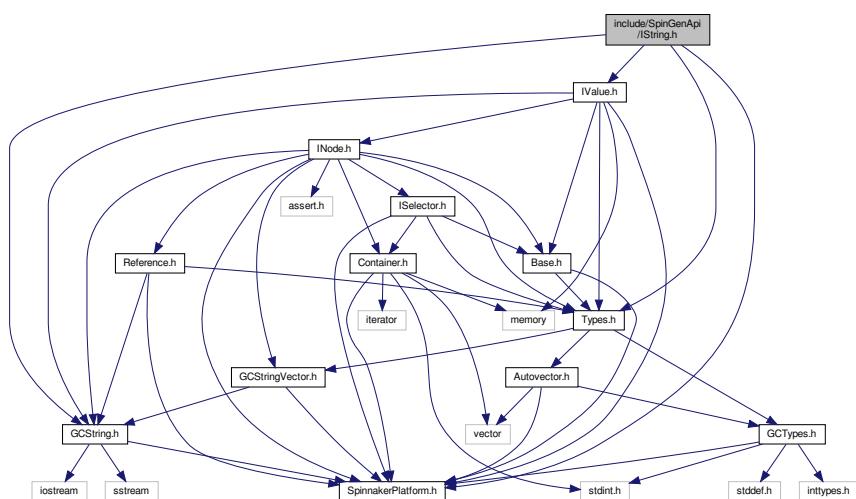
- virtual bool `SetNext` (bool Tick=true)=0  
*Sets digit to next value.*
- virtual void `Restore` ()=0  
*Restores the selectors' values found at creation.*
- virtual GenICam::gcstring `ToString` ()=0  
*Returns a string representation of the digit.*
- virtual void `GetSelectorList` (FeatureList\_t &SelectorList, bool Incremental=false)=0  
*Retrieves an ordered list of selectors.*

## Variables

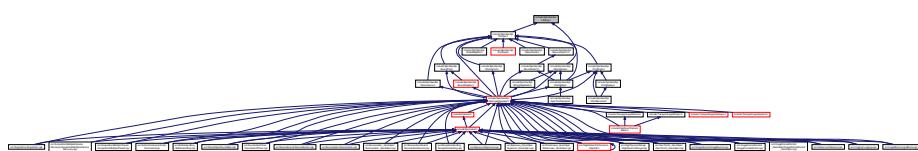
- interface SPINNAKER\_API\_ABSTRACT ISelectorDigit  
*Interface of a "digit" of the "counter" formed by the selector set.*

## 15.111 include/SpinGenApi/IString.h File Reference

Include dependency graph for IString.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker
- Spinnaker::GenApi

## Functions

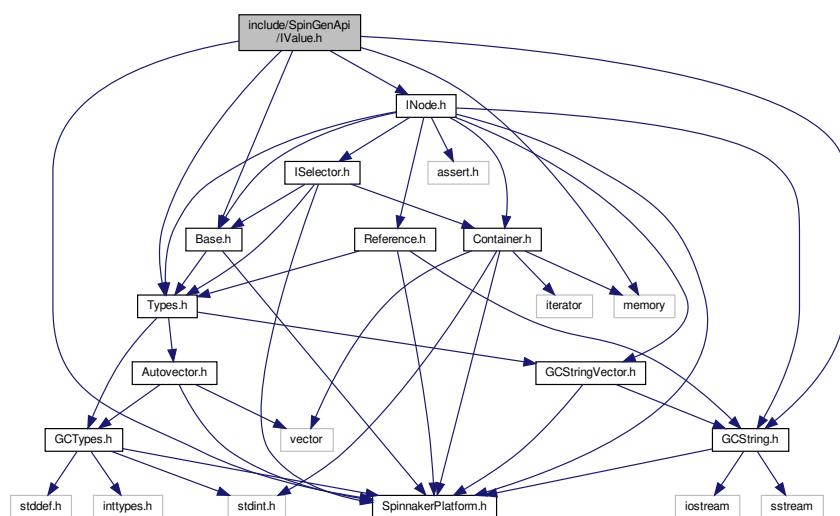
- virtual IEnumeration & **operator=** (const GenICam::gcstring &ValueStr)=0  
*Set string node value.*
- virtual bool **GetValue** (bool Verify=false, bool IgnoreCache=false) const =0  
*Get node value.*
- virtual bool **operator()** () const  
*Get node value.*
- virtual GenICam::gcstring **operator\*** ()=0  
*Get string node value.*
- virtual int64\_t **GetMaxLength** ()=0  
*Retrieves the maximum length of the string in bytes.*

## Variables

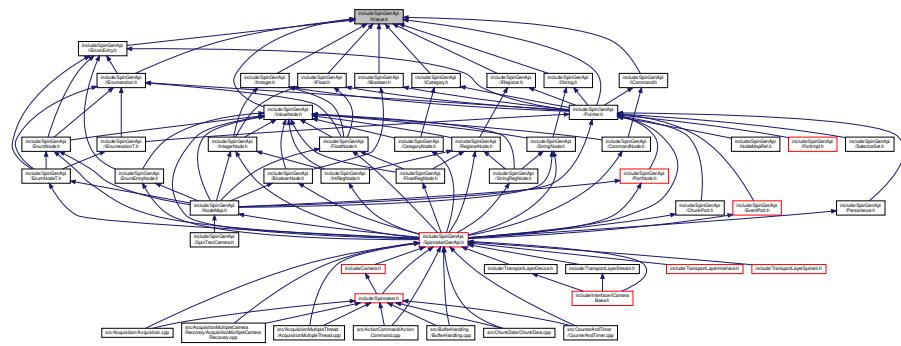
- interface SPINNAKER\_API\_ABSTRACT IString  
*Interface for string properties.*

## 15.112 include/SpinGenApi/IValue.h File Reference

Include dependency graph for IValue.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker
- Spinnaker::GenApi

## Functions

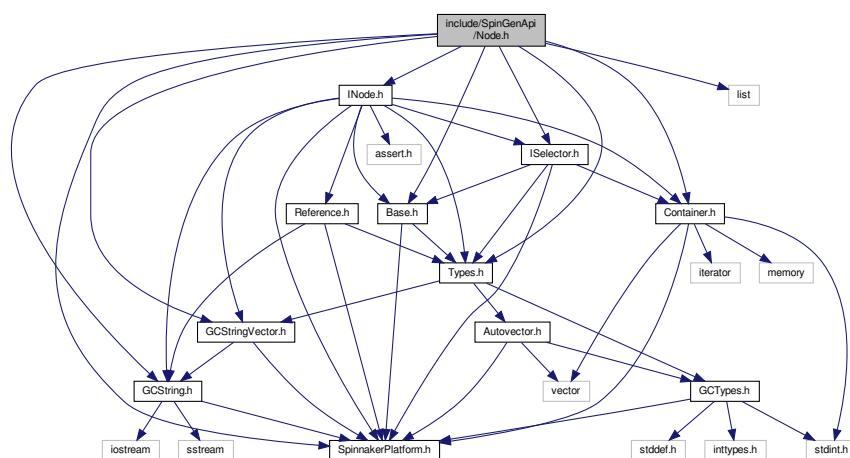
- virtual GenICam::gcstring **ToString** (bool Verify=false, bool IgnoreCache=false)=0  
*Get content of the node as string.*
- virtual void **FromString** (const GenICam::gcstring &ValueStr, bool Verify=true)=0  
*Set content of the node as string.*
- virtual bool **IsValueCacheValid** () const =0  
*Checks if the value comes from cache or is requested from another node.*

## Variables

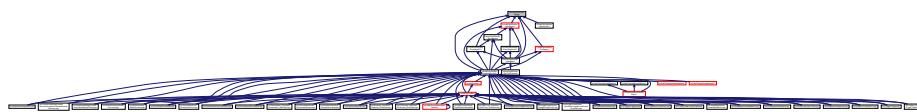
- interface SPINNAKER\_API\_ABSTRACT IValue  
*Interface for value properties.*

## 15.113 include/SpinGenApi/Node.h File Reference

Include dependency graph for Node.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Node](#)  
*class common to all nodes*

## Namespaces

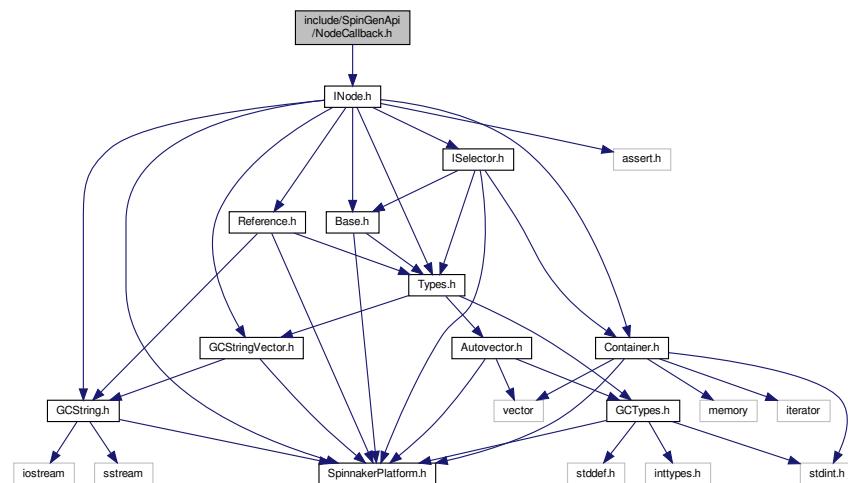
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

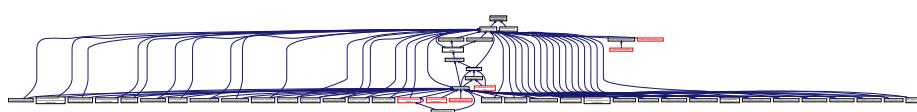
- typedef Node [CNodeRef](#)
- typedef Node [CSelectorRef](#)

## 15.114 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

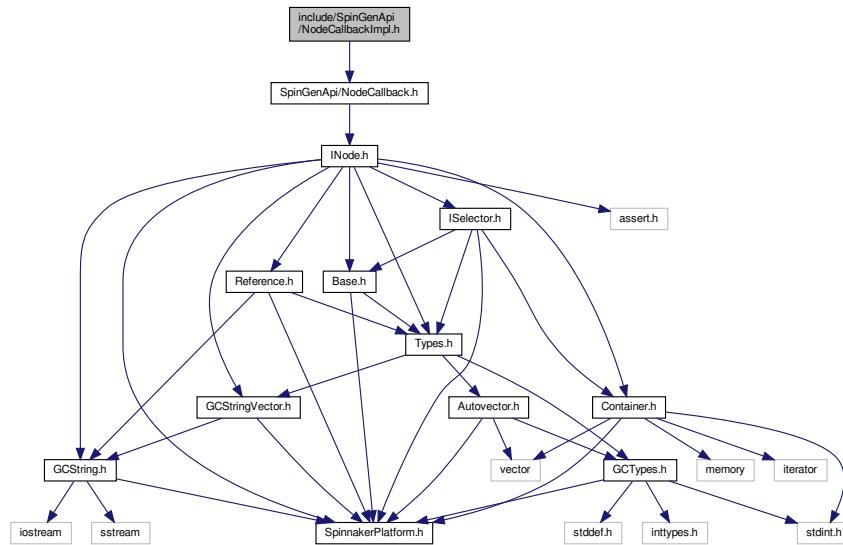
- enum [\\_ECallbackType](#) {  
  cbPostInsideLock = 1,  
  cbPostOutsideLock = 2 }
- the type of callback*

## 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.*

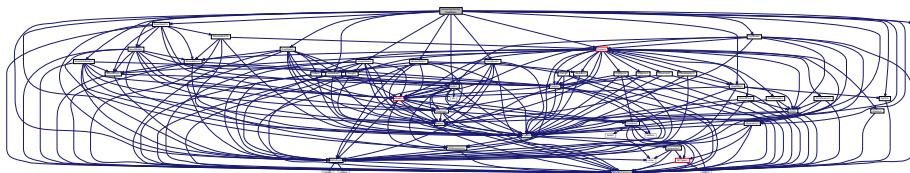
## 15.115 include/SpinGenApi/NodeCallbackImpl.h File Reference

Include dependency graph for NodeCallbackImpl.h:

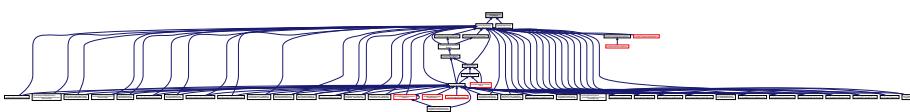


## 15.116 include/SpinGenApi/NodeMap.h File Reference

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

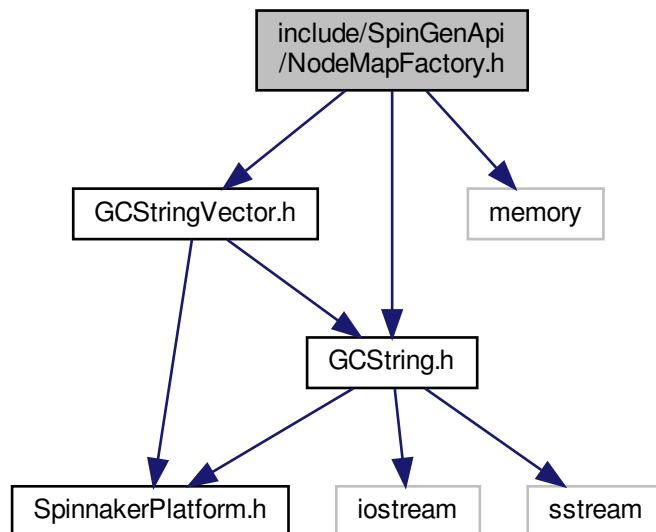
- Spinnaker
- Spinnaker::GenApi

## Typedefs

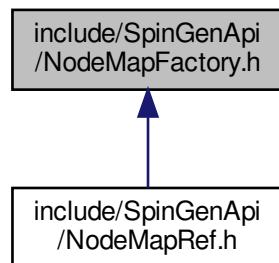
- `typedef NodeMap CNodeMapRef`

## 15.117 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

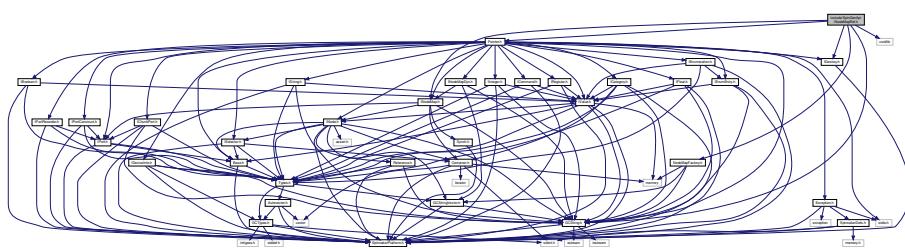
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 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.*

## 15.118 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

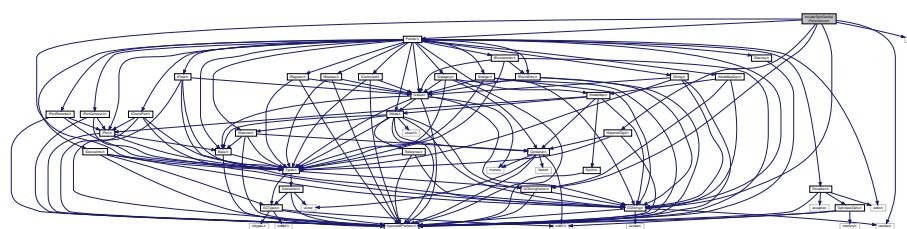
- Spinnaker
- Spinnaker::GenApi

## Functions

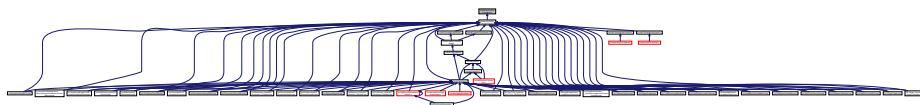
- **SPINNAKER\_API** IDestroy \* **CastToIDestroy** (INodeMap \*pNodeMap)  
*makes sure the dynamic\_cast operator is implemented in the DLL (due to a Linux bug)*
- template<class TCameraParams >  
**void \_LoadXMLFromFile** (const GenICam::gcstring &FileName)
- template<class TCameraParams >  
**void \_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** ()

## 15.119 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

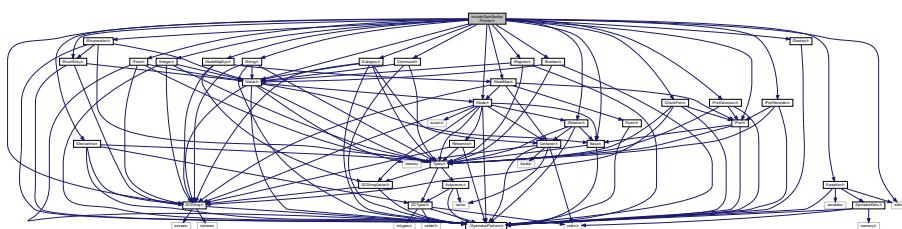
- virtual void [PersistFeature](#) (IValue &item)=0  
*Stores a feature.*
- [SPINNAKER\\_API std::istream & EatComments](#) (std::istream &is)  
*Helper function ignoring lines starting with comment character '#'.*
- [SPINNAKER\\_API std::istream & operator>>](#) (std::istream &is, CFeatureBag &FeatureBag)  
*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

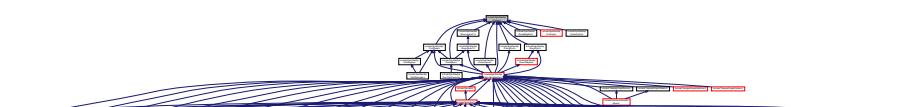
- interface [SPINNAKER\\_API\\_ABSTRACT IPersistScript](#)  
*Basic interface to persist values to.*

## 15.120 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.*

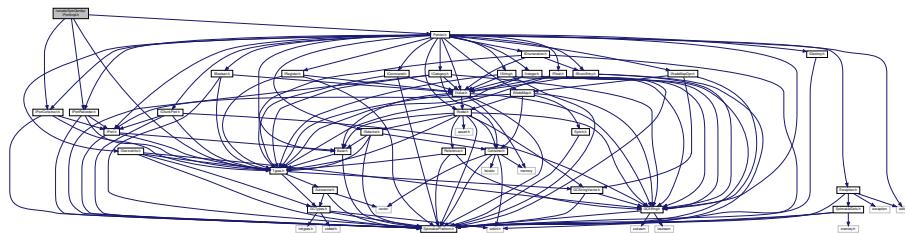
- **typedef CPointer< ISelector > CSelectorPtr**  
*SmartPointer for ISelector interface pointer.*
- **typedef CPointer< ICommand > CCommandPtr**  
*SmartPointer for ICommand interface pointer.*
- **typedef CPointer< IPotConstruct > CPortConstructPtr**  
*SmartPointer for IPotConstruct interface pointer.*

## Functions

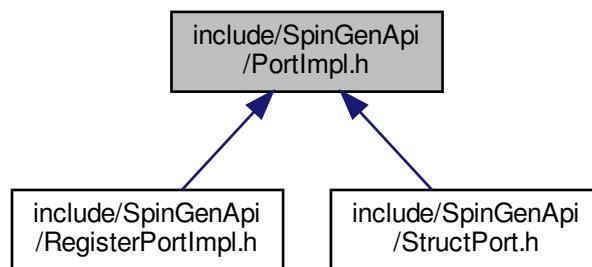
- **template<class T , class B >**  
**bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)**  
*Checks if a node is readable.*
- **template<class T , class B >**  
**bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)**  
*Checks if a node is Writable.*
- **template<class T , 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.*

## 15.121 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](#)

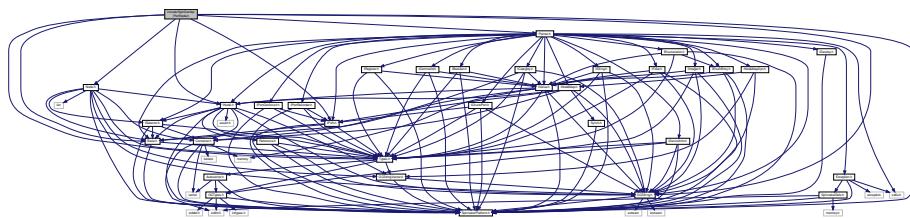
*Standard implementation for a port.*

## Namespaces

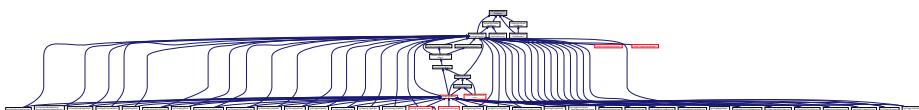
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.122 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

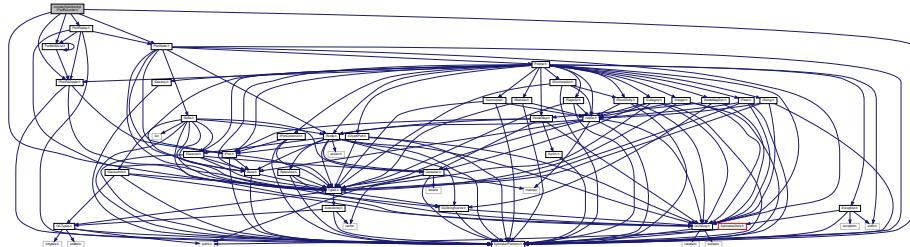
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

- typedef PortNode [CPortRef](#)

## 15.123 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



### Classes

- class [PortRecorder](#)  
*Interface for recording write commands on a port.*

### Namespaces

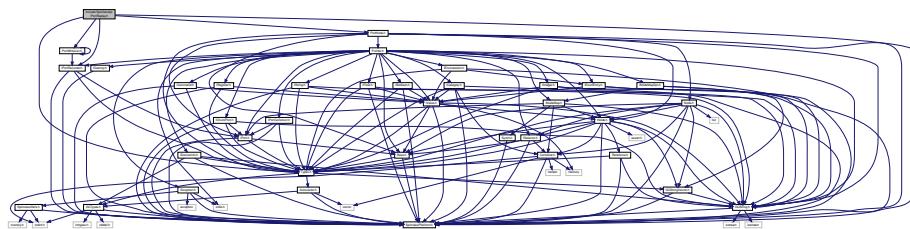
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

### Typedefs

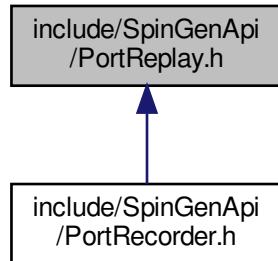
- typedef PortRecorder [CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*

## 15.124 include/SpinGenApi/PortReplay.h File Reference

Include dependency graph for PortReplay.h:



This graph shows which files directly or indirectly include this file:



## Classes

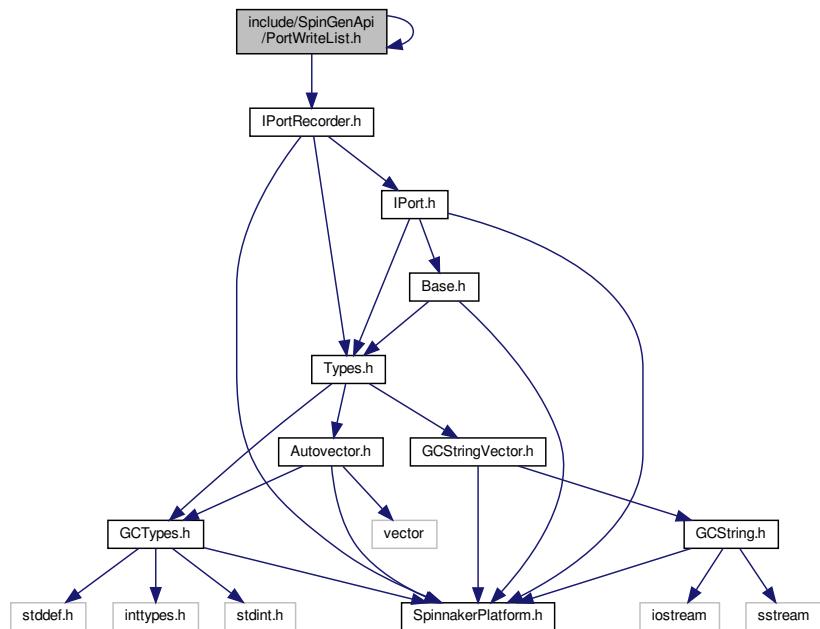
- class [PortReplay](#)  
*Interface for replaying write commands on a port.*

## Namespaces

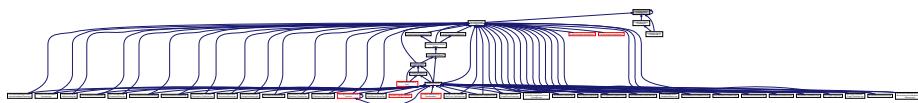
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.125 include/SpinGenApi/PortWriteList.h File Reference

Include dependency graph for PortWriteList.h:



This graph shows which files directly or indirectly include this file:



## Classes

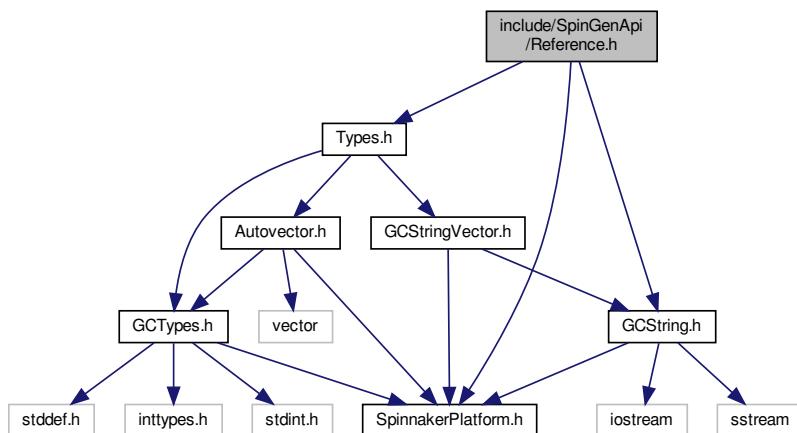
- class [CPortWriteList](#)  
*Container holding a list of port write commands.*

## Namespaces

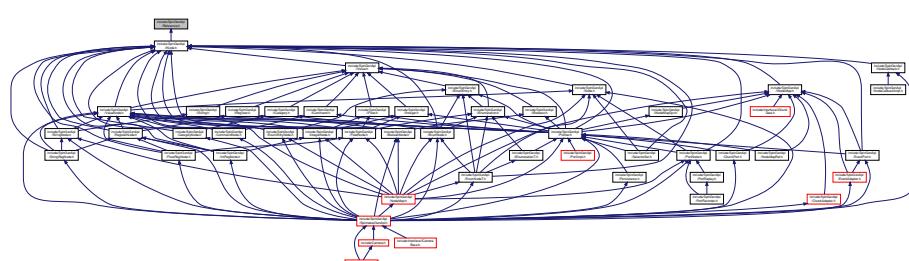
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.126 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



## Namespaces

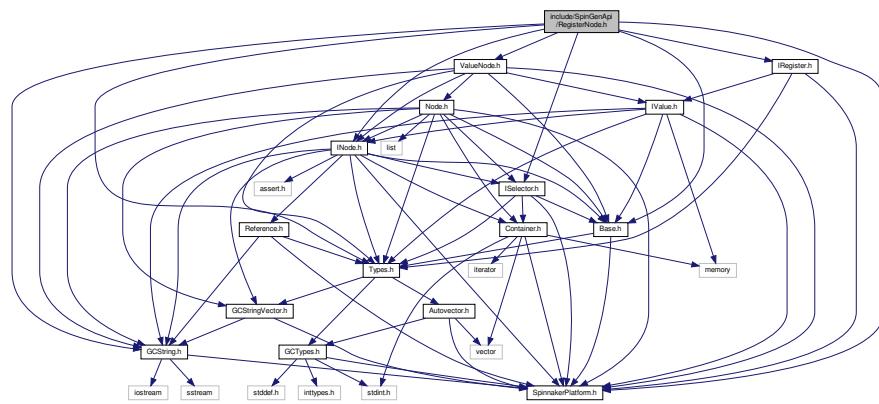
- Spinnaker
- Spinnaker::GenApi

## Functions

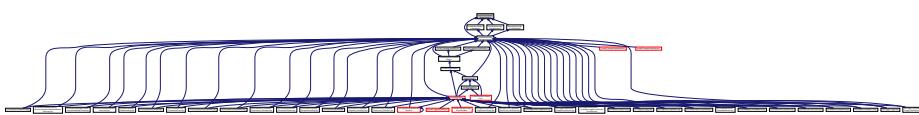
- virtual void [SetNumEnums](#) (int NumEnums)=0  
*sets the number of enum values*

## 15.127 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

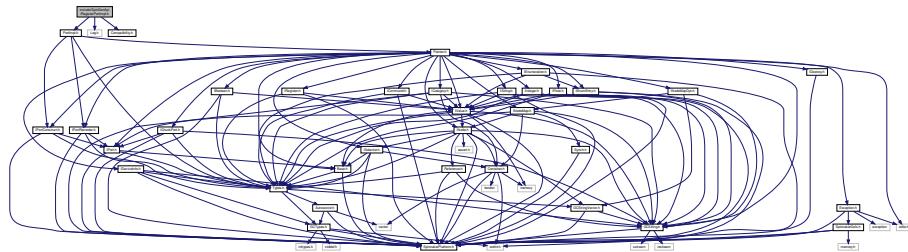
- Spinnaker
- Spinnaker::GenApi

## Typedefs

- `typedef RegisterNode CRegisterRef`

## 15.128 include/SpinGenApi/RegisterPortImpl.h File Reference

Include dependency graph for RegisterPortImpl.h:



## Classes

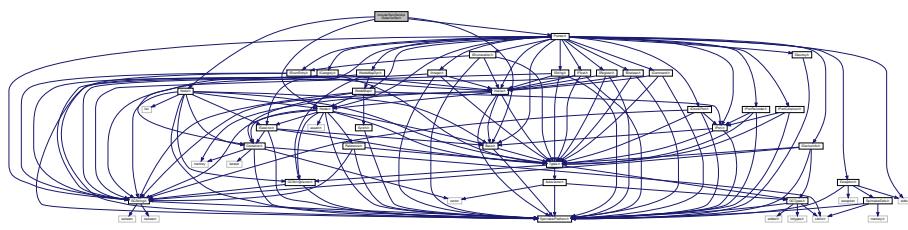
- `class CRegisterPortImpl`  
*Standard implementation for a port using a register based transport layer.*

## Namespaces

- `Spinnaker`
- `Spinnaker::GenApi`

## 15.129 include/SpinGenApi/SelectorSet.h File Reference

Include dependency graph for SelectorSet.h:



## Classes

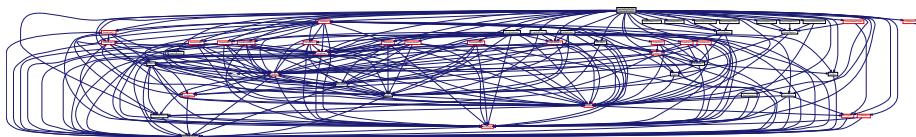
- `class CSelectorSet`  
*The set of selectors selecting a given node.*

## Namespaces

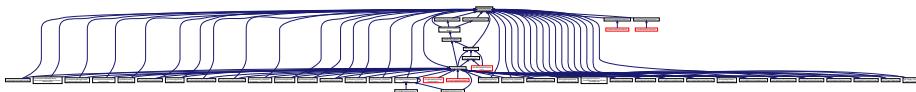
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.130 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

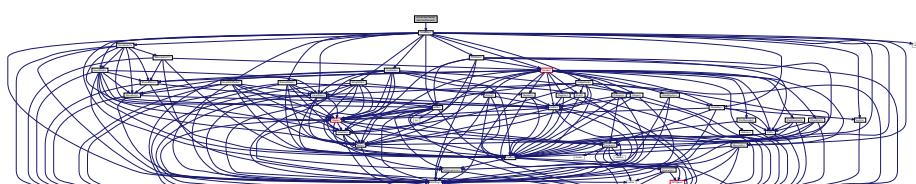


This graph shows which files directly or indirectly include this file:



## 15.131 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



## Classes

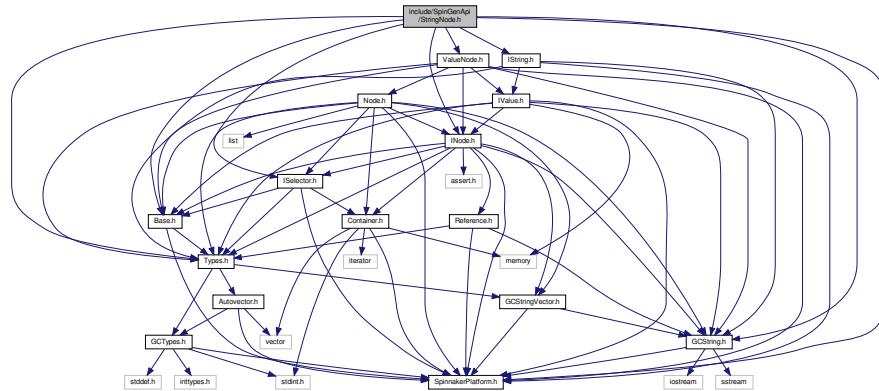
- class [SpinTestCamera](#)

## Namespaces

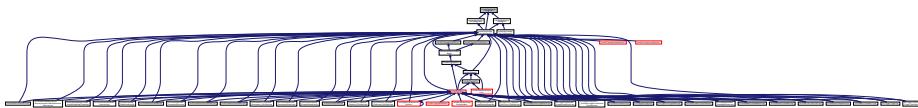
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.132 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StringNode](#)  
*Interface for string properties.*

## Namespaces

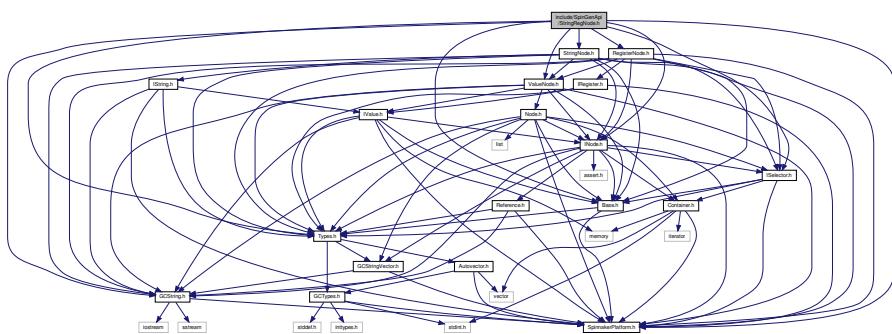
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## TypeDefs

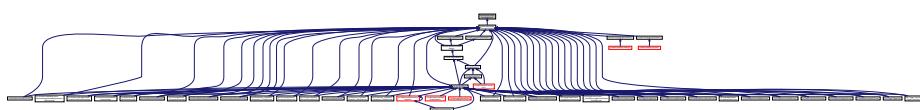
- typedef [StringNode](#) [CStringRef](#)

## 15.133 include/SpinGenApi/StringRegNode.h File Reference

Include dependency graph for StringRegNode.h:



This graph shows which files directly or indirectly include this file:



## Classes

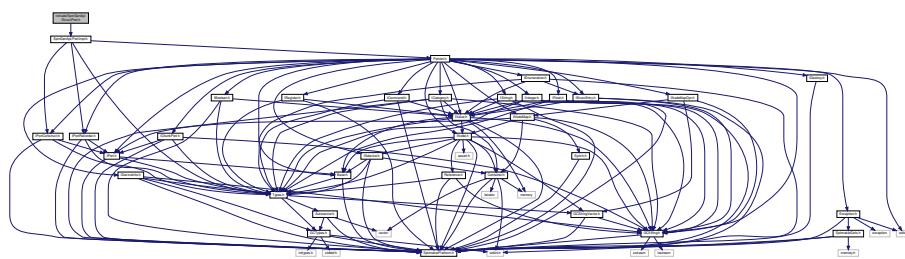
- class `StringRegNode`  
*Interface for string properties.*

## Namespaces

- Spinnaker
  - Spinnaker::GenApi

15.134 include/SpinGenApi/StructPort.h File Reference

Include dependency graph for StructPort.h:



## Classes

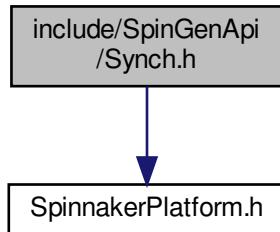
- class [CTestPortStruct< CDataStruct >](#)  
*Implements a register spaces based on a C++ struct.*

## Namespaces

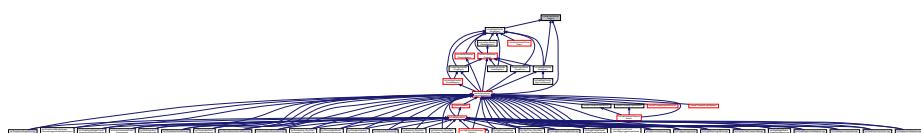
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.135 include/SpinGenApi/Synch.h File Reference

Include dependency graph for Synch.h:



This graph shows which files directly or indirectly include this file:



## Classes

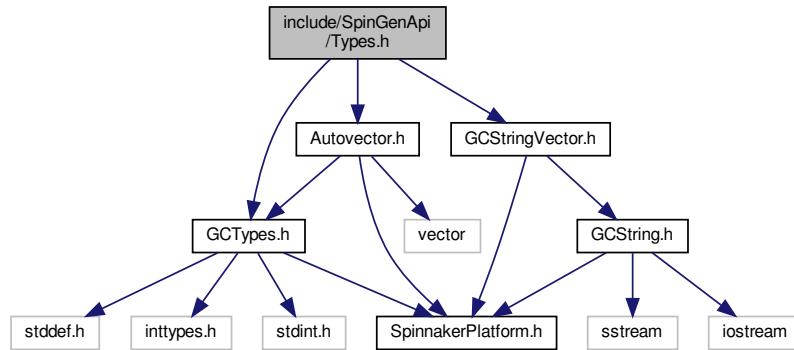
- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)

## Namespaces

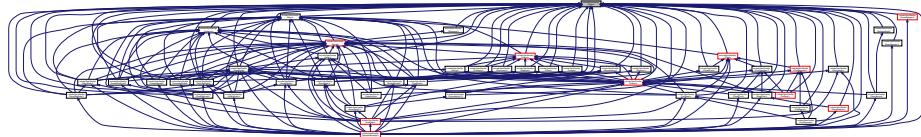
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## 15.136 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` = 0x8000000L }

*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 `_EIInterfaceType` {  
  `intfIValue`,  
  `intfIBase`,  
  `intfIInteger`,  
  `intfIBoolean`,  
  `intfICommand`,  
  `intfIFloat`,  
  `intfIString`,  
  `intfIRegister`,  
  `intfICategory`,  
  `intfIEnumeration`,  
  `intfIEnumEntry`,  
  `intfIPort` }

*typedef for interface type*

- enum `_ELinkType` {  
  `ctParentNodes`,  
  `ctReadingChildren`,  
  `ctWritingChildren`,  
  `ctlInvalidatingChildren`,  
  `ctDependingNodes`,  
  `ctTerminalNodes` }

*typedef for link type*

- enum `_EIncMode` {  
  `nolIncrement`,  
  `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.

```

## 15.136.1 Macro Definition Documentation

### 15.136.1.1 `_UndefinedRepresentation`

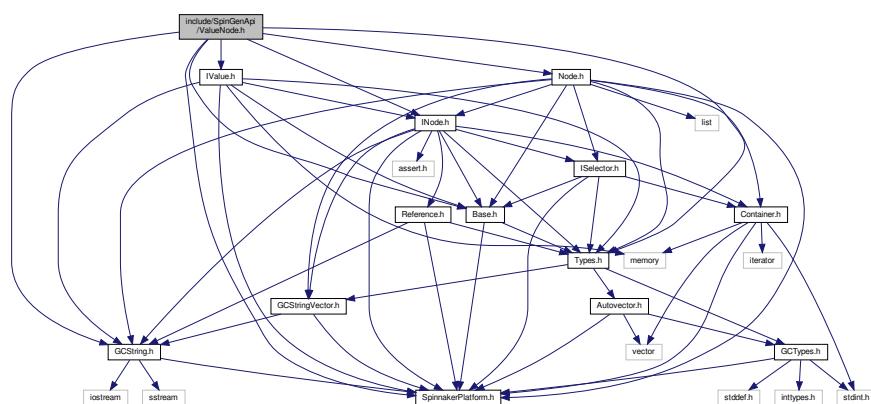
```
#define _UndefinedRepresentation _UndefinedRepresentation
```

### 15.136.1.2 `interface`

```
#define interface struct
```

## 15.137 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

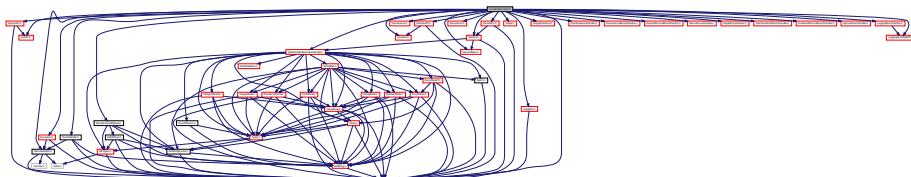
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

## Typedefs

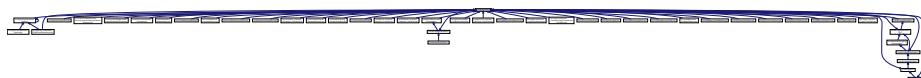
- typedef ValueNode [CValueRef](#)

## 15.138 include/Spinnaker.h File Reference

Include dependency graph for Spinnaker.h:

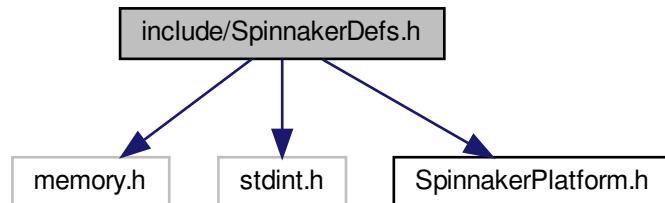


This graph shows which files directly or indirectly include this file:

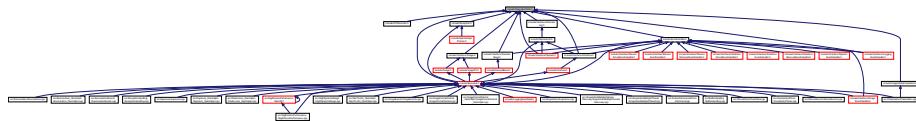


## 15.139 include/SpinnakerDefs.h File Reference

Include dependency graph for SpinnakerDefs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [PNGOption](#)  
*Options for saving PNG images.*
- 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 [LibraryVersion](#)  
*Provides easier access to the current version of Spinnaker.*
- struct [ActionCommandResult](#)  
*Action Command Result.*

## 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,
GENICAM_ERR_INVALID_ARGUMENT = -2001,
GENICAM_ERR_OUT_OF_RANGE = -2002,
GENICAM_ERR_PROPERTY = -2003,
GENICAM_ERR_RUN_TIME = -2004,
GENICAM_ERR_LOGICAL = -2005,
GENICAM_ERR_ACCESS = -2006,
GENICAM_ERR_TIMEOUT = -2007,
GENICAM_ERR_DYNAMIC_CAST = -2008,
GENICAM_ERR_GENERIC = -2009,
GENICAM_ERR_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_LOGGING_EVENT,
SPINNAKER_EVENT_UNKNOWN,
SPINNAKER_EVENT_INTERFACE_ARRIVAL_REMOVAL }
```

*Event types in Spinnaker.*

- enum PixelFormatNamespaceID {

```
SPINNAKER_PIXELFORMAT_NAMESPACE_UNKNOWN = 0,
SPINNAKER_PIXELFORMAT_NAMESPACE_GEV = 1,
SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC = 2,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT = 3,
SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT = 4,
SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID = 1000 }
```

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

- enum ColorProcessingAlgorithm {

```
DEFAULT,
NO_COLOR_PROCESSING,
NEAREST_NEIGHBOR,
NEAREST_NEIGHBOR_AVG,
BILINEAR,
EDGE_SENSING,
HQ_LINEAR,
IPP,
DIRECTIONAL_FILTER,
RIGOROUS,
WEIGHTED_DIRECTIONAL_FILTER }
```

*Color processing algorithms.*

- enum `ImageFileFormat` {  
  `FROM_FILE_EXT` = -1,  
  `PGM`,  
  `PPM`,  
  `BMP`,  
  `JPEG`,  
  `JPEG2000`,  
  `TIFF`,  
  `PNG`,  
  `RAW`,  
  `JPEG12_C`,  
  `IMAGE_FILE_FORMAT_FORCE_32BITS` = 0xFFFFFFFF }  
}

*File formats to be used for saving images to disk.*

- enum `ImageStatus` {  
  `IMAGE_UNKNOWN_ERROR` = -1,  
  `IMAGE_NO_ERROR` = 0,  
  `IMAGE_CRC_CHECK_FAILED` = 1,  
  `IMAGE_DATA_OVERFLOW` = 2,  
  `IMAGE_MISSING_PACKETS` = 3,  
  `IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT` = 4,  
  `IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT` = 5,  
  `IMAGE_PACKETID_INCONSISTENT` = 6,  
  `IMAGE_MISSING_LEADER` = 7,  
  `IMAGE_MISSING_TRAILER` = 8,  
  `IMAGE_DATA_INCOMPLETE` = 9,  
  `IMAGE_INFO_INCONSISTENT` = 10,  
  `IMAGE_CHUNK_DATA_INVALID` = 11,  
  `IMAGE_NO_SYSTEM_RESOURCES` = 12 }  
}

*Status of images returned from `GetNextImage()` call.*

- enum `StatisticsChannel` {  
  `GREY`,  
  `RED`,  
  `GREEN`,  
  `BLUE`,  
  `HUE`,  
  `SATURATION`,  
  `LIGHTNESS`,  
  `NUM_STATISTICS_CHANNELS` }  
}

*Channels that allow statistics to be calculated.*

- enum `SpinnakerLogLevel` {  
  `LOG_LEVEL_OFF` = -1,  
  `LOG_LEVEL_FATAL` = 0,  
  `LOG_LEVEL_ALERT` = 100,  
  `LOG_LEVEL_CRIT` = 200,  
  `LOG_LEVEL_ERROR` = 300,  
  `LOG_LEVEL_WARN` = 400,  
  `LOG_LEVEL_NOTICE` = 500,  
  `LOG_LEVEL_INFO` = 600,  
  `LOG_LEVEL_DEBUG` = 700,  
  `LOG_LEVEL_NOTSET` = 800 }  
}

*log levels*

- enum `PayloadTypeInfoIds` {  
  `PAYLOAD_TYPE_UNKNOWN` = 0,  
  `PAYLOAD_TYPE_IMAGE` = 1,  
  `PAYLOAD_TYPE_RAW_DATA` = 2,  
  `PAYLOAD_TYPE_FILE` = 3,  
    
}

- ```

PAYLOAD_TYPE_CHUNK_DATA = 4,
PAYLOAD_TYPE_JPEG = 5,
PAYLOAD_TYPE_JPEG2000 = 6,
PAYLOAD_TYPE_H264 = 7,
PAYLOAD_TYPE_CHUNK_ONLY = 8,
PAYLOAD_TYPE_DEVICE_SPECIFIC = 9,
PAYLOAD_TYPE_MULTI_PART = 10,
PAYLOAD_TYPE_CUSTOM_ID = 1000,
PAYLOAD_TYPE_EXTENDED_CHUNK = 1001 }

• enum ActionCommandStatus {
ACTION_COMMAND_STATUS_OK = 0,
ACTION_COMMAND_STATUS_NO_REF_TIME,
ACTION_COMMAND_STATUS_OVERFLOW = 0x8015,
ACTION_COMMAND_STATUS_ACTION_LATE,
ACTION_COMMAND_STATUS_ERROR }

    Possible Status Codes Returned from Action Command.

• enum PixelFormatIntType {
IntType_UINT8,
IntType_INT8,
IntType_UINT10,
IntType_UINT10p,
IntType_UINT10P,
IntType_UINT12,
IntType_UINT12p,
IntType_UINT12P,
IntType_UINT14,
IntType_UINT16,
IntType_INT16,
IntType_FLOAT32,
IntType_UNKNOWN }

    Possible integer types and packing used in a pixel format.

• enum BufferOwnership {
BUFFER_OWNERSHIP_SYSTEM,
BUFFER_OWNERSHIP_USER }

```

## 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

## 15.140 include/SpinnakerPlatform.h File Reference

### Macros

- #define SPINNAKER\_API\_ABSTRACT /\*nothing\*/
- #define SPINNAKER\_API \_\_attribute\_\_((visibility("default")))
- #define SPINNAKER\_LOCAL \_\_attribute\_\_((visibility("hidden"))))

### 15.140.1 Macro Definition Documentation

### 15.140.1.1 SPINNAKER\_API

```
#define SPINNAKER_API __attribute__((visibility("default")))
```

### 15.140.1.2 SPINNAKER\_API\_ABSTRACT

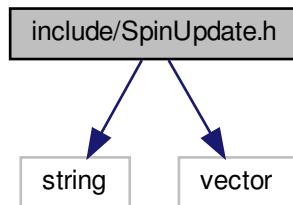
```
#define SPINNAKER_API_ABSTRACT /*nothing*/
```

### 15.140.1.3 SPINNAKER\_LOCAL

```
#define SPINNAKER_LOCAL __attribute__((visibility("hidden")))
```

## 15.141 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](#) ([UpdatorMessageCallback](#) messageCallbackFunction)
- SPINUPDATE\_API void [SetProgressCallback](#) ([UpdatorProgressCallback](#) progressCallbackFunction)
- const SPINUPDATE\_API char \* [GetErrorMessage](#) ()

## Variables

- `SPINUPDATE_API` `typedef int(* UpdatorMessageCallback )(const char *message)`
- `SPINUPDATE_API` `typedef int(* UpdatorProgressCallback )(const char *action, unsigned int address, int globalPercent, int currPercent)`

## 15.141.1 Macro Definition Documentation

### 15.141.1.1 SPINUPDATE\_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```

## 15.141.2 Function Documentation

### 15.141.2.1 GetErrorMessage()

```
const SPINUPDATE_API char* GetErrorMessage ( )
```

### 15.141.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (
    UpdatorMessageCallback messageCallbackFunction )
```

### 15.141.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (
    UpdatorProgressCallback progressCallbackFunction )
```

### 15.141.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
    std::vector< std::string > args )
```

### 15.141.2.5 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 -AFF080000 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 EEPROM to a file.

### 15.141.2.6 UpdateFirmwareGUI()

```
SPINUPDATE_API int UpdateFirmwareGUI (
    std::string args )
```

## 15.141.3 Variable Documentation

### 15.141.3.1 UpdatorMessageCallback

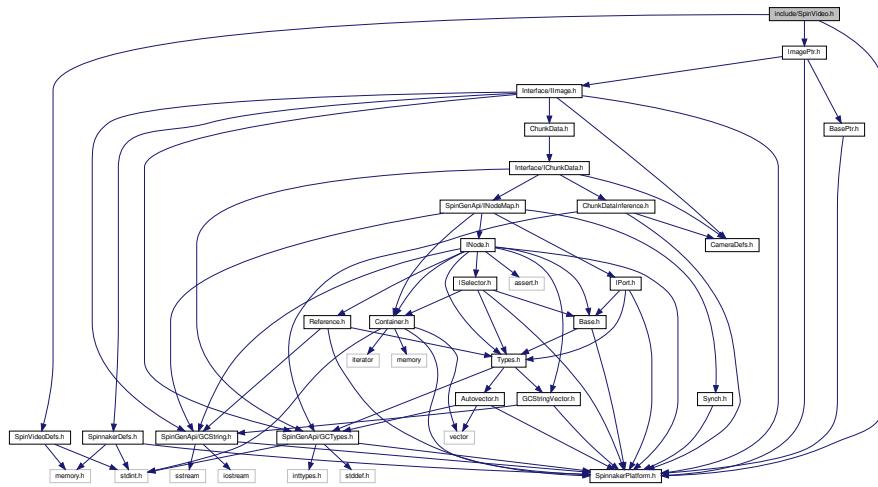
```
SPINUPDATE_API typedef int(* UpdatorMessageCallback) (const char *message)
```

### 15.141.3.2 UpdatorProgressCallback

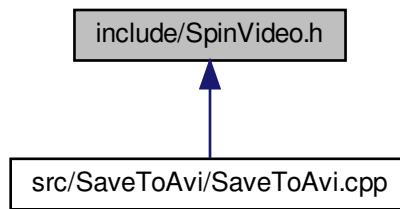
```
SPINUPDATE_API typedef int( * UpdatorProgressCallback) (const char *action, unsigned int address,
int globalPercent, int currPercent)
```

## 15.142 include/SpinVideo.h File Reference

Include dependency graph for SpinVideo.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [SpinVideo](#)

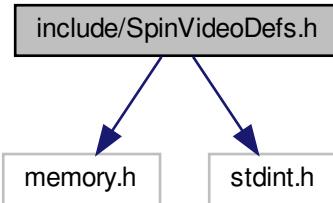
*Provides the functionality for the user to record images to an AVI/MP4 file.*

## Namespaces

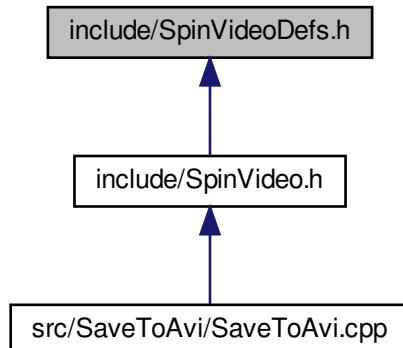
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 15.143 include/SpinVideoDefs.h File Reference

Include dependency graph for SpinVideoDefs.h:



This graph shows which files directly or indirectly include this file:



### Classes

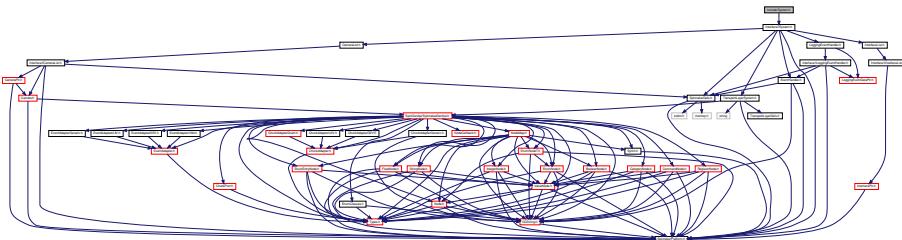
- struct [MJPEGOption](#)  
*Options for saving MJPG files.*
- struct [H264Option](#)  
*Options for saving H264 files.*
- struct [AVIOption](#)  
*Options for saving AVI files.*

### Namespaces

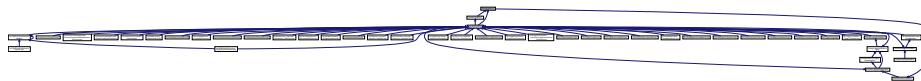
- [Spinnaker](#)
- [Spinnaker::Video](#)

## 15.144 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 2
- #define FLIR\_SPINNAKER\_VERSION\_MINOR 2
- #define FLIR\_SPINNAKER\_VERSION\_TYPE 0
- #define FLIR\_SPINNAKER\_VERSION\_BUILD 48

#### 15.144.1 Macro Definition Documentation

##### 15.144.1.1 FLIR\_SPINNAKER\_VERSION\_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 48
```

### 15.144.1.2 FLIR\_SPINNAKER\_VERSION\_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 2
```

### 15.144.1.3 FLIR\_SPINNAKER\_VERSION\_MINOR

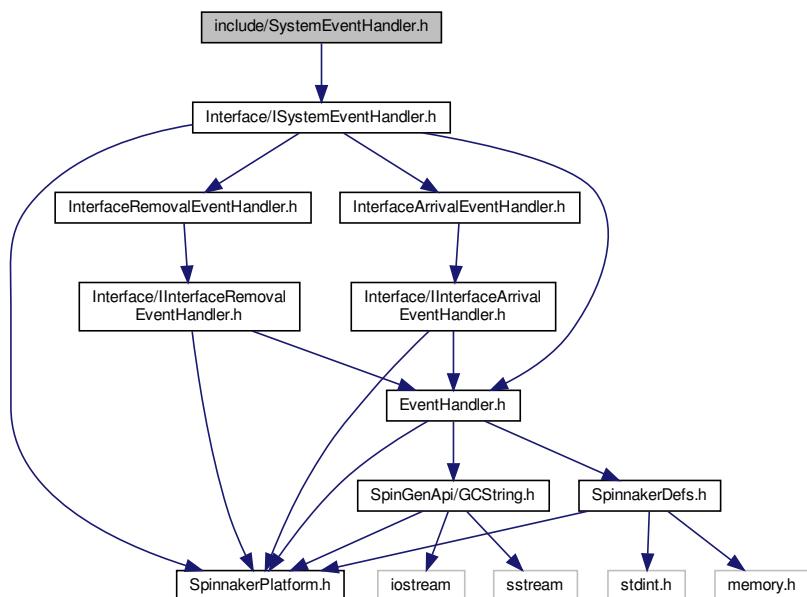
```
#define FLIR_SPINNAKER_VERSION_MINOR 2
```

### 15.144.1.4 FLIR\_SPINNAKER\_VERSION\_TYPE

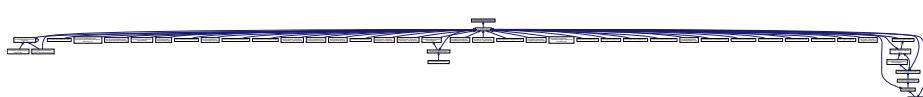
```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

## 15.145 include/SystemEventHandler.h File Reference

Include dependency graph for SystemEventHandler.h:



This graph shows which files directly or indirectly include this file:



## Classes

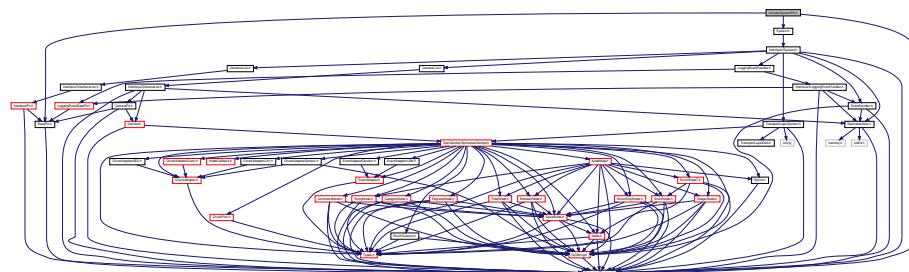
- class [SystemEventHandler](#)  
*A handler to interface arrival and removal events on the system.*

## Namespaces

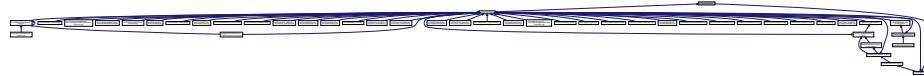
- [Spinnaker](#)

## 15.146 include/SystemPtr.h File Reference

Include dependency graph for SystemPtr.h:



This graph shows which files directly or indirectly include this file:



## Classes

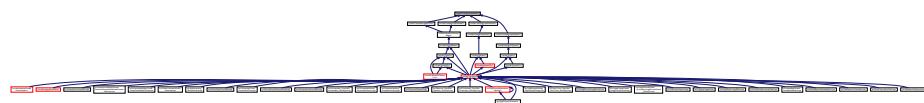
- class [SystemPtr](#)  
*A reference tracked pointer to a system object.*

## Namespaces

- [Spinnaker](#)

## 15.147 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



## Namespaces

- Spinnaker

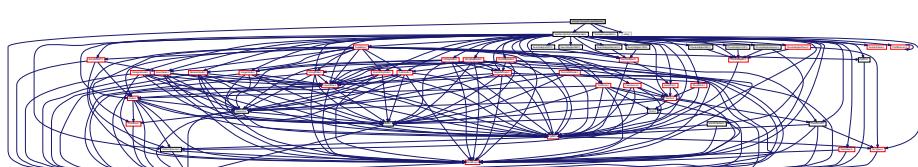
## Enumerations

- 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 StreamBufferCountModeEnum {  
    StreamBufferCountMode\_Manual,  
    StreamBufferCountMode\_Auto,  
    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 GevCCPEnum {  
    GevCCP\_EnumEntry\_GevCCP\_OpenAccess,  
    GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess,  
    GevCCP\_EnumEntry\_GevCCP\_ControlAccess,  
    NUMGEVCCP }
- enum GUIXMLLocationEnum {  
    GUIXMLLocation\_Device,  
    GUIXMLLocation\_Host,  
    NUMGUIXMLLOCATION }
- enum GenICamXMLLocationEnum {  
    GenICamXMLLocation\_Device,  
    GenICamXMLLocation\_Host,  
    NUMGENICAMXMLLOCATION }

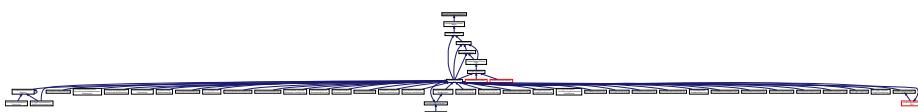
- enum `DeviceEndianessMechanismEnum` {
 `DeviceEndianessMechanism_Legacy`,
 `DeviceEndianessMechanism_Standard`,
 `NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {
 `DeviceCurrentSpeed_UnknownSpeed`,
 `DeviceCurrentSpeed_LowSpeed`,
 `DeviceCurrentSpeed_FullSpeed`,
 `DeviceCurrentSpeed_HighSpeed`,
 `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` }

## 15.148 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](#)

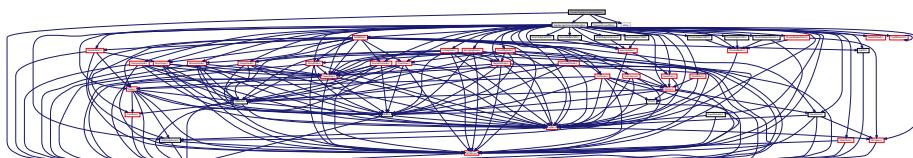
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

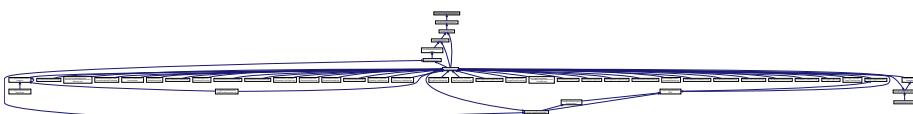
- [Spinnaker](#)

## 15.149 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](#)

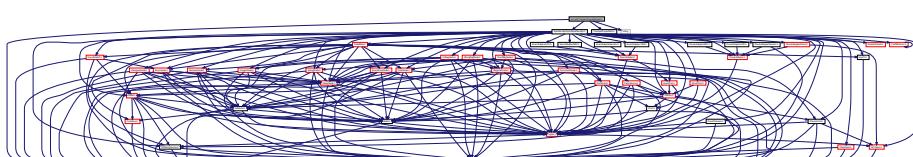
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

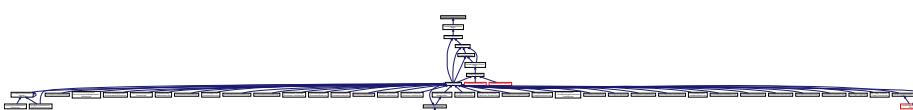
- [Spinnaker](#)

## 15.150 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](#)

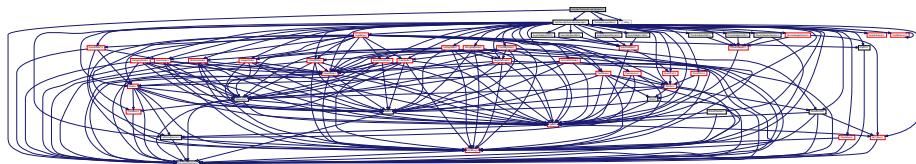
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

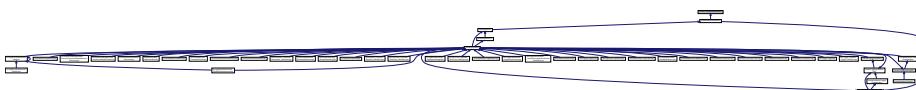
- [Spinnaker](#)

## 15.151 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](#)

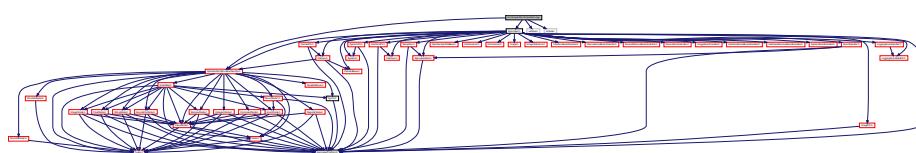
*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

## Namespaces

- [Spinnaker](#)

## 15.152 src/Acquisition/Acquisition.cpp File Reference

Include dependency graph for Acquisition.cpp:



## Functions

- int [AcquireImages](#) (`CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice`)
- int [PrintDeviceInfo](#) (`INodeMap &nodeMap`)
- int [RunSingleCamera](#) (`CameraPtr pCam`)
- int [main](#) (`int, char **`)

### 15.152.1 Function Documentation

#### 15.152.1.1 [AcquireImages\(\)](#)

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

#### 15.152.1.2 [main\(\)](#)

```
int main (
    int ,
    char ** )
```

#### 15.152.1.3 [PrintDeviceInfo\(\)](#)

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

#### 15.152.1.4 [RunSingleCamera\(\)](#)

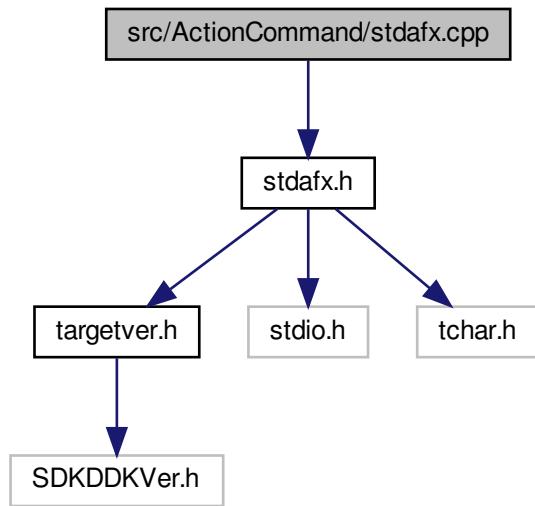
```
int RunSingleCamera (
    CameraPtr pCam )
```



- 15.153 **src/Acquisition/resource.h File Reference**
- 15.154 **src/AcquisitionMultipleCameraRecovery/resource.h File Reference**
- 15.155 **src/AcquisitionMultipleThread/resource.h File Reference**
- 15.156 **src/ActionCommand/resource.h File Reference**
- 15.157 **src/BufferHandling/resource.h File Reference**
- 15.158 **src/ChunkData/resource.h File Reference**
- 15.159 **src/CounterAndTimer/resource.h File Reference**
- 15.160 **src/DeviceEvents/resource.h File Reference**
- 15.161 **src/Enumeration/resource.h File Reference**
- 15.162 **src/Enumeration\_QuickSpin/resource.h File Reference**
- 15.163 **src/EnumerationEvents/resource.h File Reference**
- 15.164 **src/ExceptionHandling/resource.h File Reference**
- 15.165 **src/Exposure/resource.h File Reference**
- 15.166 **src/Exposure\_QuickSpin/resource.h File Reference**
- 15.167 **src/FileAccess\_QuickSpin/resource.h File Reference**
- 15.168 **src/GigEVisionPerformance/resource.h File Reference**
- 15.169 **src/HighDynamicRange/resource.h File Reference**
- 15.170 **src/GenTLInfo\_QuickSpin/resource.h File Reference**
- 15.171 **src/ImageEvents/resource.h File Reference**
- 15.172 **src/ImageFormatControl/resource.h File Reference**
- 15.173 **src/ImageFormatControl\_QuickSpin/resource.h File Reference**
- 15.174 **src/Inference/resource.h File Reference**
- 15.175 **src/Logging/resource.h File Reference**

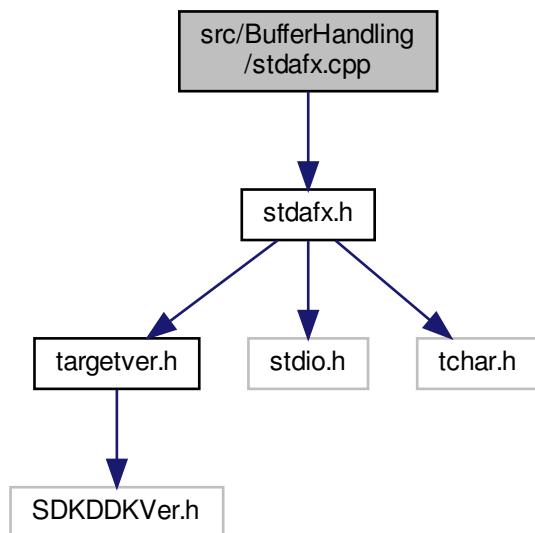
## 15.187 src/ActionCommand/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



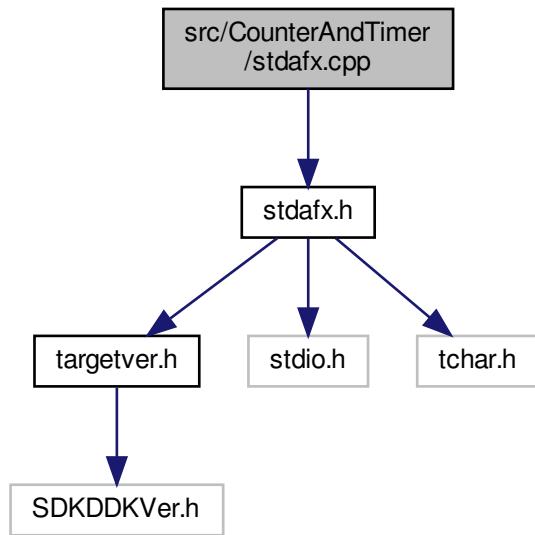
## 15.188 src/BufferHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



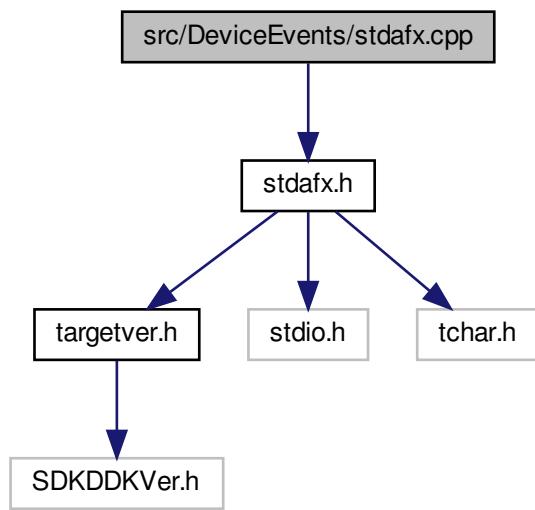
## 15.189 src/CounterAndTimer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



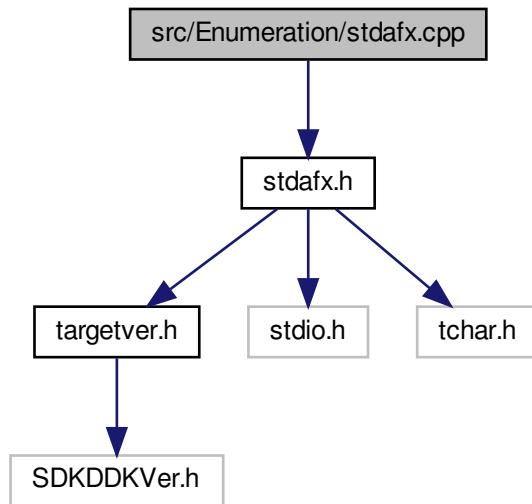
## 15.190 src/DeviceEvents/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



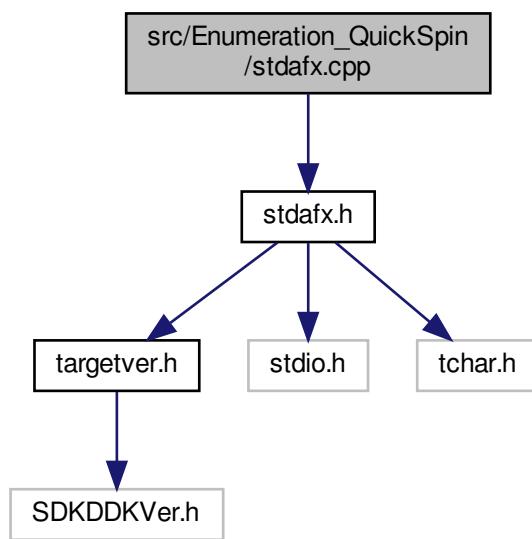
## 15.191 src/Enumeration/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



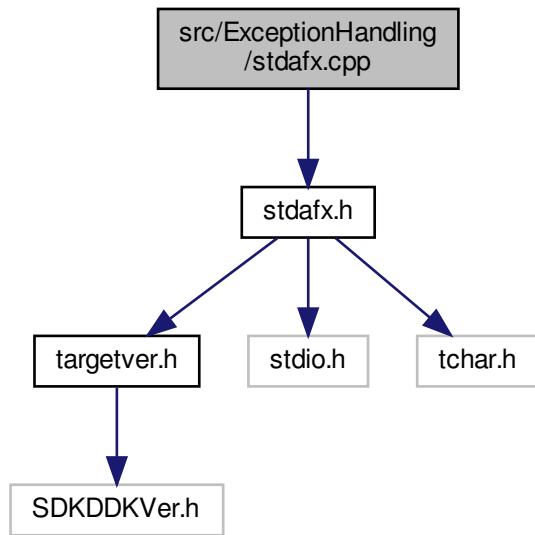
## 15.192 src/Enumeration\_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



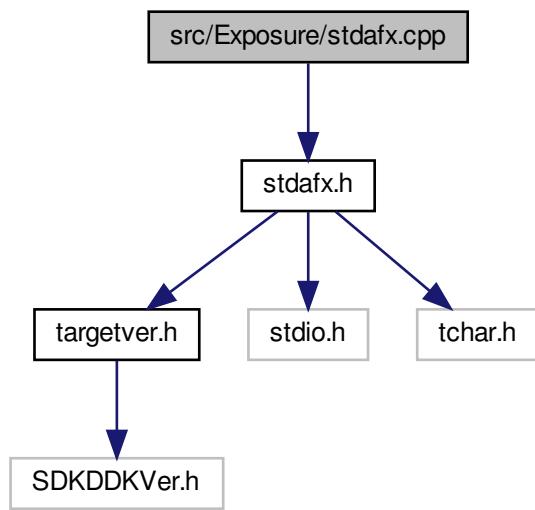
### 15.193 src/ExceptionHandling/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



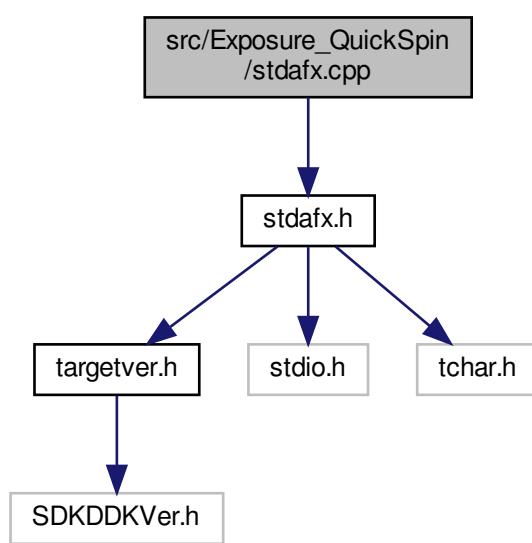
### 15.194 src/Exposure/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



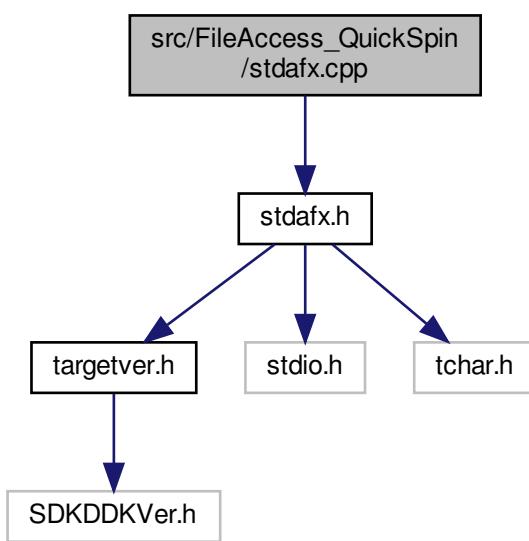
## 15.195 src/Exposure\_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



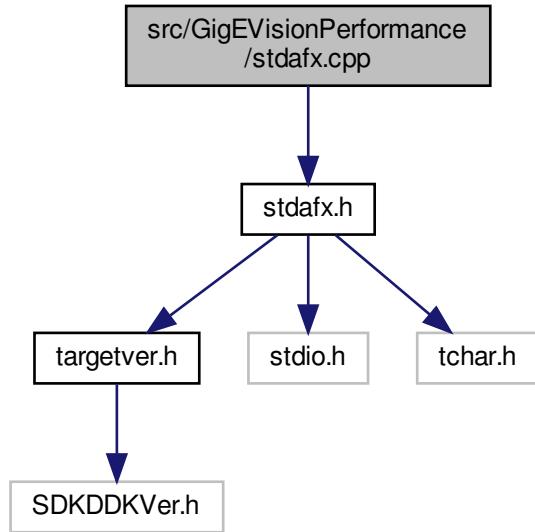
## 15.196 src/FileAccess\_QuickSpin/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



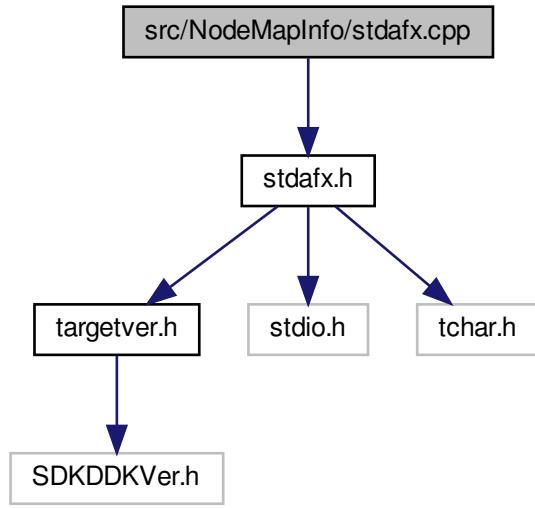
## 15.197 src/GigEVisionPerformance/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



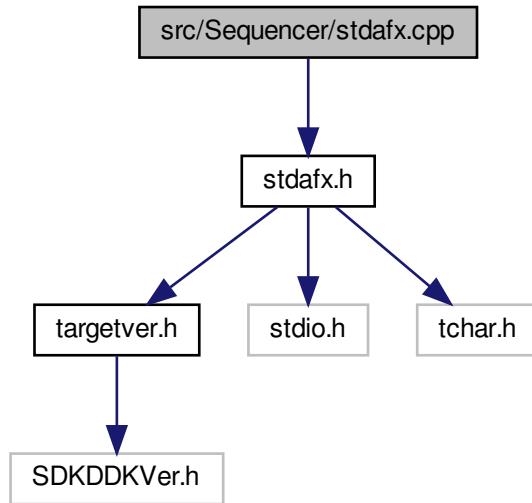
## 15.198 src/NodeMapInfo/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



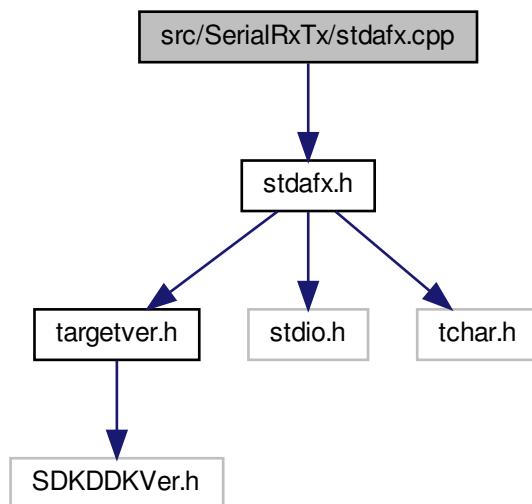
## 15.199 src/Sequencer/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:



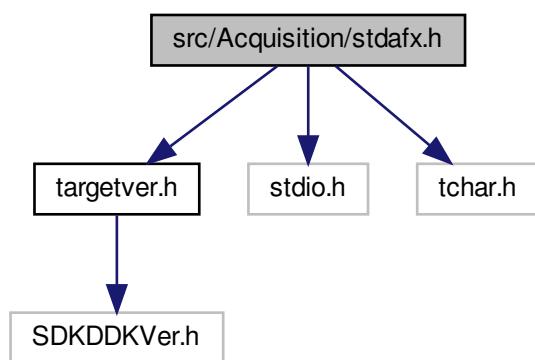
## 15.200 src/SerialRxTx/stdafx.cpp File Reference

Include dependency graph for stdafx.cpp:

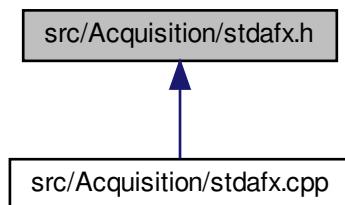


## 15.201 src/Acquisition/stdafx.h File Reference

Include dependency graph for stdafx.h:

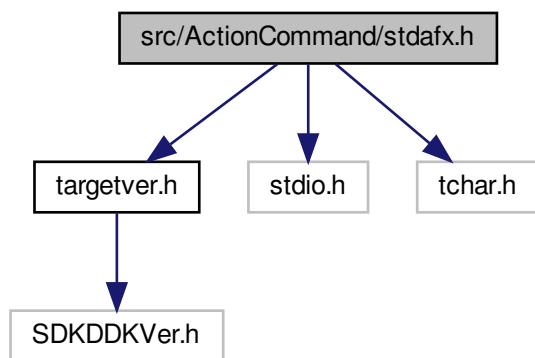


This graph shows which files directly or indirectly include this file:

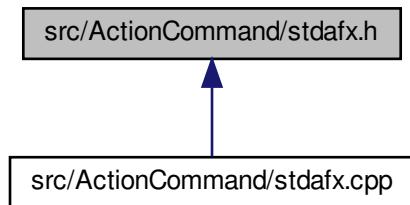


## 15.202 src/ActionCommand/stdafx.h File Reference

Include dependency graph for stdafx.h:

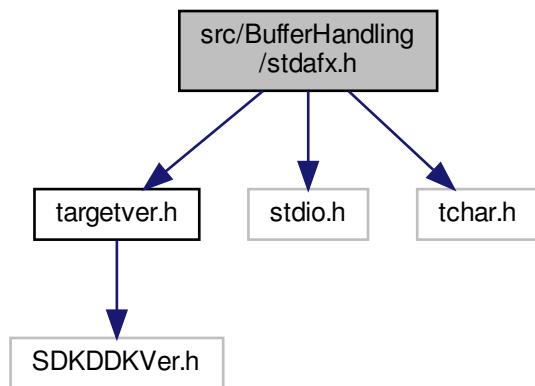


This graph shows which files directly or indirectly include this file:

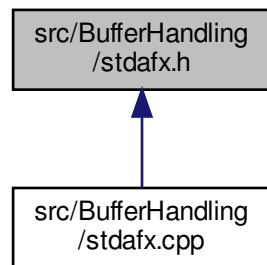


## 15.203 src/BufferHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

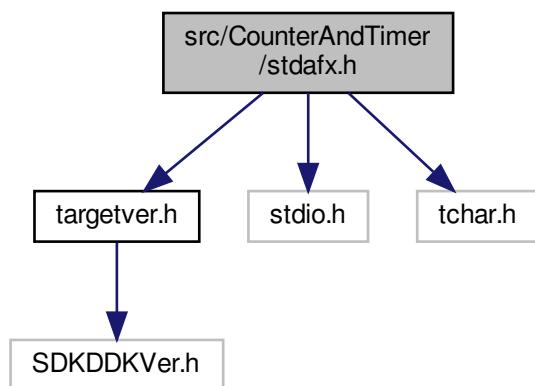


This graph shows which files directly or indirectly include this file:

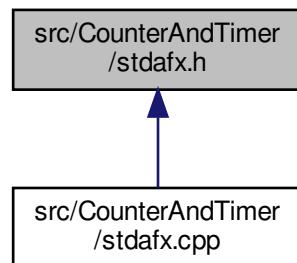


## 15.204 src/CounterAndTimer/stdafx.h File Reference

Include dependency graph for stdafx.h:

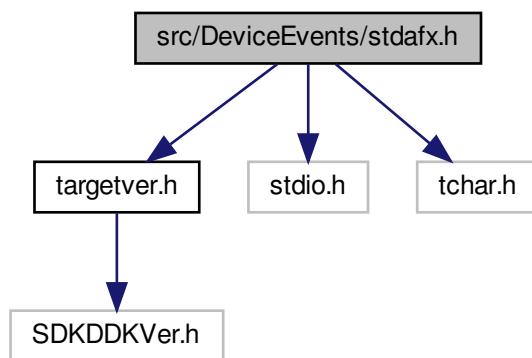


This graph shows which files directly or indirectly include this file:

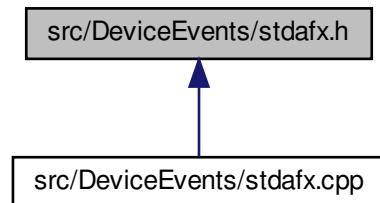


## 15.205 src/DeviceEvents/stdafx.h File Reference

Include dependency graph for stdafx.h:

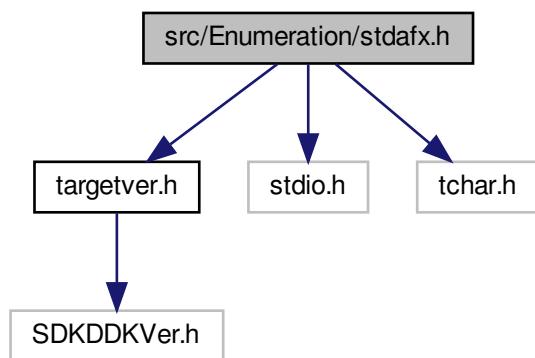


This graph shows which files directly or indirectly include this file:

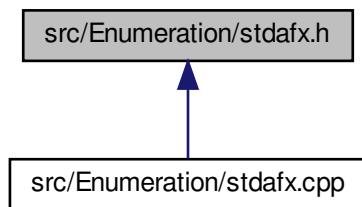


## 15.206 src/Enumeration/stdafx.h File Reference

Include dependency graph for stdafx.h:

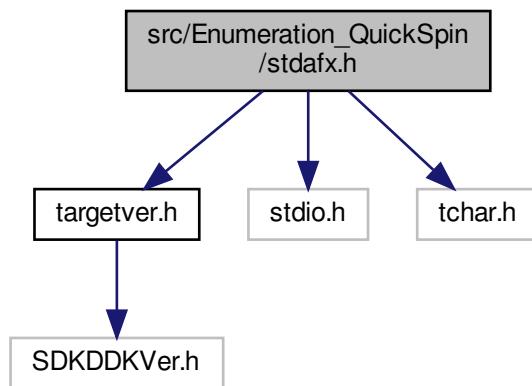


This graph shows which files directly or indirectly include this file:

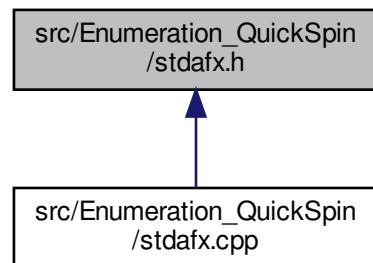


## 15.207 src/Enumeration\_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

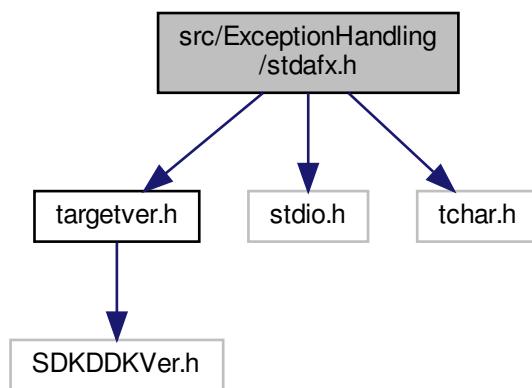


This graph shows which files directly or indirectly include this file:

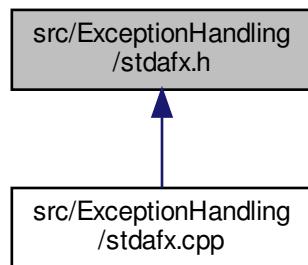


## 15.208 src/ExceptionHandling/stdafx.h File Reference

Include dependency graph for stdafx.h:

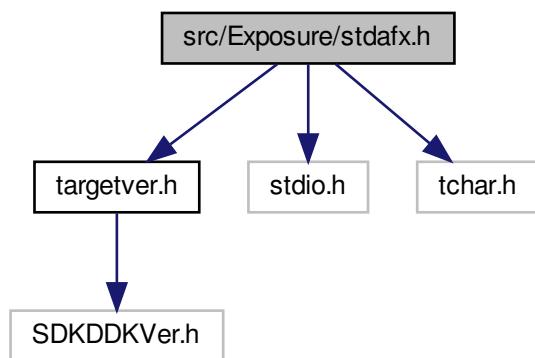


This graph shows which files directly or indirectly include this file:

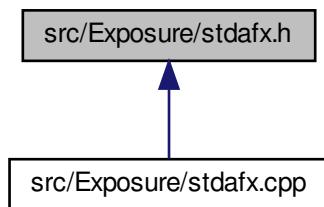


## 15.209 src/Exposure/stdafx.h File Reference

Include dependency graph for stdafx.h:

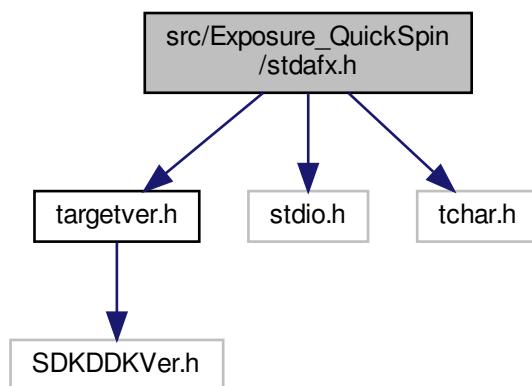


This graph shows which files directly or indirectly include this file:

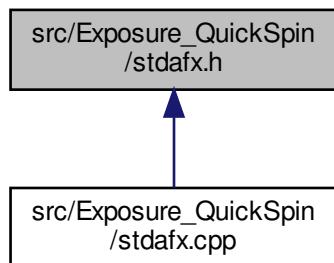


## 15.210 src/Exposure\_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

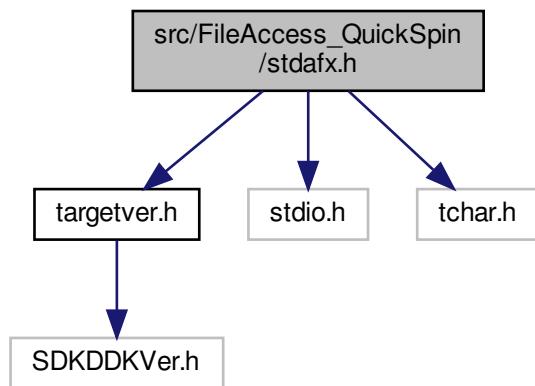


This graph shows which files directly or indirectly include this file:

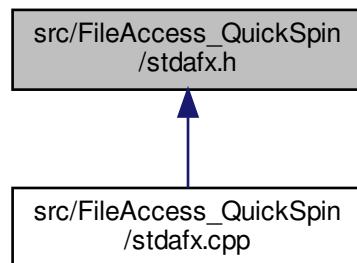


## 15.211 src/FileAccess\_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

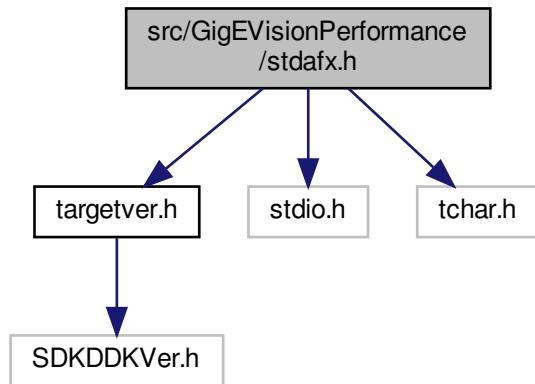


This graph shows which files directly or indirectly include this file:

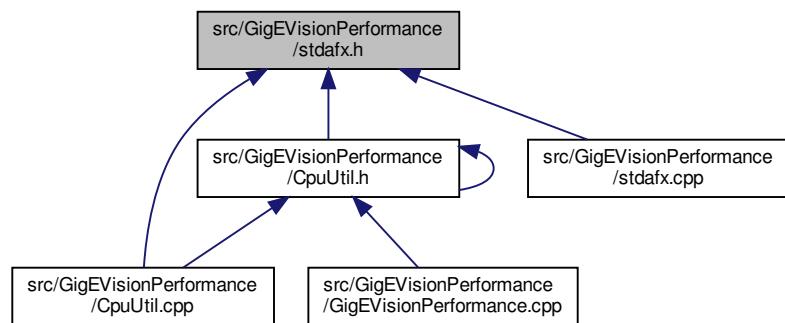


## 15.212 src/GigEVisionPerformance/stdafx.h File Reference

Include dependency graph for stdafx.h:

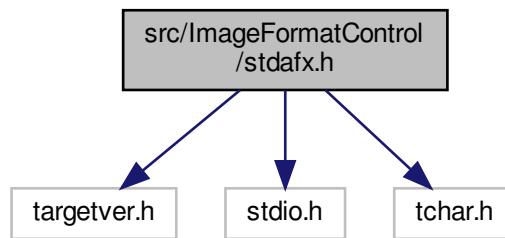


This graph shows which files directly or indirectly include this file:



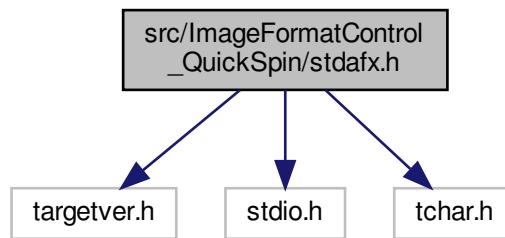
## 15.213 src/ImageFormatControl/stdafx.h File Reference

Include dependency graph for stdafx.h:



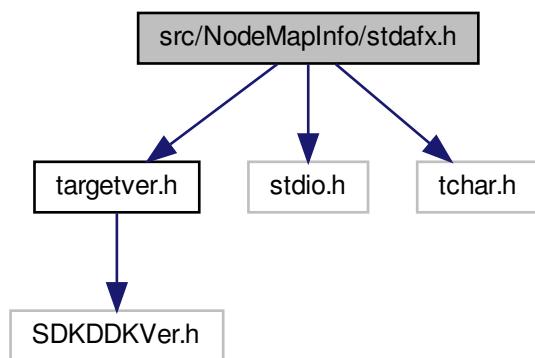
## 15.214 src/ImageFormatControl\_QuickSpin/stdafx.h File Reference

Include dependency graph for stdafx.h:

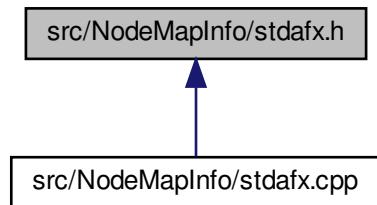


## 15.215 src/NodeMapInfo/stdafx.h File Reference

Include dependency graph for stdafx.h:

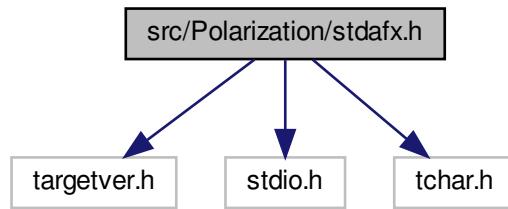


This graph shows which files directly or indirectly include this file:



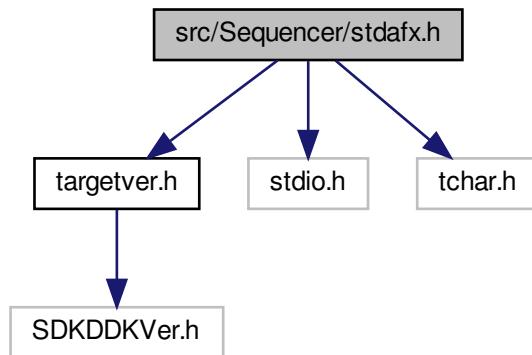
## 15.216 src/Polarization/stdafx.h File Reference

Include dependency graph for stdafx.h:

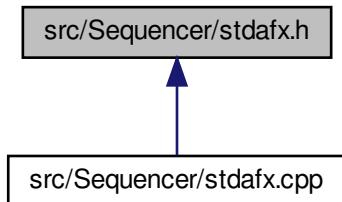


## 15.217 src/Sequencer/stdafx.h File Reference

Include dependency graph for stdafx.h:

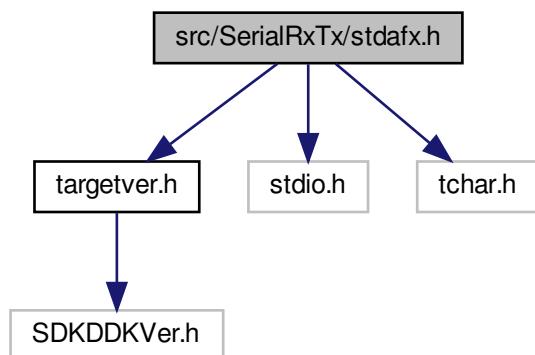


This graph shows which files directly or indirectly include this file:

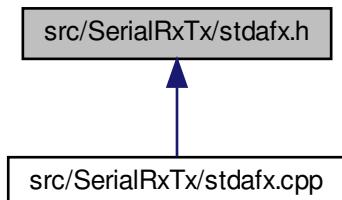


## 15.218 src/SerialRxTx/stdafx.h File Reference

Include dependency graph for stdafx.h:

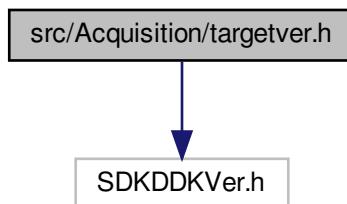


This graph shows which files directly or indirectly include this file:

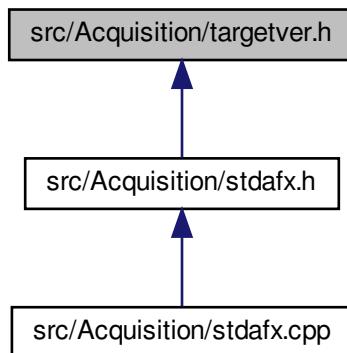


## 15.219 src/Acquisition/targetver.h File Reference

Include dependency graph for targetver.h:

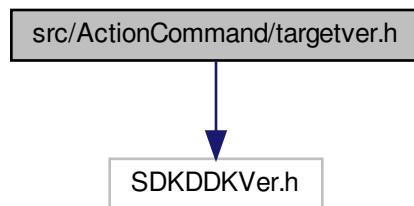


This graph shows which files directly or indirectly include this file:

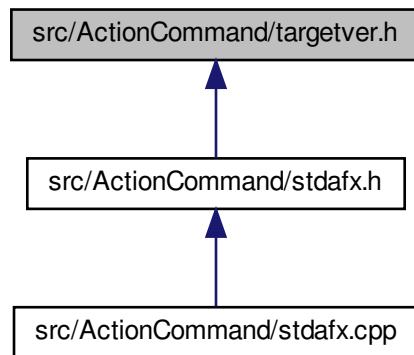


## 15.220 src/ActionCommand/targetver.h File Reference

Include dependency graph for targetver.h:

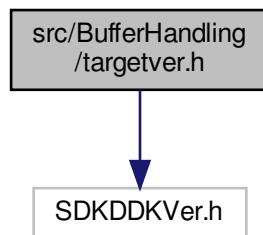


This graph shows which files directly or indirectly include this file:

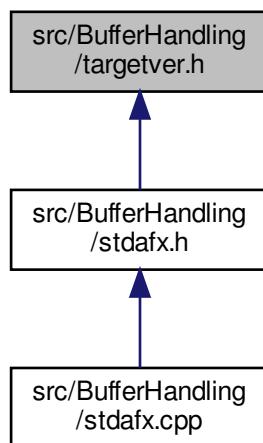


## 15.221 src/BufferHandling/targetver.h File Reference

Include dependency graph for targetver.h:

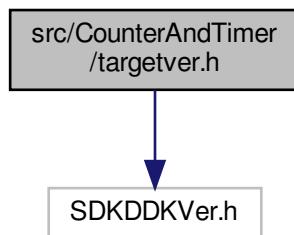


This graph shows which files directly or indirectly include this file:

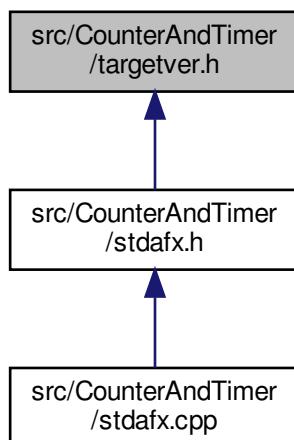


## 15.222 src/CounterAndTimer/targetver.h File Reference

Include dependency graph for targetver.h:

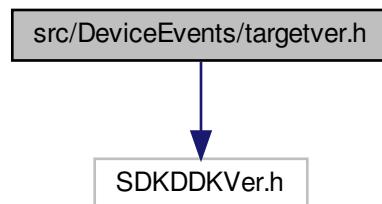


This graph shows which files directly or indirectly include this file:

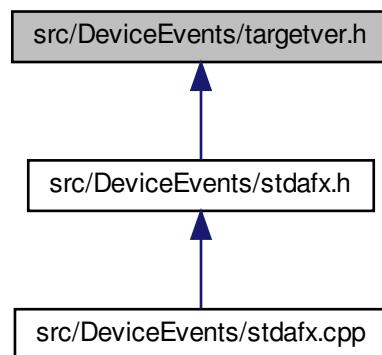


## 15.223 src/DeviceEvents/targetver.h File Reference

Include dependency graph for targetver.h:

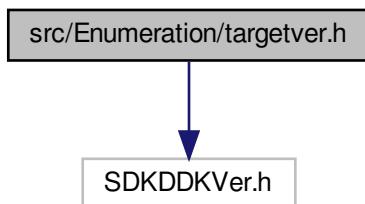


This graph shows which files directly or indirectly include this file:

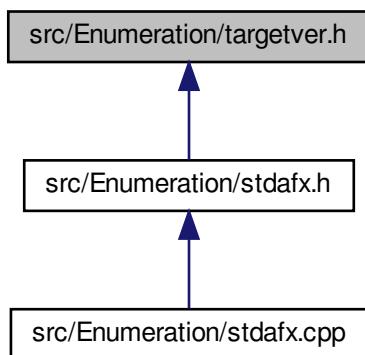


## 15.224 src/Enumeration/targetver.h File Reference

Include dependency graph for targetver.h:

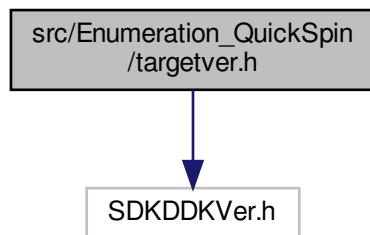


This graph shows which files directly or indirectly include this file:

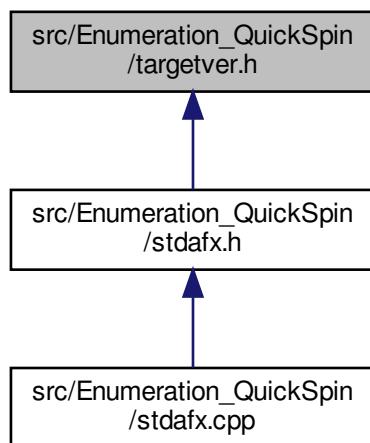


## 15.225 src/Enumeration\_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

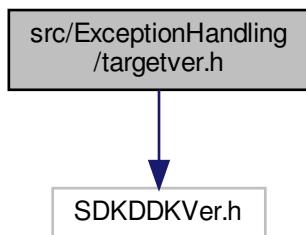


This graph shows which files directly or indirectly include this file:

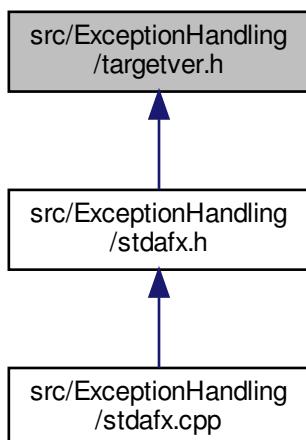


## 15.226 src/ExceptionHandling/targetver.h File Reference

Include dependency graph for targetver.h:

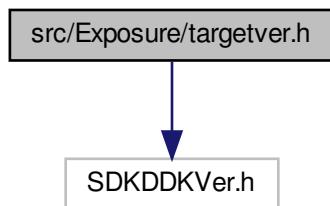


This graph shows which files directly or indirectly include this file:

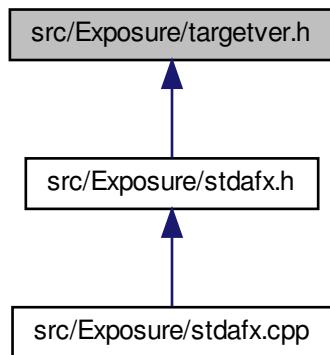


## 15.227 src/Exposure/targetver.h File Reference

Include dependency graph for targetver.h:

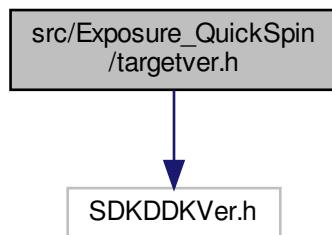


This graph shows which files directly or indirectly include this file:

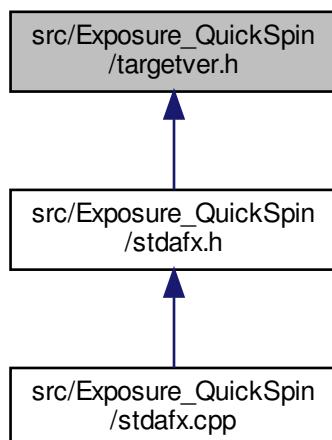


## 15.228 src/Exposure\_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

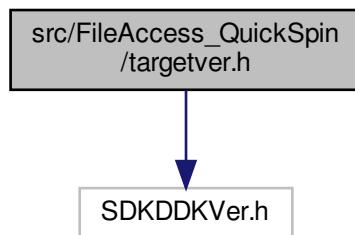


This graph shows which files directly or indirectly include this file:

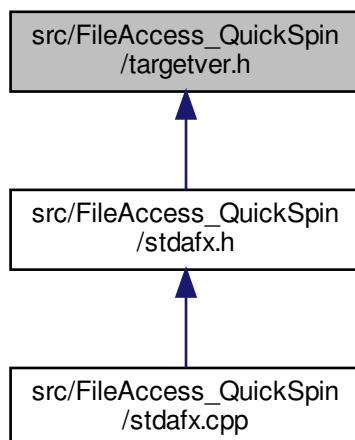


## 15.229 src/FileAccess\_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

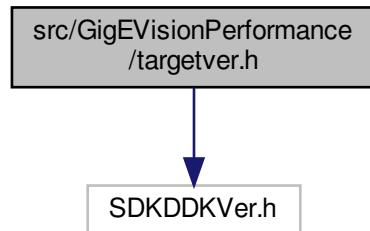


This graph shows which files directly or indirectly include this file:

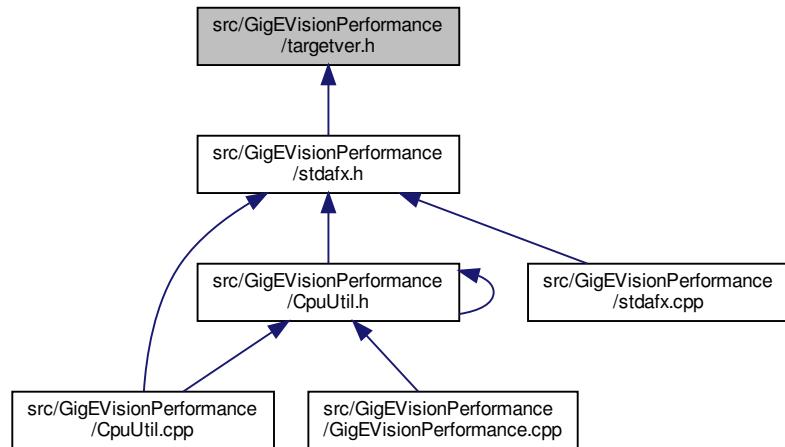


## 15.230 src/GigEVisionPerformance/targetver.h File Reference

Include dependency graph for targetver.h:

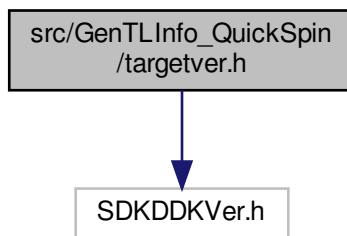


This graph shows which files directly or indirectly include this file:



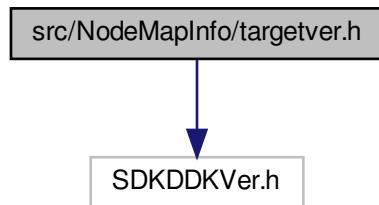
## 15.231 src/GenTLInfo\_QuickSpin/targetver.h File Reference

Include dependency graph for targetver.h:

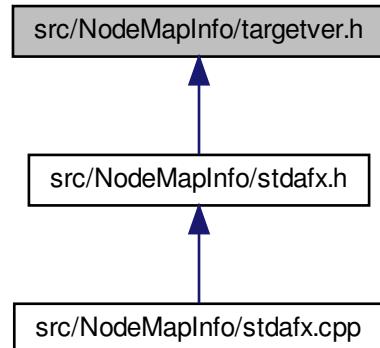


## 15.232 src/NodeMapInfo/targetver.h File Reference

Include dependency graph for targetver.h:

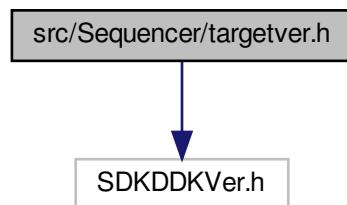


This graph shows which files directly or indirectly include this file:

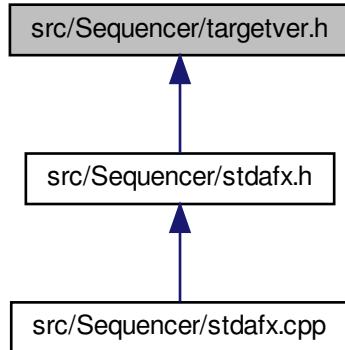


### 15.233 src/Sequencer/targetver.h File Reference

Include dependency graph for targetver.h:

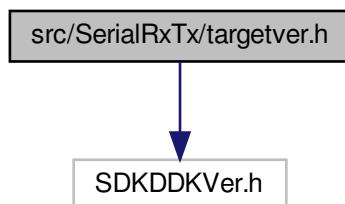


This graph shows which files directly or indirectly include this file:

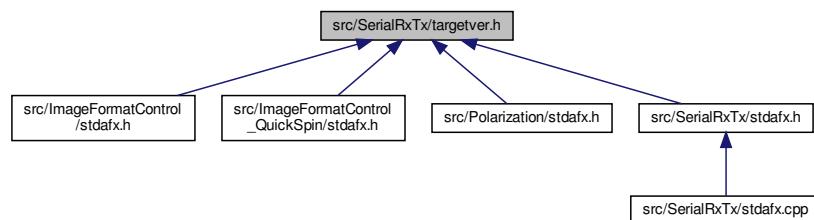


## 15.234 src/SerialRxTx/targetver.h File Reference

Include dependency graph for targetver.h:

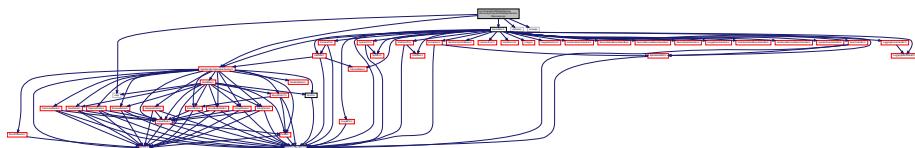


This graph shows which files directly or indirectly include this file:



## 15.235 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](#)

### 15.235.1 Function Documentation

#### 15.235.1.1 ConfigureCamera()

```
bool ConfigureCamera (
    CameraPtr pCam )
```

### **15.235.1.2 ConfigureUserSet1()**

```
bool ConfigureUserSet1 (
    CameraPtr pCam )
```

### **15.235.1.3 GetDeviceSerial()**

```
string GetDeviceSerial (
    CameraPtr pCam )
```

### **15.235.1.4 main()**

```
int main (
    int ,
    char ** )
```

### **15.235.1.5 PrintExampleStatistics()**

```
void PrintExampleStatistics ( )
```

### **15.235.1.6 RefreshCameraList()**

```
void RefreshCameraList (
    SystemPtr system )
```

### **15.235.1.7 ResetCameraUserSetToDefault()**

```
void ResetCameraUserSetToDefault (
    CameraPtr pCam )
```

### **15.235.1.8 SleepyWrapper()**

```
void SleepyWrapper (
    int milliseconds )
```

## 15.235.2 Variable Documentation

### 15.235.2.1 cameraGrabInfoMap

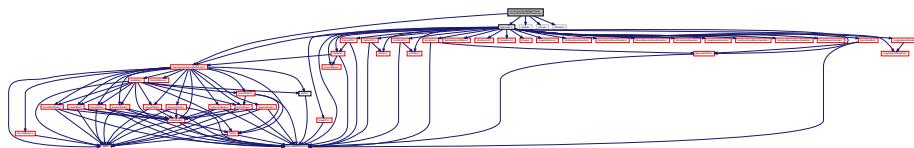
```
std::map<std::string, GrabInfo> cameraGrabInfoMap
```

### 15.235.2.2 globalCamList

```
CameraList globalCamList
```

## 15.236 src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp File Reference

Include dependency graph for AcquisitionMultipleThread.cpp:



## Functions

- int [PrintDeviceInfo](#) (INodeMap &nodeMap, std::string camSerial)
- void \* [AcquireImages](#) (void \*arg)
- int [RunMultipleCameras](#) (CameraList camList)
- int [main](#) (int, char \*\*)

## 15.236.1 Function Documentation

### 15.236.1.1 AcquireImages()

```
void* AcquireImages (
    void * arg )
```

### 15.236.1.2 main()

```
int main (
    int ,
    char ** )
```

### 15.236.1.3 PrintDeviceInfo()

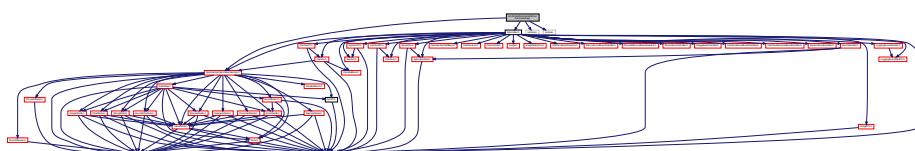
```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    std::string camSerial )
```

### 15.236.1.4 RunMultipleCameras()

```
int RunMultipleCameras (
    CameraList camList )
```

## 15.237 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 \*\*)

## 15.237.1 Function Documentation

### 15.237.1.1 AcquireImages()

```
int AcquireImages (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    CameraList camList )
```

### 15.237.1.2 ConfigureActionControl()

```
int ConfigureActionControl (
    const CameraList & camList )
```

### 15.237.1.3 ConfigureChunkData()

```
int ConfigureChunkData (
    const CameraList & camList )
```

### 15.237.1.4 ConfigureIEEE1588()

```
int ConfigureIEEE1588 (
    const CameraList & camList )
```

### 15.237.1.5 ConfigureInterface()

```
int ConfigureInterface (
    const InterfaceList & interfaceList )
```

### 15.237.1.6 ConfigureOtherNodes()

```
int ConfigureOtherNodes (
    const CameraList & camList )
```

### 15.237.1.7 ConfigureTrigger()

```
int ConfigureTrigger (
    const CameraList camList )
```

### 15.237.1.8 main()

```
int main (
    int ,
    char ** )
```

### 15.237.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap,
    const unsigned int camNum )
```

### 15.237.1.10 RunMultipleCameras()

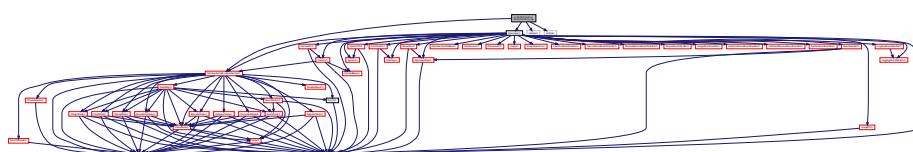
```
int RunMultipleCameras (
    const SystemPtr & system,
    const InterfaceList & interfaceList,
    const CameraList & camList )
```

### 15.237.1.11 SleepyWrapper()

```
void SleepyWrapper (
    int milliseconds )
```

## 15.238 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 \*\*)

## 15.238.1 Macro Definition Documentation

### 15.238.1.1 k\_numLoops

```
#define k_numLoops 9
```

### 15.238.1.2 numBuffers

```
#define numBuffers 3
```

### 15.238.1.3 z\_numTriggers

```
#define z_numTriggers 6
```

## 15.238.2 Function Documentation

### 15.238.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

**15.238.2.2 ConfigureTrigger()**

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

**15.238.2.3 GrabNextImageByTrigger()**

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap )
```

**15.238.2.4 main()**

```
int main (
    int ,
    char ** )
```

**15.238.2.5 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.238.2.6 ResetTrigger()**

```
int ResetTrigger (
    INodeMap & nodeMap )
```

**15.238.2.7 RunSingleCamera()**

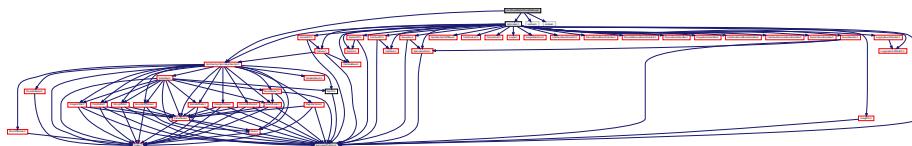
```
int RunSingleCamera (
    CameraPtr pCam )
```

**15.238.2.8 SleepyWrapper()**

```
void SleepyWrapper (
    int milliseconds )
```

## 15.239 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 plImage)
- 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`

#### 15.239.1 Enumeration Type Documentation

##### 15.239.1.1 `chunkDataType`

enum `chunkDataType`

Enumerator

<code>IMAGE</code>	
<code>NODEMAP</code>	

## 15.239.2 Function Documentation

### 15.239.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice )
```

### 15.239.2.2 ConfigureChunkData()

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

### 15.239.2.3 DisableChunkData()

```
int DisableChunkData (
    INodeMap & nodeMap )
```

### 15.239.2.4 DisplayChunkData() [1/2]

```
int DisplayChunkData (
    ImagePtr pImage )
```

### 15.239.2.5 DisplayChunkData() [2/2]

```
int DisplayChunkData (
    INodeMap & nodeMap )
```

### 15.239.2.6 main()

```
int main (
    int ,
    char ** )
```

### 15.239.2.7 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.239.2.8 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

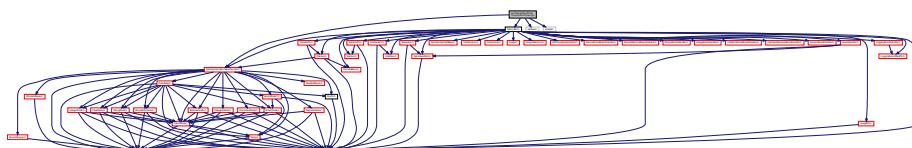
## 15.239.3 Variable Documentation

### 15.239.3.1 chosenChunkData

```
const chunkDataType chosenChunkData = IMAGE
```

## 15.240 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 \*\*)

### 15.240.1 Function Documentation

### 15.240.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice )
```

### 15.240.1.2 ConfigureDigitalIO()

```
int ConfigureDigitalIO (
    INodeMap & nodeMap )
```

### 15.240.1.3 ConfigureExposureandTrigger()

```
int ConfigureExposureandTrigger (
    INodeMap & nodeMap )
```

### 15.240.1.4 main()

```
int main (
    int ,
    char ** )
```

### 15.240.1.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.240.1.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

### 15.240.1.7 RunSingleCamera()

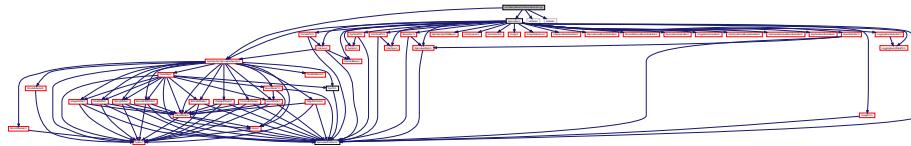
```
int RunSingleCamera (
    CameraPtr pCam )
```

### 15.240.1.8 SetupCounterAndTimer()

```
int SetupCounterAndTimer (
    INodeMap & nodeMap )
```

## 15.241 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 \*&deviceEventHandler)
- int [ResetDeviceEvents](#) (CameraPtr pCam, DeviceEventHandlerImpl \*&deviceEventHandler)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [AcquireImages](#) (CameraPtr pCam, INodeMap &nodeMap, INodeMap &nodeMapTLDevice)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char \*\*)

### Variables

- const [eventType](#) chosenEvent = [GENERIC](#)

## 15.241.1 Enumeration Type Documentation

### 15.241.1.1 eventType

```
enum eventType
```

Enumerator

GENERIC	
SPECIFIC	

## 15.241.2 Function Documentation

### 15.241.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice )
```

### 15.241.2.2 ConfigureDeviceEvents()

```
int ConfigureDeviceEvents (
    INodeMap & nodeMap,
    CameraPtr pCam,
    DeviceEventHandlerImpl *& deviceEventHandler )
```

### 15.241.2.3 main()

```
int main (
    int ,
    char ** )
```

### 15.241.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.241.2.5 ResetDeviceEvents()

```
int ResetDeviceEvents (
    CameraPtr pCam,
    DeviceEventHandlerImpl *& deviceEventHandler )
```

### 15.241.2.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

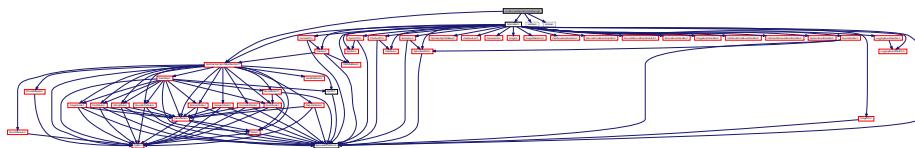
## 15.241.3 Variable Documentation

### 15.241.3.1 chosenEvent

```
const eventType chosenEvent = GENERIC
```

## 15.242 src/Enumeration/Enumeration.cpp File Reference

Include dependency graph for Enumeration.cpp:



## Functions

- int `QueryInterface` (`InterfacePtr` pInterface)
- int `main` (int, char \*\*)

### 15.242.1 Function Documentation

#### 15.242.1.1 main()

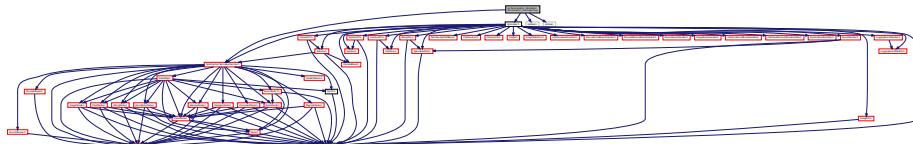
```
int main (
    int ,
    char ** )
```

#### 15.242.1.2 QueryInterface()

```
int QueryInterface (
    InterfacePtr pInterface )
```

## 15.243 src/Enumeration\_QuickSpin/Enumeration\_QuickSpin.cpp File Reference

Include dependency graph for Enumeration\_QuickSpin.cpp:



### Functions

- int [QueryInterface \(InterfacePtr pInterface\)](#)
- int [main \(int, char \\*\\*\)](#)

#### 15.243.1 Function Documentation

##### 15.243.1.1 main()

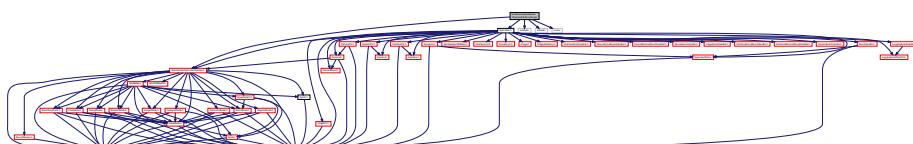
```
int main (
    int ,
    char ** )
```

##### 15.243.1.2 QueryInterface()

```
int QueryInterface (
    InterfacePtr pInterface )
```

## 15.244 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 \*\*)

### 15.244.1 Function Documentation

#### 15.244.1.1 [CheckGevEnabled\(\)](#)

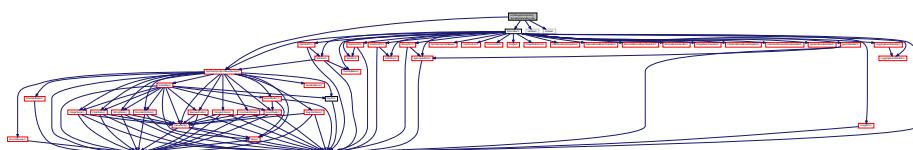
```
void CheckGevEnabled (
    SystemPtr & pSystem )
```

#### 15.244.1.2 [main\(\)](#)

```
int main (
    int ,
    char ** )
```

### 15.245 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`

### 15.245.1 Enumeration Type Documentation

#### 15.245.1.1 exceptionType

```
enum exceptionType
```

##### Enumerator

SPINNAKER_EXCEPTION
STANDARD_EXCEPTION
STANDARD_CAST_TO_SPINNAKER

### 15.245.2 Function Documentation

#### 15.245.2.1 causeSpinnakerException()

```
void causeSpinnakerException ( )
```

#### 15.245.2.2 causeStandardException()

```
void causeStandardException ( )
```

#### 15.245.2.3 main()

```
int main (
    int ,
    char ** )
```

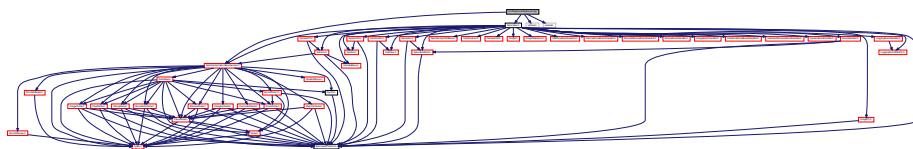
### 15.245.3 Variable Documentation

#### 15.245.3.1 chosenException

```
const exceptionType chosenException = SPINNAKER_EXCEPTION
```

## 15.246 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 \*\*)

### 15.246.1 Function Documentation

#### 15.246.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

#### 15.246.1.2 ConfigureExposure()

```
int ConfigureExposure (
    INodeMap & nodeMap )
```

#### 15.246.1.3 main()

```
int main (
    int ,
    char ** )
```

#### 15.246.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

#### 15.246.1.5 ResetExposure()

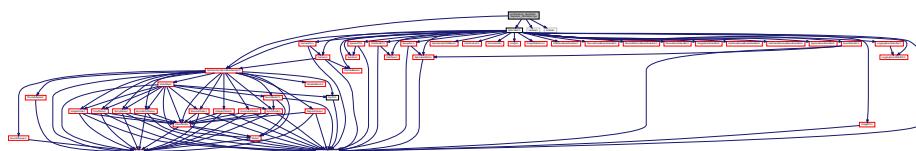
```
int ResetExposure (
    INodeMap & nodeMap )
```

#### 15.246.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.247 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 \(int, char \\*\\*\)](#)

## 15.247.1 Function Documentation

### 15.247.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

### 15.247.1.2 ConfigureExposure()

```
int ConfigureExposure (
    CameraPtr pCam )
```

### 15.247.1.3 main()

```
int main (
    int ,
    char ** )
```

### 15.247.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

### 15.247.1.5 ResetExposure()

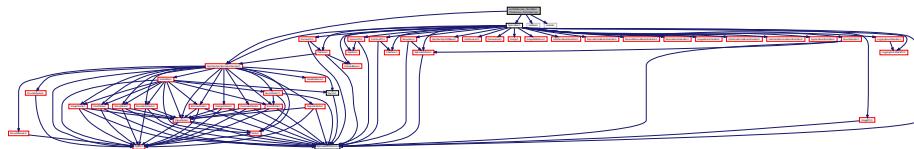
```
int ResetExposure (
    CameraPtr pCam )
```

### 15.247.1.6 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.248 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"

#### 15.248.1 Function Documentation

##### 15.248.1.1 AcquireImages()

```
bool AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImagePtr pReferenceImage )
```

**15.248.1.2 CloseFile()**

```
bool CloseFile (
    CameraPtr pCam )
```

**15.248.1.3 DownloadImage()**

```
bool DownloadImage ( )
```

**15.248.1.4 ExecuteDeleteCommand()**

```
bool ExecuteDeleteCommand (
    CameraPtr pCam )
```

**15.248.1.5 ExecuteReadCommand()**

```
bool ExecuteReadCommand (
    CameraPtr pCamera )
```

**15.248.1.6 ExecuteWriteCommand()**

```
bool ExecuteWriteCommand (
    CameraPtr pCam )
```

**15.248.1.7 InitializeSystem()**

```
bool InitializeSystem (
    SystemPtr & system,
    CameraList & camList,
    CameraPtr & pCam )
```

**15.248.1.8 main()**

```
int main (
    int argc,
    char * argv[ ] )
```

**15.248.1.9 OpenFileToRead()**

```
bool OpenFileToRead (
    CameraPtr pCamera )
```

**15.248.1.10 OpenFileToWrite()**

```
bool OpenFileToWrite (
    CameraPtr pCam )
```

**15.248.1.11 PrintDebugMessage()**

```
static void PrintDebugMessage (
    string msg ) [static]
```

**15.248.1.12 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.248.1.13 PrintResultMessage()**

```
static void PrintResultMessage (
    bool result ) [static]
```

**15.248.1.14 PrintUsage()**

```
void PrintUsage ( )
```

**15.248.1.15 UploadImage()**

```
bool UploadImage ( )
```

## 15.248.2 Variable Documentation

### 15.248.2.1 \_enableDebug

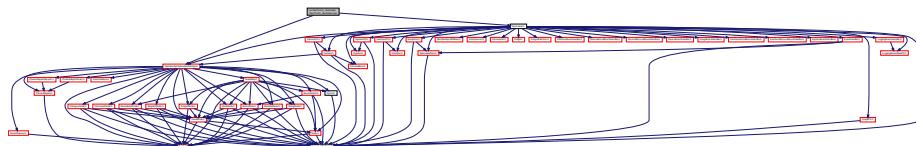
```
bool _enableDebug = false [static]
```

### 15.248.2.2 \_fileSelector

```
gcstring _fileSelector = "UserFile1" [static]
```

## 15.249 src/GenTLInfo\_QuickSpin/GenTLInfo\_QuickSpin.cpp File Reference

Include dependency graph for GenTLInfo\_QuickSpin.cpp:



## Functions

- int PrintTransportLayerDeviceInfo (CameraPtr pCamera)
- int PrintTransportLayerStreamInfo (CameraPtr pCamera)
- int PrintTransportLayerInterfaceInfo (InterfacePtr pInterface)
- int PrintApplicationLayerDeviceInfo (CameraPtr pCamera)
- int main (int, char \*\*)

## 15.249.1 Function Documentation

### 15.249.1.1 main()

```
int main (
    int ,
    char ** )
```

### 15.249.1.2 PrintApplicationLayerDeviceInfo()

```
int PrintApplicationLayerDeviceInfo (
    CameraPtr pCamera )
```

### 15.249.1.3 PrintTransportLayerDeviceInfo()

```
int PrintTransportLayerDeviceInfo (
    CameraPtr pCamera )
```

### 15.249.1.4 PrintTransportLayerInterfaceInfo()

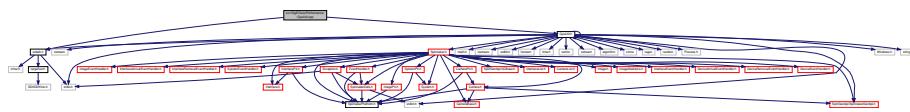
```
int PrintTransportLayerInterfaceInfo (
    InterfacePtr pInterface )
```

### 15.249.1.5 PrintTransportLayerStreamInfo()

```
int PrintTransportLayerStreamInfo (
    CameraPtr pCamera )
```

## 15.250 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

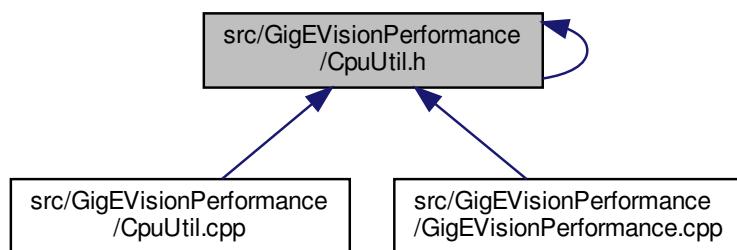
- double PCFreq
  - \_\_int64 CounterStart
  - time\_t startTime
  - time\_t endTime
  - double timeDiff

## 15.251 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

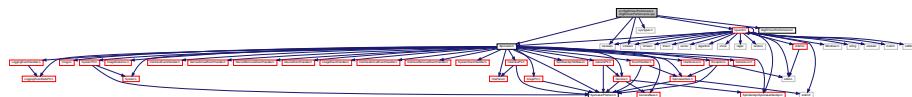
- [CpuUtil](#)
- [PerformanceCounter](#)
- [SecondsCounter](#)
- [Conversion](#)

## 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)

## 15.252 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 numImages, ToAcquire, 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 = "-?"`

## 15.252.1 Function Documentation

### 15.252.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    int numImagesToAcquire,
    int iteration )
```

### 15.252.1.2 EnableManualFramerate()

```
bool EnableManualFramerate (
    CameraPtr pCam )
```

### 15.252.1.3 getCameraCategory()

```
void getCameraCategory (
    INodeMap & nodeMap,
    string categoryString )
```

#### 15.252.1.4 main()

```
int main (
    int argc,
    char * argv[ ] )
```

#### 15.252.1.5 ParseArguments()

```
bool ParseArguments (
    int argc,
    char * argv[ ] )
```

#### 15.252.1.6 PrintAllNodes()

```
void PrintAllNodes (
    CameraPtr pCam )
```

#### 15.252.1.7 PrintCPUUsage()

```
void PrintCPUUsage ( )
```

#### 15.252.1.8 PrintDataStreamInfo()

```
void PrintDataStreamInfo (
    const Spinnaker::CameraPtr pCamera )
```

#### 15.252.1.9 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

#### 15.252.1.10 PrintUsage()

```
void PrintUsage ( )
```

**15.252.1.11 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr pCam )
```

**15.252.1.12 SetFrameRate()**

```
bool SetFrameRate (
    CameraPtr pCam )
```

**15.252.2 Variable Documentation****15.252.2.1 argBayerRG**

```
const char* argBayerRG = "-bayerrg"
```

**15.252.2.2 argDuration**

```
const char* argDuration = "-duration"
```

**15.252.2.3 argMaxFrames**

```
const char* argMaxFrames = "-maxfps"
```

**15.252.2.4 argNumImages**

```
const char* argNumImages = "-numimages"
```

**15.252.2.5 argPacketDelay**

```
const char* argPacketDelay = "-packetdelay"
```

### 15.252.2.6 argPacketSize

```
const char* argPacketSize = "-packetsize"
```

### 15.252.2.7 argPrintUsage

```
const char* argPrintUsage = "-?"
```

### 15.252.2.8 argRelease

```
const char* argRelease = "-callrelease"
```

### 15.252.2.9 argUserSetFrames

```
const char* argUserSetFrames = "-fps"
```

### 15.252.2.10 cpuUsageInfo

```
CpuUtil::CpuUsageInfo cpuUsageInfo
```

### 15.252.2.11 IsRelease

```
bool IsRelease = false
```

### 15.252.2.12 NumImagesToGrab

```
int NumImagesToGrab = 100
```

### 15.252.2.13 PacketDelayToSet

```
int PacketDelayToSet = 0
```

**15.252.2.14 PacketSizeToSet**

```
int PacketSizeToSet = 9000
```

**15.252.2.15 PixelFormatToSet**

```
char* PixelFormatToSet = nullptr
```

**15.252.2.16 TestDuration**

```
int TestDuration = 0
```

**15.252.2.17 UseDuration**

```
bool UseDuration = false
```

**15.252.2.18 UseMaxFramerate**

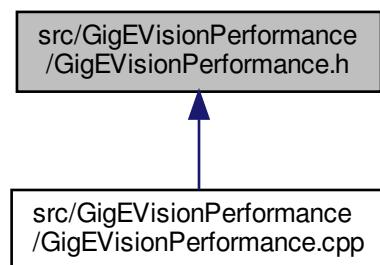
```
bool UseMaxFramerate = false
```

**15.252.2.19 UserSetFramerate**

```
float UserSetFramerate = 0.0
```

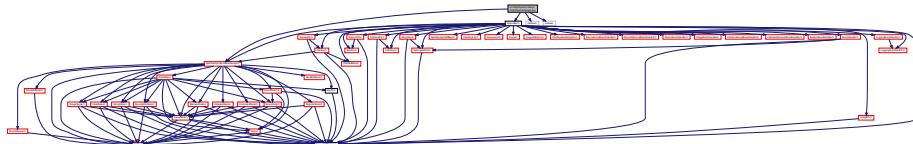
## 15.253 src/GigEVisionPerformance/GigEVisionPerformance.h File Reference

This graph shows which files directly or indirectly include this file:



## 15.254 src/HighDynamicRange/HighDynamicRange.cpp File Reference

Include dependency graph for HighDynamicRange.cpp:



### Functions

- void `PrintBuildInfo ()`
- void `PrintDeviceInfo (INodeMap &nodeMap)`
- bool `CheckNodeAccessibility (const CNodePtr nodePtr)`
- bool `ToggleHDRMode (INodeMap &nodeMap, bool hdrOn)`
- bool `InitializeHDRImages (INodeMap &nodeMap)`
- int `RunSingleCamera (CameraPtr cam)`
- int `main (int, char **)`

### Variables

- const unsigned int `k_HDRShutter1` = 1000
- const unsigned int `k_HDRShutter2` = 5000
- const unsigned int `k_HDRShutter3` = 15000
- const unsigned int `k_HDRShutter4` = 30000
- const unsigned int `k_HDRGain1` = 0
- const unsigned int `k_HDRGain2` = 5
- const unsigned int `k_HDRGain3` = 10
- const unsigned int `k_HDRGain4` = 15

### 15.254.1 Function Documentation

#### 15.254.1.1 CheckNodeAccessibility()

```
bool CheckNodeAccessibility (
    const CNodePtr nodePtr )
```

#### 15.254.1.2 InitializeHDRImages()

```
bool InitializeHDRImages (
    INodeMap & nodeMap )
```

**15.254.1.3 main()**

```
int main (
    int ,
    char ** )
```

**15.254.1.4 PrintBuildInfo()**

```
void PrintBuildInfo ( )
```

**15.254.1.5 PrintDeviceInfo()**

```
void PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.254.1.6 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr cam )
```

**15.254.1.7 ToggleHDRMode()**

```
bool ToggleHDRMode (
    INodeMap & nodeMap,
    bool hdrOn )
```

**15.254.2 Variable Documentation****15.254.2.1 k\_HDRGain1**

```
const unsigned int k_HDRGain1 = 0
```

### 15.254.2.2 k\_HDRGain2

```
const unsigned int k_HDRGain2 = 5
```

### 15.254.2.3 k\_HDRGain3

```
const unsigned int k_HDRGain3 = 10
```

### 15.254.2.4 k\_HDRGain4

```
const unsigned int k_HDRGain4 = 15
```

### 15.254.2.5 k\_HDRShutter1

```
const unsigned int k_HDRShutter1 = 1000
```

### 15.254.2.6 k\_HDRShutter2

```
const unsigned int k_HDRShutter2 = 5000
```

### 15.254.2.7 k\_HDRShutter3

```
const unsigned int k_HDRShutter3 = 15000
```

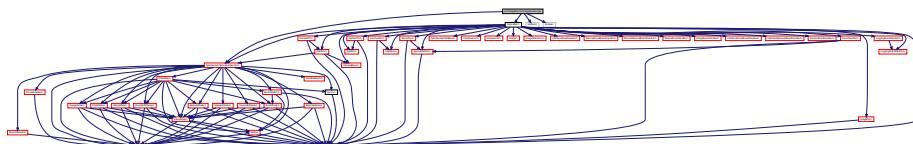
### 15.254.2.8 k\_HDRShutter4

```
const unsigned int k_HDRShutter4 = 30000
```

---

## 15.255 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 \*\*)

### 15.255.1 Function Documentation

#### 15.255.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    ImageEventHandlerImpl *& imageEventHandler )
```

#### 15.255.1.2 ConfigureImageEvents()

```
int ConfigureImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

**15.255.1.3 main()**

```
int main (
    int ,
    char ** )
```

**15.255.1.4 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.255.1.5 ResetImageEvents()**

```
int ResetImageEvents (
    CameraPtr pCam,
    ImageEventHandlerImpl *& imageEventHandler )
```

**15.255.1.6 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr pCam )
```

**15.255.1.7 SleepyWrapper()**

```
void SleepyWrapper (
    int milliseconds )
```

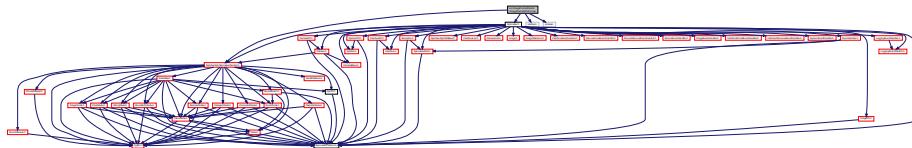
**15.255.1.8 WaitForImages()**

```
int WaitForImages (
    ImageEventHandlerImpl *& imageEventHandler )
```

---

## 15.256 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 \*\*)

#### 15.256.1 Function Documentation

##### 15.256.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

##### 15.256.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
    INodeMap & nodeMap )
```

##### 15.256.1.3 main()

```
int main (
    int ,
    char ** )
```

#### 15.256.1.4 PrintDeviceInfo()

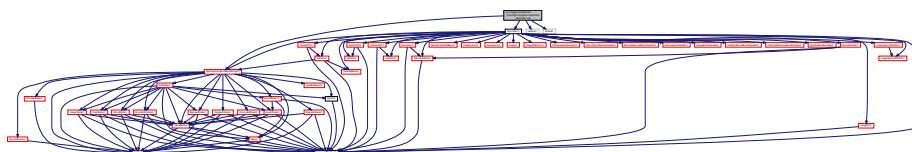
```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

#### 15.256.1.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.257 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 \(int, char \\*\\*\)](#)

### 15.257.1 Function Documentation

#### 15.257.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

### 15.257.1.2 ConfigureCustomImageSettings()

```
int ConfigureCustomImageSettings (
    CameraPtr pCam )
```

### 15.257.1.3 main()

```
int main (
    int ,
    char ** )
```

### 15.257.1.4 PrintDeviceInfo()

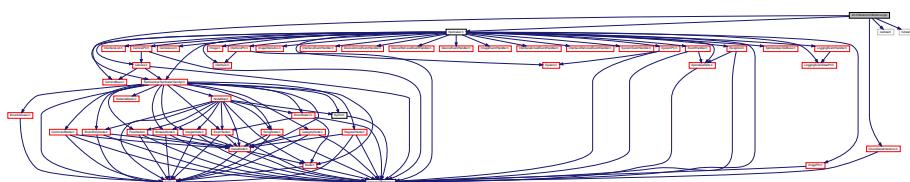
```
int PrintDeviceInfo (
    CameraPtr pCam )
```

### 15.257.1.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.258 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 pImage)
- 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 `injectedImagePath`
- 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` []

### 15.258.1 Enumeration Type Documentation

#### 15.258.1.1 FileUploadPersistence

```
enum FileUploadPersistence
```

Enumerator

FLASH	
DDR	

### 15.258.1.2 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.

## 15.258.2 Function Documentation

### 15.258.2.1 AcquireImages()

```
int AcquireImages (
    const CameraPtr & pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice )
```

### 15.258.2.2 CameraCloseFile()

```
bool CameraCloseFile (
    INodeMap & nodeMap )
```

### 15.258.2.3 CameraDeleteFile()

```
bool CameraDeleteFile (
    INodeMap & nodeMap )
```

### 15.258.2.4 CameraOpenFile()

```
bool CameraOpenFile (
    INodeMap & nodeMap )
```

**15.258.2.5 CameraWriteToFile()**

```
bool CameraWriteToFile (
    INodeMap & nodeMap )
```

**15.258.2.6 ConfigureChunkData()**

```
int ConfigureChunkData (
    INodeMap & nodeMap )
```

**15.258.2.7 ConfigureInference()**

```
int ConfigureInference (
    INodeMap & nodeMap,
    bool isEnabled )
```

**15.258.2.8 ConfigureTestPattern()**

```
int ConfigureTestPattern (
    INodeMap & nodeMap,
    bool isEnabled )
```

**15.258.2.9 ConfigureTrigger()**

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

**15.258.2.10 DeleteFileOnCamera()**

```
int DeleteFileOnCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName )
```

**15.258.2.11 DisableChunkData()**

```
int DisableChunkData (
    INodeMap & nodeMap )
```

**15.258.2.12 DisableTrigger()**

```
int DisableTrigger (
    INodeMap & nodeMap )
```

**15.258.2.13 DisplayChunkData()**

```
int DisplayChunkData (
    const ImagePtr pImage )
```

**15.258.2.14 labelClassification()**

```
const std::vector<std::string> labelClassification (
    arrayLabelClassification ,
    end(arrayLabelClassification) )
```

**15.258.2.15 labelDetection()**

```
const std::vector<std::string> labelDetection (
    arrayLabelDetection ,
    end(arrayLabelDetection) )
```

**15.258.2.16 LoadFileIntoMemory()**

```
std::vector<char> LoadFileIntoMemory (
    const string & filename )
```

**15.258.2.17 main()**

```
int main (
    int ,
    char ** )
```

**15.258.2.18 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.258.2.19 RunSingleCamera()**

```
int RunSingleCamera (
    const CameraPtr & pCam )
```

**15.258.2.20 SetChunkEnable()**

```
int SetChunkEnable (
    INodeMap & nodeMap,
    const gcstring & entryName,
    const bool enable )
```

**15.258.2.21 UploadFileToCamera()**

```
int UploadFileToCamera (
    INodeMap & nodeMap,
    const std::string & fileSelectorEntryName,
    const std::string & filePath )
```

**15.258.3 Variable Documentation****15.258.3.1 arrayLabelClassification**

```
const char* arrayLabelClassification[] = {"daisy", "dandelion", "roses", "sunflowers", "tulips"}
```

### 15.258.3.2 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"}
```

### 15.258.3.3 chosenFileUploadPersistence

```
const FileUploadPersistence chosenFileUploadPersistence = DDR
```

### 15.258.3.4 chosenInferenceNetworkType

```
const InferenceNetworkType chosenInferenceNetworkType = CLASSIFICATION
```

### 15.258.3.5 injectedImagePath

```
const std::string injectedImagePath
```

**Initial value:**

```
=
(chosenInferenceNetworkType == CLASSIFICATION ? "Injected_Image_Classification.raw"
 : "Injected_Image_Detection.raw")
```

### 15.258.3.6 injectedImageHeight

```
const unsigned int injectedImageHeight = (chosenInferenceNetworkType == CLASSIFICATION ? 400
: 540)
```

### 15.258.3.7 injectedImageWidth

```
const unsigned int injectedImageWidth = (chosenInferenceNetworkType == CLASSIFICATION ? 640
: 720)
```

### 15.258.3.8 networkFilePath

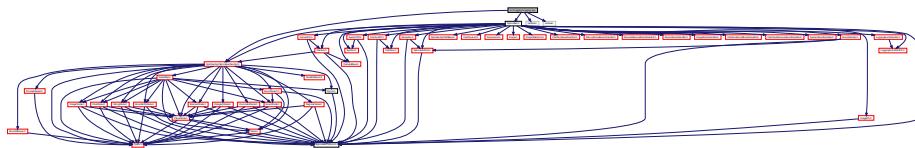
```
const std::string networkFilePath
```

#### Initial value:

```
=  
  (chosenInferenceNetworkType == CLASSIFICATION ? "Network_Classification" : "Network_Detection")
```

## 15.259 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](#) = LOG\_LEVEL\_DEBUG

### 15.259.1 Function Documentation

#### 15.259.1.1 main()

```
int main (  
    int ,  
    char ** )
```

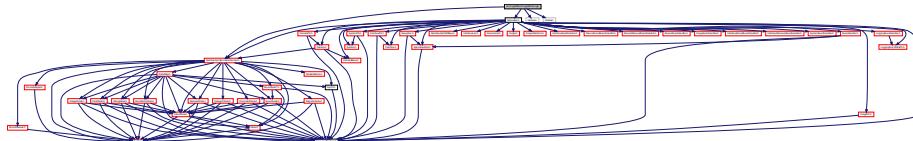
### 15.259.2 Variable Documentation

### 15.259.2.1 k\_LoggingLevel

```
const SpinnakerLogLevel k_LoggingLevel = LOG_LEVEL_DEBUG
```

## 15.260 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 \*\*)

### 15.260.1 Function Documentation

#### 15.260.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

#### 15.260.1.2 ConfigureLogicBlock()

```
int ConfigureLogicBlock (
    INodeMap & nodeMap )
```

**15.260.1.3 ConfigureTrigger()**

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

**15.260.1.4 GrabTwoImages()**

```
int GrabTwoImages (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

**15.260.1.5 main()**

```
int main (
    int ,
    char ** )
```

**15.260.1.6 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.260.1.7 ResetExposure()**

```
int ResetExposure (
    INodeMap & nodeMap )
```

**15.260.1.8 ResetTrigger()**

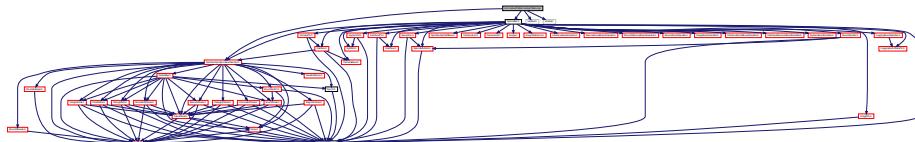
```
int ResetTrigger (
    INodeMap & nodeMap )
```

**15.260.1.9 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.261 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 \*\*)

### 15.261.1 Function Documentation

#### 15.261.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice )
```

#### 15.261.1.2 ConfigureLookupTables()

```
int ConfigureLookupTables (
    INodeMap & nodeMap )
```

#### 15.261.1.3 main()

```
int main (
    int ,
    char ** )
```

#### 15.261.1.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

#### 15.261.1.5 PrintRetrieveNodeFailure()

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

#### 15.261.1.6 ResetLookupTables()

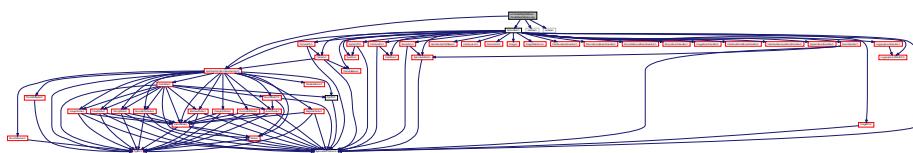
```
int ResetLookupTables (
    INodeMap & nodeMap )
```

#### 15.261.1.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.262 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, int64\_t &callbackHeight, int64\_t &callbackGain)
- int [ChangeHeightAndGain](#) (INodeMap &nodeMap)
- int [ResetCallbacks](#) (INodeMap &nodeMap, int64\_t &callbackHeight, int64\_t &callbackGain)
- int [PrintDeviceInfo](#) (INodeMap &nodeMap)
- int [RunSingleCamera](#) (CameraPtr pCam)
- int [main](#) (int, char \*\*)

## 15.262.1 Function Documentation

### 15.262.1.1 ChangeHeightAndGain()

```
int ChangeHeightAndGain (
    INodeMap & nodeMap )
```

### 15.262.1.2 ConfigureCallbacks()

```
int ConfigureCallbacks (
    INodeMap & nodeMap,
    int64_t & callbackHeight,
    int64_t & callbackGain )
```

### 15.262.1.3 main()

```
int main (
    int ,
    char ** )
```

### 15.262.1.4 OnGainNodeUpdate()

```
void OnGainNodeUpdate (
    INode * node )
```

### 15.262.1.5 OnHeightNodeUpdate()

```
void OnHeightNodeUpdate (
    INode * node )
```

### 15.262.1.6 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.262.1.7 ResetCallbacks()

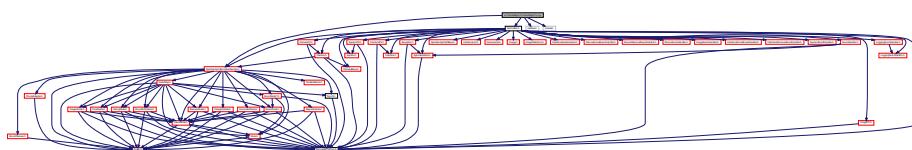
```
int ResetCallbacks (
    INodeMap & nodeMap,
    int64_t & callbackHeight,
    int64_t & callbackGain )
```

### 15.262.1.8 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

## 15.263 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`

### 15.263.1 Enumeration Type Documentation

#### 15.263.1.1 `readType`

```
enum readType
```

Enumerator

VALUE	
INDIVIDUAL	

### 15.263.2 Function Documentation

#### 15.263.2.1 `Indent()`

```
void Indent (
    unsigned int level )
```

#### 15.263.2.2 `main()`

```
int main (
    int ,
    char ** )
```

#### 15.263.2.3 `PrintBooleanNode()`

```
int PrintBooleanNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.4 PrintCategoryNodeAndAllFeatures()**

```
int PrintCategoryNodeAndAllFeatures (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.5 PrintCommandNode()**

```
int PrintCommandNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.6 PrintEnumerationNodeAndCurrentEntry()**

```
int PrintEnumerationNodeAndCurrentEntry (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.7 PrintEnumerationSelector()**

```
int PrintEnumerationSelector (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.8 PrintFloatNode()**

```
int PrintFloatNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.9 PrintIntegerNode()**

```
int PrintIntegerNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.10 PrintNode()**

```
int PrintNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.11 PrintStringNode()**

```
int PrintStringNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.12 PrintValueNode()**

```
int PrintValueNode (
    CNodePtr node,
    unsigned int level )
```

**15.263.2.13 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr cam )
```

**15.263.3 Variable Documentation****15.263.3.1 chosenRead**

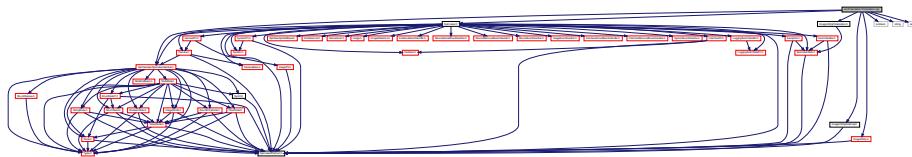
```
const readType chosenRead = VALUE
```

**15.263.3.2 maxChars**

```
const unsigned int maxChars = 35
```

## 15.264 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 ImageUtilityPolarization::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, ImageUtility::SourceDataRange srcDataRange=ImageUtility::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

#### 15.264.1 Function Documentation

##### 15.264.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMapTLDevice )
```

**15.264.1.2 ConfigureStream()**

```
int ConfigureStream (
    INodeMap & nodeMap )
```

**15.264.1.3 CreateAndSaveAolpDolpImages()**

```
int CreateAndSaveAolpDolpImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

**15.264.1.4 CreateAndSaveGlareReducedImage()**

```
int CreateAndSaveGlareReducedImage (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

**15.264.1.5 CreateAndSaveStokesImages()**

```
int CreateAndSaveStokesImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

**15.264.1.6 CreateHeatmapImages()**

```
int CreateHeatmapImages (
    const ImagePtr & mono8Image,
    const string baseFilename,
    gcstring & deviceSerialNumber )
```

**15.264.1.7 CreateNormalizedImage()**

```
int CreateNormalizedImage (
    const ImagePtr & imageToNormalize,
    const string baseFilename,
    gcstring & deviceSerialNumber,
    ImageUtility::SourceDataRange srcDataRange = ImageUtility::IMAGE_DATA_RANGE )
```

**15.264.1.8 ExtractAndSavePolarQuadImages()**

```
int ExtractAndSavePolarQuadImages (
    const ImagePtr & pRawPolarizedImage,
    gcstring & deviceSerialNumber )
```

**15.264.1.9 GetQuadFileNameAppendage()**

```
std::string GetQuadFileNameAppendage (
    const ImageUtilityPolarization::PolarizationQuadrant quadrant )
```

**15.264.1.10 main()**

```
int main (
    int ,
    char ** )
```

**15.264.1.11 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.264.1.12 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr pCam )
```

**15.264.1.13 SaveImage()**

```
int SaveImage (
    const ImagePtr & pImage,
    const string filename,
    gcstring & serialNumber )
```

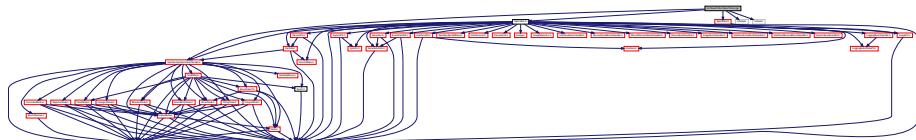
**15.264.2 Variable Documentation**

### 15.264.2.1 isPixelFormatColor

```
bool isPixelFormatColor = false [static]
```

## 15.265 src/SaveToAvi/SaveToAvi.cpp File Reference

Include dependency graph for SaveToAvi.cpp:



### Enumerations

- enum `videoType` {
 `UNCOMPRESSED`,
 `MJPG`,
 `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`

### 15.265.1 Enumeration Type Documentation

#### 15.265.1.1 `videoType`

```
enum videoType
```

##### Enumerator

<code>UNCOMPRESSED</code>	
<code>MJPG</code>	
<code>H264</code>	

## 15.265.2 Function Documentation

### 15.265.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    vector< ImagePtr > & images )
```

### 15.265.2.2 main()

```
int main (
    int ,
    char ** )
```

### 15.265.2.3 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.265.2.4 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

### 15.265.2.5 SaveVectorToVideo()

```
int SaveVectorToVideo (
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDevice,
    vector< ImagePtr > & images )
```

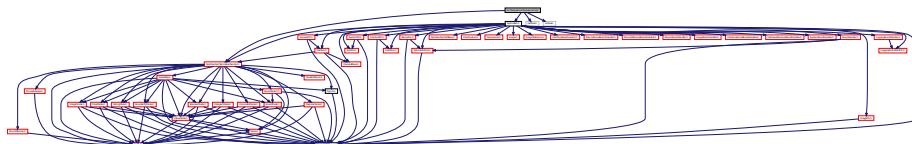
## 15.265.3 Variable Documentation

### 15.265.3.1 chosenVideoType

```
const videoType chosenVideoType = UNCOMPRESSED
```

## 15.266 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 \*\*)

### 15.266.1 Function Documentation

#### 15.266.1.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapGenTL,
    uint64_t timeout )
```

#### 15.266.1.2 ConfigureSequencerPartOne()

```
int ConfigureSequencerPartOne (
    INodeMap & nodeMap )
```

**15.266.1.3 ConfigureSequencerPartTwo()**

```
int ConfigureSequencerPartTwo (
    INodeMap & nodeMap )
```

**15.266.1.4 main()**

```
int main (
    int ,
    char ** )
```

**15.266.1.5 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

**15.266.1.6 PrintRetrieveNodeFailure()**

```
void PrintRetrieveNodeFailure (
    string node,
    string name )
```

**15.266.1.7 ResetSequencer()**

```
int ResetSequencer (
    INodeMap & nodeMap )
```

**15.266.1.8 RunSingleCamera()**

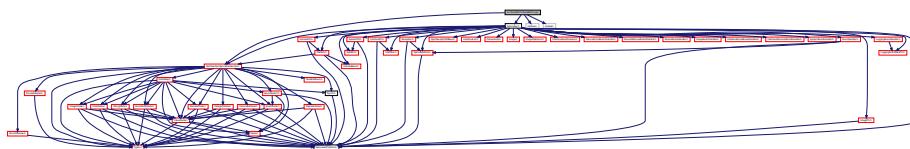
```
int RunSingleCamera (
    CameraPtr pCam )
```

### 15.266.1.9 SetSingleState()

```
int SetSingleState (
    INodeMap & nodeMap,
    unsigned int sequenceNumber,
    int64_t widthToSet,
    int64_t heightToSet,
    double exposureTimeToSet,
    double gainToSet )
```

## 15.267 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 \*\*)

### 15.267.1 Macro Definition Documentation

### 15.267.1.1 COM\_PORT\_COUNT\_MAX

```
#define COM_PORT_COUNT_MAX 256
```

### 15.267.1.2 DATA\_BITS

```
#define DATA_BITS 8
```

### 15.267.1.3 MILLISECOND

```
#define MILLISECOND 1000
```

### 15.267.1.4 SERIAL\_PORT\_BAUD\_RATE

```
#define SERIAL_PORT_BAUD_RATE 19200
```

### 15.267.1.5 SERIAL\_PORT\_COMMUNICATION\_TIMEOUT\_MILLISECOND

```
#define SERIAL_PORT_COMMUNICATION_TIMEOUT_MILLISECOND 1000
```

### 15.267.1.6 SERIAL\_PORT\_DELAY

```
#define SERIAL_PORT_DELAY 1500
```

### 15.267.1.7 SERIAL\_PORT\_PARITY\_BITS

```
#define SERIAL_PORT_PARITY_BITS 0
```

### 15.267.1.8 SERIAL\_PORT\_STOP\_BITS

```
#define SERIAL_PORT_STOP_BITS 0
```

### 15.267.1.9 TWO\_SECOND\_DELAY

```
#define TWO_SECOND_DELAY 2000
```

## 15.267.2 Function Documentation

### 15.267.2.1 CleanUp()

```
int CleanUp (
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

### 15.267.2.2 ConfigureDevice()

```
int ConfigureDevice (
    CameraPtr pCam,
    HANDLE & hFileHandle )
```

### 15.267.2.3 main()

```
int main (
    int ,
    char ** )
```

### 15.267.2.4 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.267.2.5 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

### 15.267.2.6 SerialRx()

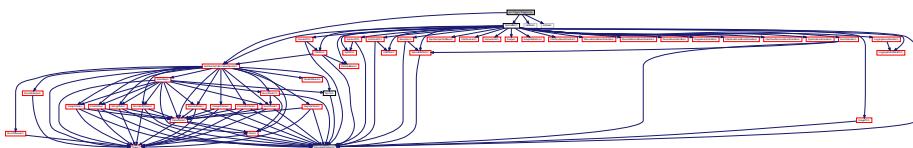
```
int SerialRx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

### 15.267.2.7 SerialTx()

```
int SerialTx (
    CameraPtr pCam,
    INodeMap & nodeMap,
    HANDLE & hFileHandle )
```

## 15.268 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

## 15.268.1 Enumeration Type Documentation

### 15.268.1.1 triggerType

```
enum triggerType
```

**Enumerator**

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

## 15.268.2 Function Documentation

### 15.268.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam,
    INodeMap & nodeMap,
    INodeMap & nodeMapTLDDevice )
```

### 15.268.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    INodeMap & nodeMap )
```

### 15.268.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    INodeMap & nodeMap,
    CameraPtr pCam )
```

### 15.268.2.4 main()

```
int main (
    int ,
    char ** )
```

### 15.268.2.5 PrintDeviceInfo()

```
int PrintDeviceInfo (
    INodeMap & nodeMap )
```

### 15.268.2.6 ResetTrigger()

```
int ResetTrigger (
    INodeMap & nodeMap )
```

### 15.268.2.7 RunSingleCamera()

```
int RunSingleCamera (
    CameraPtr pCam )
```

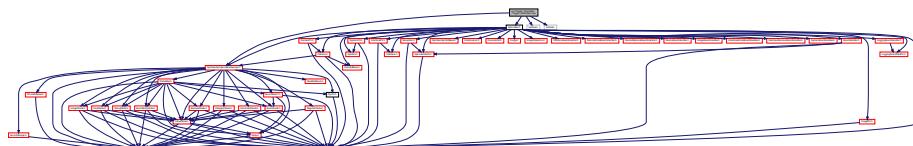
## 15.268.3 Variable Documentation

### 15.268.3.1 chosenTrigger

```
const triggerType chosenTrigger = SOFTWARE
```

## 15.269 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

### 15.269.1 Enumeration Type Documentation

#### 15.269.1.1 triggerType

```
enum triggerType
```

Enumerator

SOFTWARE	
HARDWARE	
SOFTWARE	
HARDWARE	

### 15.269.2 Function Documentation

#### 15.269.2.1 AcquireImages()

```
int AcquireImages (
    CameraPtr pCam )
```

#### 15.269.2.2 ConfigureTrigger()

```
int ConfigureTrigger (
    CameraPtr pCam )
```

#### 15.269.2.3 GrabNextImageByTrigger()

```
int GrabNextImageByTrigger (
    CameraPtr pCam,
    ImagePtr & pResultImage )
```

**15.269.2.4 main()**

```
int main (
    int ,
    char ** )
```

**15.269.2.5 PrintDeviceInfo()**

```
int PrintDeviceInfo (
    CameraPtr pCam )
```

**15.269.2.6 ResetTrigger()**

```
int ResetTrigger (
    CameraPtr pCam )
```

**15.269.2.7 RunSingleCamera()**

```
int RunSingleCamera (
    CameraPtr pCam )
```

**15.269.3 Variable Documentation****15.269.3.1 chosenTrigger**

```
const triggerType chosenTrigger = SOFTWARE
```

# Chapter 16

## Example Documentation

### 16.1 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.

### 16.2 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.

### 16.3 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.

## 16.4 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.

## 16.5 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.

## 16.6 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>

## 16.7 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.

## 16.8 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](#).

## 16.9 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.

## 16.10 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.

## 16.11 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.

---

## 16.12 GenTLInfo\_QuickSpin.cpp

[GenTLInfo\\_QuickSpin.cpp](#) shows how to access node information from interfaces and cameras in C++ with the QuickSpin API. QuickSpin is a subset of [Spinnaker](#) that eases access to camera information via direct node access. If you're not already familiar with the basics of [Spinnaker](#), we suggest starting with the Enumeration\_QuickSpin example.

The example demonstrates the retrieval of information from interface, transport layer device, transport layer stream, and application layer nodes. The retrieval of information of different data types is also touched on.

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.

## 16.13 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.

## 16.14 HighDynamicRange.cpp

This example shows how to set High Dynamic Range (HDR) if it is available on the camera.

## 16.15 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](#).

## 16.16 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.

## 16.17 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.

## 16.18 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.

## 16.19 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.

## 16.20 LogicBlock.cpp

[LogicBlock.cpp](#) shows how to use logic blocks to detect missing triggers and refire. 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->

## 16.21 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.

## 16.22 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.

## 16.23 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.

## 16.24 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-polarized-camera>

## 16.25 SaveToAvi.cpp

SaveToAvi.cpp shows how to create an 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.

## 16.26 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.

## 16.27 SerialRxTx.cpp

SerialRxTx.cpp shows how to communicate using Serial ports. It sets serial port settings in Spinnaker, open and operate File access and create Com Port handle. After the setup, it transmits and received simple data. It verifies the transmission by reading data and transmitting data to COM Port.

THIS EXAMPLE ONLY WORKS IN WINDOWS OS

## 16.28 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.

## 16.29 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 >, [670](#)  
    Spinnaker::GenApi, [354](#)

\_Connect  
    CNodeMapRefT< TCameraParams >, [671](#)  
    Spinnaker::GenApi, [354](#)

\_CycleDetectAccesMode  
    Spinnaker::GenApi, [347](#)

\_Destroy  
    CNodeMapRefT< TCameraParams >, [671](#)

\_EAccessMode  
    Spinnaker::GenApi, [347](#)

\_ECachingMode  
    Spinnaker::GenApi, [347](#)

\_ECallbackType  
    Spinnaker::GenApi, [348](#)

\_EDisplayNotation  
    Spinnaker::GenApi, [348](#)

\_EEndianess  
    Spinnaker::GenApi, [348](#)

\_EGenApiSchemaVersion  
    Spinnaker::GenApi, [349](#)

\_EIncMode  
    Spinnaker::GenApi, [349](#)

\_EInputDirection  
    Spinnaker::GenApi, [349](#)

\_EInterfaceType  
    Spinnaker::GenApi, [350](#)

\_ELinkType  
    Spinnaker::GenApi, [350](#)

\_ENameSpace  
    Spinnaker::GenApi, [350](#)

\_ERepresentation  
    Spinnaker::GenApi, [351](#)

\_ESign  
    Spinnaker::GenApi, [351](#)

\_ESlope  
    Spinnaker::GenApi, [351](#)

\_EStandardNameSpace  
    Spinnaker::GenApi, [352](#)

\_EVisibility  
    Spinnaker::GenApi, [352](#)

\_EXMLValidation  
    Spinnaker::GenApi, [352](#)

\_EYesNo  
    Spinnaker::GenApi, [353](#)

\_GetDeviceName  
    CNodeMapRefT< TCameraParams >, [671](#)  
    Spinnaker::GenApi, [354](#)

\_GetNode  
    CNodeMapRefT< TCameraParams >, [671](#)  
    Spinnaker::GenApi, [355](#)

\_GetNodes  
    CNodeMapRefT< TCameraParams >, [671](#)  
    Spinnaker::GenApi, [355](#)

\_GetSupportedSchemaVersions  
    CNodeMapRefT< TCameraParams >, [672](#)  
    Spinnaker::GenApi, [355](#)

\_Initialize  
    CGeneric\_XMLLoaderParams, [630](#)

\_InvalidateNodes  
    CNodeMapRefT< TCameraParams >, [672](#)  
    Spinnaker::GenApi, [355](#)

\_LoadXMLFromFile  
    CNodeMapRefT< TCameraParams >, [672](#)  
    Spinnaker::GenApi, [355](#)

\_LoadXMLFromFileInject  
    CNodeMapRefT< TCameraParams >, [672](#)  
    Spinnaker::GenApi, [355](#)

\_LoadXMLFromString  
    CNodeMapRefT< TCameraParams >, [672](#)  
    Spinnaker::GenApi, [355](#)

\_LoadXMLFromStringInject  
    CNodeMapRefT< TCameraParams >, [673](#)  
    Spinnaker::GenApi, [356](#)

\_LoadXMLFromZIPData  
    CNodeMapRefT< TCameraParams >, [673](#)  
    Spinnaker::GenApi, [356](#)

\_LoadXMLFromZIPFile  
    CNodeMapRefT< TCameraParams >, [673](#)  
    Spinnaker::GenApi, [356](#)

\_Poll  
    CNodeMapRefT< TCameraParams >, [673](#)  
    Spinnaker::GenApi, [356](#)

\_Ptr  
    CNodeMapRefT< TCameraParams >, [674](#)  
    NodeMap, [1001](#)

\_TO\_STRING  
    GCUtilities.h, [1210](#)

\_UndefinedRepresentation  
    Types.h, [1264](#)

\_Undefined  
    Spinnaker::GenApi, [349](#)

\_UndefinedAccesMode  
    Spinnaker::GenApi, [347](#)

\_UndefinedCachingMode  
    Spinnaker::GenApi, [348](#)

\_UndefinedEDisplayNotation

Spinnaker::GenApi, 348  
 ~UndefinedESlope  
     Spinnaker::GenApi, 352  
 ~UndefinedEXMLValidation  
     Spinnaker::GenApi, 353  
 ~UndefinedEndian  
     Spinnaker::GenApi, 349  
 ~UndefinedNameSpace  
     Spinnaker::GenApi, 351  
 ~UndefinedRepresentation  
     Spinnaker::GenApi, 351  
 ~UndefinedSign  
     Spinnaker::GenApi, 351  
 ~UndefinedStandardNameSpace  
     Spinnaker::GenApi, 352  
 ~UndefinedVisibility  
     Spinnaker::GenApi, 352  
 ~UndefinedYesNo  
     Spinnaker::GenApi, 353  
 ~ERR  
     GCUtilities.h, 1209  
 ~LINE\_STR  
     GCUtilities.h, 1210  
 ~LOCATION  
     GCUtilities.h, 1210  
 ~OUTPUT\_FORMATER  
     GCUtilities.h, 1210  
 ~STDC\_CONSTANT\_MACROS  
     GCTypes.h, 1206  
 ~STDC\_LIMIT\_MACROS  
     GCTypes.h, 1206  
 ~TODO  
     GCUtilities.h, 1210  
 ~WARN  
     GCUtilities.h, 1210  
 \_enableDebug  
      FileAccess\_QuickSpin.cpp, 1348  
 \_fileSelector  
      FileAccess\_QuickSpin.cpp, 1348  
 \_npos  
     gcstring, 764  
 \_pCount  
     double\_automap\_t, 715  
     int64\_automap\_t, 923  
 \_pv  
     double\_automap\_t, 715  
     int64\_automap\_t, 923  
 ~AutoLock  
     AutoLock, 400, 401  
 ~BasePtr  
     BasePtr< T, B >, 403  
 ~BooleanNode  
     BooleanNode, 409  
 ~CChunkAdapter  
     CChunkAdapter, 587  
 ~CChunkAdapterDcam  
     CChunkAdapterDcam, 590  
 ~CChunkAdapterGEV  
     CChunkAdapterGEV, 595  
 ~CChunkAdapterGeneric  
     CChunkAdapterGeneric, 593  
 ~CChunkAdapterU3V  
     CChunkAdapterU3V, 597  
 ~CChunkPort  
     CChunkPort, 600  
 ~CEnumerationTRef  
     CEnumerationTRef< EnumT >, 606  
 ~CEventAdapter  
     CEventAdapter, 610  
 ~CEventAdapter1394  
     CEventAdapter1394, 612  
 ~CEventAdapterGEV  
     CEventAdapterGEV, 617  
 ~CEventAdapterGeneric  
     CEventAdapterGeneric, 614  
 ~CEventAdapterU3V  
     CEventAdapterU3V, 619  
 ~CEventPort  
     CEventPort, 622  
 ~CFeatureBag  
     CFeatureBag, 626  
 ~CGlobalLock  
     CGlobalLock, 632  
 ~CGlobalLockUnlocker  
     CGlobalLockUnlocker, 634  
 ~Clock  
     Clock, 648, 650  
 ~CNodeCallback  
     CNodeCallback, 655  
 ~CNodeMapFactory  
     CNodeMapFactory, 659  
 ~CNodeMapRefT  
     CNodeMapRefT< TCameraParams >, 670  
 ~CPointer  
     CPointer< T, B >, 680  
 ~CPortImpl  
     CPortImpl, 685  
 ~CPortWriteList  
     CPortWriteList, 688  
 ~CRegisterPortImpl  
     CRegisterPortImpl, 692  
 ~CSelectorSet  
     CSelectorSet, 696  
 ~Camera  
     Camera, 437  
 ~CameraBase  
     CameraBase, 563  
 ~CameraList  
     CameraList, 577  
 ~CategoryNode  
     CategoryNode, 585  
 ~ChunkData  
     ChunkData, 638  
 ~CommandNode  
     CommandNode, 676  
 ~DeviceArrivalEventHandler

DeviceArrivalEventHandler, 704  
~DeviceEventHandler  
    DeviceEventHandler, 706  
~DeviceEventHandlerImpl  
    DeviceEventHandlerImpl, 709  
~DeviceRemovalEventHandler  
    DeviceRemovalEventHandler, 712  
~EnumEntryNode  
    EnumEntryNode, 724  
~EnumNode  
    EnumNode, 728  
~EventHandler  
    EventHandler, 737  
~Exception  
    Exception, 743  
~FileProtocolAdapter  
    FileProtocolAdapter, 747  
~FloatNode  
    FloatNode, 753  
~FloatRegNode  
    FloatRegNode, 759  
~ICameraBase  
    ICameraBase, 786  
~ICameraList  
    ICameraList, 794  
~IChunkData  
    IChunkData, 798  
~IDataStream  
    IDataStream, 806  
~IDevFileStreamBuf  
    IDevFileStreamBuf< CharType, Traits >, 815  
~IDeviceArrivalEventHandler  
    IDeviceArrivalEventHandler, 818  
~IDeviceEventHandler  
    IDeviceEventHandler, 820  
~IDeviceRemovalEventHandler  
    IDeviceRemovalEventHandler, 822  
~IImage  
    IImage, 825  
~IImageEventHandler  
    IImageEventHandler, 836  
~IImageStatistics  
    IImageStatistics, 838  
~IInterface  
    IInterface, 842  
~IInterfaceArrivalEventHandler  
    IInterfaceArrivalEventHandler, 847  
~IInterfaceEventHandler  
    IInterfaceEventHandler, 849  
~IInterfaceList  
    IInterfaceList, 851  
~IInterfaceRemovalEventHandler  
    IInterfaceRemovalEventHandler, 854  
~ILoggingEventHandler  
    ILoggingEventHandler, 856  
~ISystem  
    ISystem, 956  
~ISystemEventHandler  
    ISystemEventHandler, 962  
~Image  
    Image, 861  
~ImageEventHandler  
    ImageEventHandler, 884  
~ImageEventHandlerImpl  
    ImageEventHandlerImpl, 886  
~ImagePtr  
    ImagePtr, 889  
~ImageStatistics  
    ImageStatistics, 892  
~InferenceBoundingBoxResult  
    InferenceBoundingBoxResult, 916  
~IntRegNode  
    IntRegNode, 953  
~IntegerNode  
    IntegerNode, 925  
~Interface  
    Interface, 930  
~InterfaceArrivalEventHandler  
    InterfaceArrivalEventHandler, 935  
~InterfaceEventHandler  
    InterfaceEventHandler, 938  
~InterfaceEventHandlerImpl  
    InterfaceEventHandlerImpl, 941  
~InterfaceList  
    InterfaceList, 944  
~InterfaceRemovalEventHandler  
    InterfaceRemovalEventHandler, 950  
~Lock  
    LockableObject< Object >::Lock, 967  
~LoggingEventData  
    LoggingEventData, 970  
~LoggingEventHandler  
    LoggingEventHandler, 976  
~Node  
    Node, 985  
~NodeMap  
    NodeMap, 996  
~ODevFileStreamBuf  
    ODevFileStreamBuf< CharType, Traits >, 1006  
~PortNode  
    PortNode, 1012  
~PortRecorder  
    PortRecorder, 1017  
~PortReplay  
    PortReplay, 1021  
~RegisterNode  
    RegisterNode, 1026  
~SpinVideo  
    SpinVideo, 1032  
~StringNode  
    StringNode, 1038  
~StringRegNode  
    StringRegNode, 1042  
~System  
    System, 1045  
~SystemEventHandler

SystemEventHandler, 1054  
 ~SystemEventHandlerImpl  
     SystemEventHandlerImpl, 1057  
 ~SystemPtr  
     SystemPtr, 1060  
 ~TransportLayerDevice  
     TransportLayerDevice, 1065  
 ~TransportLayerInterface  
     TransportLayerInterface, 1075  
 ~TransportLayerStream  
     TransportLayerStream, 1086  
 ~TransportLayerSystem  
     TransportLayerSystem, 1094  
 ~ValueNode  
     ValueNode, 1104  
 ~double\_automap\_t  
     double\_automap\_t, 713  
 ~gcstring  
     gcstring, 764  
 ~int64\_automap\_t  
     int64\_automap\_t, 921  
  
 AasRoiEnable  
     Camera, 437  
 AasRoiHeight  
     Camera, 438  
 AasRoiOffsetX  
     Camera, 438  
 AasRoiOffsetY  
     Camera, 438  
 AasRoiWidth  
     Camera, 438  
 ABSOLUTE\_DATA\_RANGE  
     ImageUtility, 898  
 ABSOLUTE\_MIN\_IMAGE\_MAX  
     ImageUtility, 898  
 AcquireImages  
     Acquisition.cpp, 1282  
     AcquisitionMultipleThread.cpp, 1326  
     ActionCommand.cpp, 1328  
     BufferHandling.cpp, 1330  
     ChunkData.cpp, 1333  
     CounterAndTimer.cpp, 1334  
     DeviceEvents.cpp, 1337  
     Exposure.cpp, 1342  
     Exposure\_QuickSpin.cpp, 1344  
     FileAccess\_QuickSpin.cpp, 1345  
     GigEVisionPerformance.cpp, 1352  
     ImageEvents.cpp, 1360  
     ImageFormatControl.cpp, 1362  
     ImageFormatControl\_QuickSpin.cpp, 1363  
     Inference.cpp, 1366  
     LogicBlock.cpp, 1372  
     LookupTable.cpp, 1374  
     Polarization.cpp, 1381  
     SaveToAvi.cpp, 1385  
     Sequencer.cpp, 1386  
     Trigger.cpp, 1393  
     Trigger\_QuickSpin.cpp, 1395  
  
 Acquisition.cpp  
     AcquireImages, 1282  
     main, 1282  
     PrintDeviceInfo, 1282  
     RunSingleCamera, 1282  
 AcquisitionAbort  
     Camera, 439  
 AcquisitionArm  
     Camera, 439  
 AcquisitionBurstFrameCount  
     Camera, 439  
 AcquisitionFrameCount  
     Camera, 439  
 AcquisitionFrameRate  
     Camera, 440  
 AcquisitionFrameRateEnable  
     Camera, 440  
 AcquisitionLineRate  
     Camera, 440  
 AcquisitionMode  
     Camera, 440  
 AcquisitionMode\_Continuous  
     Spinnaker, 230  
 AcquisitionMode\_MultiFrame  
     Spinnaker, 230  
 AcquisitionMode\_SingleFrame  
     Spinnaker, 230  
 AcquisitionModeEnums  
     Spinnaker, 230  
 AcquisitionMultipleCameraRecovery.cpp  
     cameraGrabInfoMap, 1326  
     ConfigureCamera, 1324  
     ConfigureUserSet1, 1324  
     GetDeviceSerial, 1325  
     globalCamList, 1326  
     main, 1325  
     PrintExampleStatistics, 1325  
     RefreshCameraList, 1325  
     ResetCameraUserSetToDefault, 1325  
     SleepyWrapper, 1325  
 AcquisitionMultipleThread.cpp  
     AcquireImages, 1326  
     main, 1326  
     PrintDeviceInfo, 1327  
     RunMultipleCameras, 1327  
 AcquisitionResultingFrameRate  
     Camera, 440  
 AcquisitionStart  
     Camera, 441  
 AcquisitionStatus  
     Camera, 441  
 AcquisitionStatusSelector  
     Camera, 441  
 AcquisitionStatusSelector\_AcquisitionActive  
     Spinnaker, 230  
 AcquisitionStatusSelector\_AcquisitionTransfer  
     Spinnaker, 231  
 AcquisitionStatusSelector\_AcquisitionTriggerWait

Spinnaker, 230  
AcquisitionStatusSelector\_ExposureActive  
    Spinnaker, 231  
AcquisitionStatusSelector\_FrameActive  
    Spinnaker, 231  
AcquisitionStatusSelector\_FrameTriggerWait  
    Spinnaker, 231  
AcquisitionStatusSelectorEnums  
    Spinnaker, 230  
AcquisitionStop  
    Camera, 441  
ACTION\_COMMAND\_STATUS\_ACTION\_LATE  
    Spinnaker, 231  
ACTION\_COMMAND\_STATUS\_ERROR  
    Spinnaker, 231  
ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME  
    Spinnaker, 231  
ACTION\_COMMAND\_STATUS\_OK  
    Spinnaker, 231  
ACTION\_COMMAND\_STATUS\_OVERFLOW  
    Spinnaker, 231  
ActionCommand  
    TransportLayerInterface, 1076  
ActionCommand.cpp  
    AcquireImages, 1328  
    ConfigureActionControl, 1328  
    ConfigureChunkData, 1328  
    ConfigureIEEE1588, 1328  
    ConfigureInterface, 1328  
    ConfigureOtherNodes, 1328  
    ConfigureTrigger, 1328  
    main, 1329  
    PrintDeviceInfo, 1329  
    RunMultipleCameras, 1329  
    SleepyWrapper, 1329  
ActionCommandResult, 393  
    DeviceAddress, 393  
    Status, 393  
ActionCommandStatus  
    Spinnaker, 231  
ActionDeviceKey  
    Camera, 441  
ActionGroupKey  
    Camera, 441  
ActionGroupMask  
    Camera, 442  
ActionQueueSize  
    Camera, 442  
ActionSelector  
    Camera, 442  
ActionUnconditionalMode  
    Camera, 442  
ActionUnconditionalMode\_Off  
    Spinnaker, 231  
ActionUnconditionalMode\_On  
    Spinnaker, 231  
ActionUnconditionalModeEnums  
    Spinnaker, 231  
AdapterConfig, 183  
AdapterConfigErr, 184  
AutoPopulateAdapterInfo, 184  
AutoPopulateAdvancedProperties, 184  
ConfigureAdapter, 184  
GetAuto10GDesc, 185  
GetAutoGigabitDesc, 185  
GetAutoStartIp, 185  
GetAutoSubnetMask, 185  
GetAutoSubnetMaskLength, 185  
GetConfigLogFileName, 185  
GetEnumerationLogFileName, 185  
GetMaxIpAddress, 186  
GetMinIpAddress, 186  
GetSubnetMaskLength, 186  
HOST\_ADDRESS\_ZERO, 184  
IP\_ADDRESS\_INVALID, 184  
IP\_ADDRESS\_IS\_NOT\_V4, 184  
IP\_ADDRESS\_TOO\_LARGE, 184  
IP\_ADDRESS\_TOO\_SMALL, 184  
IsOnSameSubnet, 186  
IsValidIpAddress, 186  
IsValidSubnetMask, 186  
PopulateAdapterIpInfo, 186  
RetrieveAllAdapters, 187  
SUBNET\_MASK\_INVALID, 184  
VALID\_SUBNET\_NOT\_FOUND, 184  
ValidateIpAddress, 187  
AdapterConfig.h  
    ADAPTERCONFIG\_API, 1109  
ADAPTERCONFIG\_API  
    AdapterConfig.h, 1109  
AdapterConfigErr  
    AdapterConfig, 184  
AdapterConfigException, 394  
    AdapterConfigException, 394  
    ErrCode, 395  
    GetParamStr, 395  
adapterDescription  
    AdapterInfo, 396  
adapterGUID  
    AdapterInfo, 396  
AdapterInfo, 395  
    adapterDescription, 396  
    adapterGUID, 396  
    AdapterInfo, 396  
    adapterMACAddress, 396  
    adapterName, 397  
    dhcpEnabled, 397  
    ipInfo, 397  
    jumboPackets, 397  
    jumboPacketsRegKey, 397  
    jumboPacketValidValues, 397  
    receiveBuffers, 397  
    receiveBuffersMax, 397  
    receiveBuffersMin, 398  
    receiveBuffersRegKey, 398  
    receiveBuffersStep, 398

transmitBuffers, 398  
 transmitBuffersMax, 398  
 transmitBuffersMin, 398  
 transmitBuffersRegKey, 398  
 transmitBuffersStep, 398  
 adapterMACAddress  
     AdapterInfo, 396  
 adapterName  
     AdapterInfo, 397  
 AdaptiveCompressionEnable  
     Camera, 442  
 AdcBitDepth  
     Camera, 442  
 AdcBitDepth\_Bit10  
     Spinnaker, 232  
 AdcBitDepth\_Bit12  
     Spinnaker, 232  
 AdcBitDepth\_Bit14  
     Spinnaker, 232  
 AdcBitDepth\_Bit8  
     Spinnaker, 232  
 AdcBitDepthEnums  
     Spinnaker, 231  
 AddInjectionData  
     CNodeMapFactory, 661  
 Address  
     Spinnaker::GenApi, 379  
 ADOBE\_DEFLATE  
     TIFFOption, 1062  
 AnnounceImage  
     IDataStream, 807  
 aPAUSEMACCtrlFramesReceived  
     Camera, 443  
 aPAUSEMACCtrlFramesTransmitted  
     Camera, 443  
 Append  
     CameraList, 578  
     ICameraList, 794  
     SpinVideo, 1032  
 append  
     gcstring, 764  
 ApplyStyleSheet  
     CNodeMapFactory, 661  
 argBayerRG  
     GigEVisionPerformance.cpp, 1354  
 argDuration  
     GigEVisionPerformance.cpp, 1354  
 argMaxFrames  
     GigEVisionPerformance.cpp, 1354  
 argNumImages  
     GigEVisionPerformance.cpp, 1354  
 argPacketDelay  
     GigEVisionPerformance.cpp, 1354  
 argPacketSize  
     GigEVisionPerformance.cpp, 1354  
 argPrintUsage  
     GigEVisionPerformance.cpp, 1355  
 argRelease  
     GigEVisionPerformance.cpp, 1355  
     GigEVisionPerformance.cpp, 1355  
     argUserSetFrames  
         GigEVisionPerformance.cpp, 1355  
 arrayLabelClassification  
     Inference.cpp, 1369  
 arrayLabelDetection  
     Inference.cpp, 1369  
 assign  
     gcstring, 764, 765  
 attach  
     FileProtocolAdapter, 747  
 AttachBuffer  
     CChunkAdapter, 587  
     CChunkAdapterDcam, 590  
     CChunkAdapterGeneric, 593  
     CChunkAdapterGEV, 595  
     CChunkAdapterU3V, 598  
     IDataStream, 807  
 AttachChunk  
     CChunkPort, 601  
 AttachEvent  
     CEventPort, 622  
 AttachNode  
     CEventPort, 622  
 AttachNodeMap  
     CChunkAdapter, 587  
     CEventAdapter, 610  
 AttachPort  
     CChunkPort, 601  
 AttachStatistics\_t, 399  
     NumAttachedChunks, 399  
     NumChunkPorts, 399  
     NumChunks, 399  
 AutoAlgorithmSelector  
     Camera, 443  
 AutoAlgorithmSelector\_Ae  
     Spinnaker, 232  
 AutoAlgorithmSelector\_Awb  
     Spinnaker, 232  
 AutoAlgorithmSelectorEnums  
     Spinnaker, 232  
 AutoExposureControlLoopDamping  
     Camera, 443  
 AutoExposureControlPriority  
     Camera, 443  
 AutoExposureControlPriority\_ExposureTime  
     Spinnaker, 232  
 AutoExposureControlPriority\_Gain  
     Spinnaker, 232  
 AutoExposureControlPriorityEnums  
     Spinnaker, 232  
 AutoExposureEVCompensation  
     Camera, 444  
 AutoExposureExposureTimeLowerLimit  
     Camera, 444  
 AutoExposureExposureTimeUpperLimit  
     Camera, 444  
 AutoExposureGainLowerLimit

Camera, 444  
AutoExposureGainUpperLimit  
    Camera, 445  
AutoExposureGreyValueLowerLimit  
    Camera, 445  
AutoExposureGreyValueUpperLimit  
    Camera, 445  
AutoExposureLightingMode  
    Camera, 445  
AutoExposureLightingMode\_AutoDetect  
    Spinnaker, 233  
AutoExposureLightingMode\_Backlight  
    Spinnaker, 233  
AutoExposureLightingMode\_Frontlight  
    Spinnaker, 233  
AutoExposureLightingMode\_Normal  
    Spinnaker, 233  
AutoExposureLightingModeEnums  
    Spinnaker, 232  
AutoExposureMeteringMode  
    Camera, 446  
AutoExposureMeteringMode\_Average  
    Spinnaker, 233  
AutoExposureMeteringMode\_CenterWeighted  
    Spinnaker, 233  
AutoExposureMeteringMode\_HistogramPeak  
    Spinnaker, 233  
AutoExposureMeteringMode\_Partial  
    Spinnaker, 233  
AutoExposureMeteringMode\_Spot  
    Spinnaker, 233  
AutoExposureMeteringModeEnums  
    Spinnaker, 233  
AutoExposureTargetGreyValue  
    Camera, 446  
AutoExposureTargetGreyValueAuto  
    Camera, 446  
AutoExposureTargetGreyValueAuto\_Continuous  
    Spinnaker, 233  
AutoExposureTargetGreyValueAuto\_Off  
    Spinnaker, 233  
AutoExposureTargetGreyValueAutoEnums  
    Spinnaker, 233  
AutoLock, 400  
    ~AutoLock, 400, 401  
    AutoLock, 400, 401  
Automatic  
    Spinnaker::GenApi, 352  
AutoPopulateAdapterInfo  
    AdapterConfig, 184  
AutoPopulateAdvancedProperties  
    AdapterConfig, 184  
AutoVector Class, 94  
AVI Recorder Class, 40  
AVIOption, 401  
    AVIOption, 401  
    frameRate, 402  
    reserved, 402  
BalanceRatio  
    Camera, 447  
BalanceRatioSelector  
    Camera, 447  
BalanceRatioSelector\_Blue  
    Spinnaker, 234  
BalanceRatioSelector\_Red  
    Spinnaker, 234  
BalanceRatioSelectorEnums  
    Spinnaker, 234  
BalanceWhiteAuto  
    Camera, 447  
BalanceWhiteAuto\_Continuous  
    Spinnaker, 234  
BalanceWhiteAuto\_Off  
    Spinnaker, 234  
BalanceWhiteAuto\_Once  
    Spinnaker, 234  
BalanceWhiteAutoDamping  
    Camera, 447  
BalanceWhiteAutoEnums  
    Spinnaker, 234  
BalanceWhiteAutoLowerLimit  
    Camera, 448  
BalanceWhiteAutoProfile  
    Camera, 448  
BalanceWhiteAutoProfile\_Indoor  
    Spinnaker, 234  
BalanceWhiteAutoProfile\_Outdoor  
    Spinnaker, 234  
BalanceWhiteAutoProfileEnums  
    Spinnaker, 234  
BalanceWhiteAutoUpperLimit  
    Camera, 448  
BasePtr  
    BasePtr< T, B >, 403  
BasePtr Class, 41  
BasePtr< T, B >, 402  
    ~BasePtr, 403  
    BasePtr, 403  
    get, 404  
    isValid, 404  
    m\_pT, 406  
    operator bool, 404  
    operator T\*, 404  
    operator->, 404  
    operator=, 405  
    operator==, 405, 406  
BeginAcquisition  
    CameraBase, 564  
    ICameraBase, 786  
Beginner  
    Spinnaker::GenApi, 352  
BigEndian  
    Spinnaker::GenApi, 349  
BILINEAR  
    Spinnaker, 246  
binaryFile

PGMOOption, 1008  
 PPMOption, 1024  
 BinningHorizontal  
     Camera, 448  
 BinningHorizontalMode  
     Camera, 449  
 BinningHorizontalMode\_Average  
     Spinnaker, 235  
 BinningHorizontalMode\_Sum  
     Spinnaker, 235  
 BinningHorizontalModeEnums  
     Spinnaker, 235  
 BinningSelector  
     Camera, 449  
 BinningSelector\_All  
     Spinnaker, 235  
 BinningSelector\_ISP  
     Spinnaker, 235  
 BinningSelector\_Sensor  
     Spinnaker, 235  
 BinningSelectorEnums  
     Spinnaker, 235  
 BinningVertical  
     Camera, 449  
 BinningVerticalMode  
     Camera, 449  
 BinningVerticalMode\_Average  
     Spinnaker, 235  
 BinningVerticalMode\_Sum  
     Spinnaker, 235  
 BinningVerticalModeEnums  
     Spinnaker, 235  
 bitrate  
     H264Option, 783  
 BlackLevel  
     Camera, 450  
 BlackLevelAuto  
     Camera, 450  
 BlackLevelAuto\_Continuous  
     Spinnaker, 236  
 BlackLevelAuto\_Off  
     Spinnaker, 236  
 BlackLevelAuto\_Once  
     Spinnaker, 236  
 BlackLevelAutoBalance  
     Camera, 450  
 BlackLevelAutoBalance\_Continuous  
     Spinnaker, 236  
 BlackLevelAutoBalance\_Off  
     Spinnaker, 236  
 BlackLevelAutoBalance\_Once  
     Spinnaker, 236  
 BlackLevelAutoBalanceEnums  
     Spinnaker, 236  
 BlackLevelAutoEnums  
     Spinnaker, 236  
 BlackLevelClampingEnable  
     Camera, 450  
 BlackLevelRaw  
     Camera, 450  
 BlackLevelSelector  
     Camera, 451  
 BlackLevelSelector\_All  
     Spinnaker, 236  
 BlackLevelSelector\_Analog  
     Spinnaker, 236  
 BlackLevelSelector\_Digital  
     Spinnaker, 236  
 BlackLevelSelectorEnums  
     Spinnaker, 236  
 BlockId  
     GVCP\_EVENT\_ITEM, 773  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 775  
 BlockId64High  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 775  
 BlockId64Low  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776  
 BLUE  
     Spinnaker, 313  
 BMP  
     Spinnaker, 282  
 BMPOption, 406  
     BMPOption, 407  
     indexedColor\_8bit, 407  
     reserved, 407  
 Boolean  
     Spinnaker::GenApi, 351  
 BooleanNode, 408  
     ~BooleanNode, 409  
     BooleanNode, 409  
     GetValue, 409  
     operator=, 410  
     SetReference, 410  
     SetValue, 410  
 BooleanNode Class, 97  
 bottomRightXCoord  
     InferenceBoxRect, 919  
     InferenceBoxRotatedRect, 920  
 bottomRightYCoord  
     InferenceBoxRect, 919  
     InferenceBoxRotatedRect, 920  
 boxType  
     InferenceBoundingBox, 914  
 BUFFER\_OWNERSHIP\_SYSTEM  
     Spinnaker, 237  
 BUFFER\_OWNERSHIP\_USER  
     Spinnaker, 237  
 BufferHandling.cpp  
     AcquireImages, 1330  
     ConfigureTrigger, 1330  
     GrabNextImageByTrigger, 1331  
     k\_numLoops, 1330  
     main, 1331  
     numBuffers, 1330  
     PrintDeviceInfo, 1331  
     ResetTrigger, 1331

RunSingleCamera, 1331  
SleepyWrapper, 1331  
z\_numTriggers, 1330  
BufferOwnership  
    Spinnaker, 237  
build  
    LibraryVersion, 966  
  
c\_str  
    gcstring, 765  
CacheChunkData  
    PortNode, 1012  
        Spinnaker::GenApi, 356  
CacheUsage\_Automatic  
    Spinnaker::GenApi, 353  
CacheUsage\_ForceRead  
    Spinnaker::GenApi, 353  
CacheUsage\_ForceWrite  
    Spinnaker::GenApi, 353  
CacheUsage\_Ignore  
    Spinnaker::GenApi, 353  
CalculateStatistics  
    IImage, 825  
    Image, 862  
CallbackHandleType  
    Spinnaker::GenApi, 344  
Camera, 411  
    ~Camera, 437  
    AasRoiEnable, 437  
    AasRoiHeight, 438  
    AasRoiOffsetX, 438  
    AasRoiOffsetY, 438  
    AasRoiWidth, 438  
    AcquisitionAbort, 439  
    AcquisitionArm, 439  
    AcquisitionBurstFrameCount, 439  
    AcquisitionFrameCount, 439  
    AcquisitionFrameRate, 440  
    AcquisitionFrameRateEnable, 440  
    AcquisitionLineRate, 440  
    AcquisitionMode, 440  
    AcquisitionResultingFrameRate, 440  
    AcquisitionStart, 441  
    AcquisitionStatus, 441  
    AcquisitionStatusSelector, 441  
    AcquisitionStop, 441  
    ActionDeviceKey, 441  
    ActionGroupKey, 441  
    ActionGroupMask, 442  
    ActionQueueSize, 442  
    ActionSelector, 442  
    ActionUnconditionalMode, 442  
    AdaptiveCompressionEnable, 442  
    AdcBitDepth, 442  
    aPAUSEMACCtrlFramesReceived, 443  
    aPAUSEMACCtrlFramesTransmitted, 443  
    AutoAlgorithmSelector, 443  
    AutoExposureControlLoopDamping, 443  
    AutoExposureControlPriority, 443  
    AutoExposureEVCompensation, 444  
    AutoExposureExposureTimeLowerLimit, 444  
    AutoExposureExposureTimeUpperLimit, 444  
    AutoExposureGainLowerLimit, 444  
    AutoExposureGainUpperLimit, 445  
    AutoExposureGreyValueLowerLimit, 445  
    AutoExposureGreyValueUpperLimit, 445  
    AutoExposureLightingMode, 445  
    AutoExposureMeteringMode, 446  
    AutoExposureTargetGreyValue, 446  
    AutoExposureTargetGreyValueAuto, 446  
    BalanceRatio, 447  
    BalanceRatioSelector, 447  
    BalanceWhiteAuto, 447  
    BalanceWhiteAutoDamping, 447  
    BalanceWhiteAutoLowerLimit, 448  
    BalanceWhiteAutoProfile, 448  
    BalanceWhiteAutoUpperLimit, 448  
    BinningHorizontal, 448  
    BinningHorizontalMode, 449  
    BinningSelector, 449  
    BinningVertical, 449  
    BinningVerticalMode, 449  
    BlackLevel, 450  
    BlackLevelAuto, 450  
    BlackLevelAutoBalance, 450  
    BlackLevelClampingEnable, 450  
    BlackLevelRaw, 450  
    BlackLevelSelector, 451  
    Camera, 437  
    ChunkBlackLevel, 451  
    ChunkBlackLevelSelector, 451  
    ChunkCounterSelector, 451  
    ChunkCounterValue, 451  
    ChunkCRC, 452  
    ChunkEnable, 452  
    ChunkEncoderSelector, 452  
    ChunkEncoderStatus, 452  
    ChunkEncoderValue, 452  
    ChunkExposureEndLineStatusAll, 452  
    ChunkExposureTime, 453  
    ChunkExposureTimeSelector, 453  
    ChunkFrameID, 453  
    ChunkGain, 453  
    ChunkGainSelector, 453  
    ChunkHeight, 453  
    ChunkImage, 454  
    ChunkImageComponent, 454  
    ChunkInferenceBoundingBoxResult, 454  
    ChunkInferenceConfidence, 454  
    ChunkInferenceFrameId, 454  
    ChunkInferenceResult, 454  
    ChunkLinePitch, 455  
    ChunkLineStatusAll, 455  
    ChunkModeActive, 455  
    ChunkOffsetX, 455  
    ChunkOffsetY, 455  
    ChunkPartSelector, 455

ChunkPixelDynamicRangeMax, 456  
ChunkPixelDynamicRangeMin, 456  
ChunkPixelFormat, 456  
ChunkRegionID, 456  
ChunkScan3dAxisMax, 456  
ChunkScan3dAxisMin, 456  
ChunkScan3dCoordinateOffset, 457  
ChunkScan3dCoordinateReferenceSelector, 457  
ChunkScan3dCoordinateReferenceValue, 457  
ChunkScan3dCoordinateScale, 457  
ChunkScan3dCoordinateSelector, 457  
ChunkScan3dCoordinateSystem, 457  
ChunkScan3dCoordinateSystemReference, 458  
ChunkScan3dCoordinateTransformSelector, 458  
ChunkScan3dDistanceUnit, 458  
ChunkScan3dInvalidDataFlag, 458  
ChunkScan3dInvalidDataValue, 458  
ChunkScan3dOutputMode, 458  
ChunkScan3dTransformValue, 459  
ChunkScanLineSelector, 459  
ChunkSelector, 459  
ChunkSequencerSetActive, 459  
ChunkSerialData, 459  
ChunkSerialDataLength, 459  
ChunkSerialReceiveOverflow, 460  
ChunkSourceID, 460  
ChunkStreamChannelID, 460  
ChunkTimerSelector, 460  
ChunkTimerValue, 460  
ChunkTimestamp, 460  
ChunkTimestampLatchValue, 461  
ChunkTransferBlockID, 461  
ChunkTransferQueueCurrentBlockCount, 461  
ChunkTransferStreamID, 461  
ChunkWidth, 461  
CIConfiguration, 461  
CITimeSlotsCount, 462  
ColorTransformationEnable, 462  
ColorTransformationSelector, 462  
ColorTransformationValue, 462  
ColorTransformationValueSelector, 462  
CompressionRatio, 463  
CounterDelay, 463  
CounterDuration, 463  
CounterEventActivation, 463  
CounterEventSource, 463  
CounterReset, 464  
CounterResetActivation, 464  
CounterResetSource, 464  
CounterSelector, 464  
CounterStatus, 464  
CounterTriggerActivation, 464  
CounterTriggerSource, 465  
CounterValue, 465  
CounterValueAtReset, 465  
CxpConnectionSelector, 465  
CxpConnectionTestErrorCount, 465  
CxpConnectionTestMode, 465  
CxpConnectionTestPacketCount, 466  
CxpLinkConfiguration, 466  
CxpLinkConfigurationPreferred, 466  
CxpLinkConfigurationStatus, 466  
CxpPoCxpAuto, 466  
CxpPoCxpStatus, 466  
CxpPoCxpTripReset, 467  
CxpPoCxpTurnOff, 467  
DecimationHorizontal, 467  
DecimationHorizontalMode, 467  
DecimationSelector, 467  
DecimationVertical, 468  
DecimationVerticalMode, 468  
DefectCorrectionMode, 468  
DefectCorrectStaticEnable, 468  
DefectTableApply, 469  
DefectTableCoordinateX, 469  
DefectTableCoordinateY, 469  
DefectTableFactoryRestore, 469  
DefectTableIndex, 470  
DefectTablePixelCount, 470  
DefectTableSave, 470  
Deinterlacing, 470  
DeviceCharacterSet, 471  
DeviceClockFrequency, 471  
DeviceClockSelector, 471  
DeviceConnectionSelector, 471  
DeviceConnectionSpeed, 471  
DeviceConnectionStatus, 472  
DeviceEventChannelCount, 472  
DeviceFamilyName, 472  
DeviceFeaturePersistenceEnd, 472  
DeviceFeaturePersistenceStart, 472  
DeviceFirmwareVersion, 473  
DeviceGenCPVersionMajor, 473  
DeviceGenCPVersionMinor, 473  
DeviceID, 473  
DeviceIndicatorMode, 473  
DeviceLinkBandwidthReserve, 473  
DeviceLinkCommandTimeout, 474  
DeviceLinkConnectionCount, 474  
DeviceLinkCurrentThroughput, 474  
DeviceLinkHeartbeatMode, 474  
DeviceLinkHeartbeatTimeout, 474  
DeviceLinkSelector, 475  
DeviceLinkSpeed, 475  
DeviceLinkThroughputLimit, 475  
DeviceLinkThroughputLimitMode, 475  
DeviceManifestEntrySelector, 475  
DeviceManifestPrimaryURL, 476  
DeviceManifestSchemaMajorVersion, 476  
DeviceManifestSchemaMinorVersion, 476  
DeviceManifestSecondaryURL, 476  
DeviceManifestXMLMajorVersion, 476  
DeviceManifestXMLMinorVersion, 476  
DeviceManifestXMLSubMinorVersion, 477  
DeviceManufacturerInfo, 477  
DeviceMaxThroughput, 477

DevicemodelName, 477  
DevicePowerSupplySelector, 477  
DeviceRegistersCheck, 478  
DeviceRegistersEndianness, 478  
DeviceRegistersStreamingEnd, 478  
DeviceRegistersStreamingStart, 478  
DeviceRegistersValid, 478  
DeviceReset, 479  
DeviceScanType, 479  
DeviceSerialNumber, 479  
DeviceSerialPortBaudRate, 479  
DeviceSerialPortSelector, 479  
DeviceSFNCVersionMajor, 480  
DeviceSFNCVersionMinor, 480  
DeviceSFNCVersionSubMinor, 480  
DeviceStreamChannelCount, 480  
DeviceStreamChannelEndianness, 480  
DeviceStreamChannelLink, 481  
DeviceStreamChannelPacketSize, 481  
DeviceStreamChannelSelector, 481  
DeviceStreamChannelType, 481  
DeviceTapGeometry, 481  
DeviceTemperature, 481  
DeviceTemperatureSelector, 482  
DeviceTLType, 482  
DeviceTlVersionMajor, 482  
DeviceTlVersionMinor, 482  
DeviceTlVersionSubMinor, 482  
DeviceType, 483  
DeviceUptime, 483  
DeviceUserID, 483  
DeviceVendorName, 483  
DeviceVersion, 483  
EncoderDivide, 483  
EncoderMode, 484  
EncoderOutputMode, 484  
EncoderReset, 484  
EncoderResetActivation, 484  
EncoderResetSource, 484  
EncoderSelector, 484  
EncoderSourceA, 485  
EncoderSourceB, 485  
EncoderStatus, 485  
EncoderTimeout, 485  
EncoderValue, 485  
EncoderValueAtReset, 485  
EnumerationCount, 486  
EventAcquisitionEnd, 486  
EventAcquisitionEndFrameID, 486  
EventAcquisitionEndTimestamp, 486  
EventAcquisitionError, 486  
EventAcquisitionErrorFrameID, 486  
EventAcquisitionErrorTimestamp, 487  
EventAcquisitionStart, 487  
EventAcquisitionStartFrameID, 487  
EventAcquisitionStartTimestamp, 487  
EventAcquisitionTransferEnd, 487  
EventAcquisitionTransferEndFrameID, 487  
EventAcquisitionTransferEndTimestamp, 488  
EventAcquisitionTransferStart, 488  
EventAcquisitionTransferStartFrameID, 488  
EventAcquisitionTransferStartTimestamp, 488  
EventAcquisitionTrigger, 488  
EventAcquisitionTriggerFrameID, 488  
EventAcquisitionTriggerTimestamp, 489  
EventActionLate, 489  
EventActionLateFrameID, 489  
EventActionLateTimestamp, 489  
EventCounter0End, 489  
EventCounter0EndFrameID, 489  
EventCounter0EndTimestamp, 490  
EventCounter0Start, 490  
EventCounter0StartFrameID, 490  
EventCounter0StartTimestamp, 490  
EventCounter1End, 490  
EventCounter1EndFrameID, 490  
EventCounter1EndTimestamp, 491  
EventCounter1Start, 491  
EventCounter1StartFrameID, 491  
EventCounter1StartTimestamp, 491  
EventEncoder0Restarted, 491  
EventEncoder0RestartedFrameID, 491  
EventEncoder0RestartedTimestamp, 492  
EventEncoder0Stopped, 492  
EventEncoder0StoppedFrameID, 492  
EventEncoder0StoppedTimestamp, 492  
EventEncoder1Restarted, 492  
EventEncoder1RestartedFrameID, 492  
EventEncoder1RestartedTimestamp, 493  
EventEncoder1Stopped, 493  
EventEncoder1StoppedFrameID, 493  
EventEncoder1StoppedTimestamp, 493  
EventError, 493  
EventErrorCode, 493  
EventErrorFrameID, 494  
EventErrorTimestamp, 494  
EventExposureEnd, 494  
EventExposureEndFrameID, 494  
EventExposureEndTimestamp, 494  
EventExposureStart, 494  
EventExposureStartFrameID, 495  
EventExposureStartTimestamp, 495  
EventFrameBurstEnd, 495  
EventFrameBurstEndFrameID, 495  
EventFrameBurstEndTimestamp, 495  
EventFrameBurstStart, 495  
EventFrameBurstStartFrameID, 496  
EventFrameBurstStartTimestamp, 496  
EventFrameEnd, 496  
EventFrameEndFrameID, 496  
EventFrameEndTimestamp, 496  
EventFrameStart, 496  
EventFrameStartFrameID, 497  
EventFrameStartTimestamp, 497  
EventFrameTransferEnd, 497  
EventFrameTransferEndFrameID, 497

EventFrameTransferEndTimestamp, 497  
 EventFrameTransferStart, 497  
 EventFrameTransferStartFrameID, 498  
 EventFrameTransferStartTimestamp, 498  
 EventFrameTrigger, 498  
 EventFrameTriggerFrameID, 498  
 EventFrameTriggerTimestamp, 498  
 EventLine0AnyEdge, 498  
 EventLine0AnyEdgeFrameID, 499  
 EventLine0AnyEdgeTimestamp, 499  
 EventLine0FallingEdge, 499  
 EventLine0FallingEdgeFrameID, 499  
 EventLine0FallingEdgeTimestamp, 499  
 EventLine0RisingEdge, 499  
 EventLine0RisingEdgeFrameID, 500  
 EventLine0RisingEdgeTimestamp, 500  
 EventLine1AnyEdge, 500  
 EventLine1AnyEdgeFrameID, 500  
 EventLine1AnyEdgeTimestamp, 500  
 EventLine1FallingEdge, 500  
 EventLine1FallingEdgeFrameID, 501  
 EventLine1FallingEdgeTimestamp, 501  
 EventLine1RisingEdge, 501  
 EventLine1RisingEdgeFrameID, 501  
 EventLine1RisingEdgeTimestamp, 501  
 EventLinkSpeedChange, 501  
 EventLinkSpeedChangeFrameID, 502  
 EventLinkSpeedChangeTimestamp, 502  
 EventLinkTrigger0, 502  
 EventLinkTrigger0FrameID, 502  
 EventLinkTrigger0Timestamp, 502  
 EventLinkTrigger1, 502  
 EventLinkTrigger1FrameID, 503  
 EventLinkTrigger1Timestamp, 503  
 EventNotification, 503  
 EventSelector, 503  
 EventSequencerSetChange, 503  
 EventSequencerSetChangeFrameID, 503  
 EventSequencerSetChangeTimestamp, 504  
 EventSerialData, 504  
 EventSerialDataLength, 504  
 EventSerialPortReceive, 504  
 EventSerialPortReceiveTimestamp, 504  
 EventSerialReceiveOverflow, 504  
 EventStream0TransferBlockEnd, 505  
 EventStream0TransferBlockEndFrameID, 505  
 EventStream0TransferBlockEndTimestamp, 505  
 EventStream0TransferBlockStart, 505  
 EventStream0TransferBlockStartFrameID, 505  
 EventStream0TransferBlockStartTimestamp, 505  
 EventStream0TransferBlockTrigger, 506  
 EventStream0TransferBlockTriggerFrameID, 506  
 EventStream0TransferBlockTriggerTimestamp, 506  
 EventStream0TransferBurstEnd, 506  
 EventStream0TransferBurstEndFrameID, 506  
 EventStream0TransferBurstEndTimestamp, 506  
 EventStream0TransferBurstStart, 507  
 EventStream0TransferBurstStartFrameID, 507  
 EventStream0TransferBurstStartTimestamp, 507  
 EventStream0TransferEnd, 507  
 EventStream0TransferEndFrameID, 507  
 EventStream0TransferEndTimestamp, 507  
 EventStream0TransferOverflow, 508  
 EventStream0TransferOverflowFrameID, 508  
 EventStream0TransferOverflowTimestamp, 508  
 EventStream0TransferPause, 508  
 EventStream0TransferPauseFrameID, 508  
 EventStream0TransferPauseTimestamp, 508  
 EventStream0TransferResume, 509  
 EventStream0TransferResumeFrameID, 509  
 EventStream0TransferResumeTimestamp, 509  
 EventStream0TransferStart, 509  
 EventStream0TransferStartFrameID, 509  
 EventStream0TransferStartTimestamp, 509  
 EventTest, 510  
 EventTestTimestamp, 510  
 EventTimer0End, 510  
 EventTimer0EndFrameID, 510  
 EventTimer0EndTimestamp, 510  
 EventTimer0Start, 510  
 EventTimer0StartFrameID, 511  
 EventTimer0StartTimestamp, 511  
 EventTimer1End, 511  
 EventTimer1EndFrameID, 511  
 EventTimer1EndTimestamp, 511  
 EventTimer1Start, 511  
 EventTimer1StartFrameID, 512  
 EventTimer1StartTimestamp, 512  
 ExposureActiveMode, 512  
 ExposureAuto, 512  
 ExposureMode, 512  
 ExposureTime, 512  
 ExposureTimeMode, 513  
 ExposureTimeSelector, 513  
 FactoryReset, 513  
 FileAccessBuffer, 513  
 FileAccessLength, 513  
 FileAccessOffset, 514  
 FileMode, 514  
 FileOperationExecute, 514  
 FileOperationResult, 514  
 FileOperationSelector, 514  
 FileOperationStatus, 515  
 FileSelector, 515  
 FileSize, 515  
 Gain, 515  
 GainAuto, 515  
 GainAutoBalance, 516  
 GainSelector, 516  
 Gamma, 516  
 GammaEnable, 516  
 GevActiveLinkCount, 516  
 GevCCP, 517  
 GevCurrentDefaultGateway, 517  
 GevCurrentIPAddress, 517  
 GevCurrentIPConfigurationDHCP, 517

GevCurrentIPConfigurationLLA, 517  
GevCurrentIPConfigurationPersistentIP, 517  
GevCurrentPhysicalLinkConfiguration, 518  
GevCurrentSubnetMask, 518  
GevDiscoveryAckDelay, 518  
GevFirstURL, 518  
GevGVCPExtendedStatusCodes, 518  
GevGVCPExtendedStatusCodesSelector, 518  
GevGVCPHeartbeatDisable, 519  
GevGVCPPendingAck, 519  
GevGVCPPendingTimeout, 519  
GevGVSPExtendedIDMode, 519  
GevHeartbeatTimeout, 519  
GevIEEE1588, 519  
GevIEEE1588ClockAccuracy, 520  
GevIEEE1588Mode, 520  
GevIEEE1588Status, 520  
GevInterfaceSelector, 520  
GevIPConfigurationStatus, 520  
GevMACAddress, 520  
GevMCDA, 521  
GevMCPHostPort, 521  
GevMCRC, 521  
GevMCSP, 521  
GevMCTT, 521  
GevNumberOfInterfaces, 521  
GevPAUSEFrameReception, 522  
GevPAUSEFrameTransmission, 522  
GevPersistentDefaultGateway, 522  
GevPersistentIPAddress, 522  
GevPersistentSubnetMask, 522  
GevPhysicalLinkConfiguration, 522  
GevPrimaryApplicationIPAddress, 523  
GevPrimaryApplicationSocket, 523  
GevPrimaryApplicationSwitchoverKey, 523  
GevSCCFGAllInTransmission, 523  
GevSCCFGExtendedChunkData, 523  
GevSCCFGPacketResendDestination, 523  
GevSCCFGUnconditionalStreaming, 524  
GevSCDA, 524  
GevSCPD, 524  
GevSCPDirection, 524  
GevSCPHostPort, 524  
GevSCPInterfaceIndex, 524  
GevSCPSBigEndian, 525  
GevSCPSDoNotFragment, 525  
GevSCPSFireTestPacket, 525  
GevSCPSPacketSize, 525  
GevSCSP, 525  
GevSCZoneConfigurationLock, 525  
GevSCZoneCount, 526  
GevSCZoneDirectionAll, 526  
GevSecondURL, 526  
GevStreamChannelSelector, 526  
GevSupportedOption, 526  
GevSupportedOptionSelector, 526  
GevTimestampTickFrequency, 527  
GuiXmlManifestAddress, 527  
Height, 527  
HeightMax, 527  
ImageComponentEnable, 527  
ImageComponentSelector, 528  
ImageCompressionBitrate, 528  
ImageCompressionJPEGFormatOption, 528  
ImageCompressionMode, 528  
ImageCompressionQuality, 528  
ImageCompressionRateOption, 528  
Init, 437  
IsPEnable, 529  
LineFilterWidth, 529  
LineFormat, 529  
LineInputFilterSelector, 529  
LineInverter, 529  
LineMode, 530  
LinePitch, 530  
LineSelector, 530  
LineSource, 530  
LineStatus, 530  
LineStatusAll, 530  
LinkErrorCount, 531  
LinkUptime, 531  
LogicBlockLUTInputActivation, 531  
LogicBlockLUTInputSelector, 531  
LogicBlockLUTInputSource, 531  
LogicBlockLUTOutputValue, 531  
LogicBlockLUTOutputValueAll, 532  
LogicBlockLUTRowIndex, 532  
LogicBlockLUTSelector, 532  
LogicBlockSelector, 532  
LUTEnable, 532  
LUTIndex, 532  
LUTSelector, 533  
LUTValue, 533  
LUTValueAll, 533  
MaxDeviceResetTime, 533  
OffsetX, 534  
OffsetY, 534  
PacketResendRequestCount, 534  
PayloadSize, 534  
PixelColorFilter, 534  
PixelDynamicRangeMax, 535  
PixelDynamicRangeMin, 535  
PixelFormat, 535  
PixelFormatInfoID, 535  
PixelFormatInfoSelector, 535  
PixelSize, 536  
PowerSupplyCurrent, 536  
PowerSupplyVoltage, 536  
RegionDestination, 536  
RegionMode, 536  
RegionSelector, 537  
ReverseX, 537  
ReverseY, 537  
RgbTransformLightSource, 537  
Saturation, 537  
SaturationEnable, 538

Scan3dAxisMax, 538  
Scan3dAxisMin, 538  
Scan3dCoordinateOffset, 538  
Scan3dCoordinateReferenceSelector, 538  
Scan3dCoordinateReferenceValue, 538  
Scan3dCoordinateScale, 539  
Scan3dCoordinateSelector, 539  
Scan3dCoordinateSystem, 539  
Scan3dCoordinateSystemReference, 539  
Scan3dCoordinateTransformSelector, 539  
Scan3dDistanceUnit, 539  
Scan3dInvalidDataFlag, 540  
Scan3dInvalidDataValue, 540  
Scan3dOutputMode, 540  
Scan3dTransformValue, 540  
SensorDescription, 540  
SensorDigitizationTaps, 540  
SensorHeight, 541  
SensorShutterMode, 541  
SensorTaps, 541  
SensorWidth, 541  
SequencerConfigurationMode, 541  
SequencerConfigurationValid, 541  
SequencerFeatureEnable, 542  
SequencerMode, 542  
SequencerPathSelector, 542  
SequencerSetActive, 542  
SequencerSetLoad, 543  
SequencerSetNext, 543  
SequencerSetSave, 543  
SequencerSetSelector, 543  
SequencerSetStart, 543  
SequencerSetValid, 544  
SequencerTriggerActivation, 544  
SequencerTriggerSource, 544  
SerialPortBaudRate, 544  
SerialPortDataBits, 545  
SerialPortParity, 545  
SerialPortSelector, 545  
SerialPortSource, 545  
SerialPortStopBits, 545  
SerialReceiveFramingErrorCount, 545  
SerialReceiveParityErrorCount, 546  
SerialReceiveQueueClear, 546  
SerialReceiveQueueCurrentCharacterCount, 546  
SerialReceiveQueueMaxCharacterCount, 546  
SerialTransmitQueueCurrentCharacterCount, 546  
SerialTransmitQueueMaxCharacterCount, 546  
Sharpening, 547  
SharpeningAuto, 547  
SharpeningEnable, 547  
SharpeningThreshold, 547  
SoftwareSignalPulse, 548  
SoftwareSignalSelector, 548  
SourceCount, 548  
SourceSelector, 548  
Test0001, 548  
TestEventGenerate, 549  
TestPattern, 549  
TestPatternGeneratorSelector, 549  
TestPendingAck, 549  
TimerDelay, 549  
TimerDuration, 550  
TimerReset, 550  
TimerSelector, 550  
TimerStatus, 550  
TimerTriggerActivation, 550  
TimerTriggerSource, 550  
TimerValue, 551  
Timestamp, 551  
TimestampLatch, 551  
TimestampLatchValue, 551  
TimestampReset, 551  
TLParamsLocked, 551  
TransferAbort, 552  
TransferBlockCount, 552  
TransferBurstCount, 552  
TransferComponentSelector, 552  
TransferControlMode, 552  
TransferOperationMode, 552  
TransferPause, 553  
TransferQueueCurrentBlockCount, 553  
TransferQueueMaxBlockCount, 553  
TransferQueueMode, 553  
TransferQueueOverflowCount, 553  
TransferResume, 553  
TransferSelector, 554  
TransferStart, 554  
TransferStatus, 554  
TransferStatusSelector, 554  
TransferStop, 554  
TransferStreamChannel, 554  
TransferTriggerActivation, 555  
TransferTriggerMode, 555  
TransferTriggerSelector, 555  
TransferTriggerSource, 555  
TriggerActivation, 555  
TriggerDelay, 555  
TriggerDivider, 556  
TriggerEventTest, 556  
TriggerMode, 556  
TriggerMultiplier, 556  
TriggerOverlap, 556  
TriggerSelector, 557  
TriggerSoftware, 557  
TriggerSource, 557  
UserOutputSelector, 557  
UserOutputValue, 557  
UserOutputValueAll, 558  
UserOutputValueAllMask, 558  
UserSetDefault, 558  
UserSetFeatureEnable, 558  
UserSetLoad, 558  
UserSetSave, 559  
UserSetSelector, 559  
V3\_3Enable, 559

WhiteClip, 559  
WhiteClipSelector, 560  
Width, 560  
WidthMax, 560  
Camera Base Class, 43  
Camera Base Interface Class, 86  
Camera Class, 42  
Camera List Class, 45  
CameraBase, 561  
  ~CameraBase, 563  
  BeginAcquisition, 564  
  CameraBase, 563, 564  
  DelInit, 564  
  DiscoverMaxPacketSize, 564  
  EndAcquisition, 565  
  ForceIP, 565  
  GetAccessMode, 565  
  GetBufferOwnership, 566  
  GetGuiXml, 566  
  GetNextImage, 566  
  GetNodeMap, 567  
  GetNumDataStreams, 567  
  GetNumImagesInUse, 568  
  GetTLDeviceNodeMap, 568  
  GetTlStreamNodeMap, 568  
  GetUniqueId, 569  
   GetUserBufferCount, 569  
   GetUserBufferSize, 569  
   GetUserBufferTotalSize, 570  
  Init, 570  
  InterfaceImpl, 575  
  IsInitialized, 571  
  IsStreaming, 571  
  IsValid, 571  
  operator=, 572  
  ReadPort, 572  
  RegisterEventHandler, 572, 573  
  SetBufferOwnership, 573  
  SetUserBuffers, 573, 574  
  TransportLayerDevice, 1065  
  TransportLayerStream, 1087  
  UnregisterEventHandler, 575  
  WritePort, 575  
CameraCloseFile  
  Inference.cpp, 1366  
CameraDefs Class, 44  
CameraDeleteFile  
  Inference.cpp, 1366  
cameraGrabInfoMap  
  AcquisitionMultipleCameraRecovery.cpp, 1326  
CameralInternal  
  ICameraBase, 792  
    TransportLayerDevice, 1065  
    TransportLayerStream, 1087  
  CameraList, 576  
    ~CameraList, 577  
    Append, 578  
    CameraList, 577, 578  
  Clear, 578  
  GetByDeviceID, 578  
  GetByIndex, 579  
  GetBySerial, 579  
  GetSize, 580  
  operator=, 580  
  operator[], 580  
  RemoveByDeviceID, 580  
  RemoveByIndex, 581  
  RemoveBySerial, 581  
CameraListImpl  
  ICameraList, 796  
CameraOpenFile  
  Inference.cpp, 1366  
CameraPtr, 581  
  CameraPtr, 583  
CameraPtr Class, 46  
CameraWriteToFile  
  Inference.cpp, 1366  
CastToIDestroy  
  Spinnaker::GenApi, 356  
CategoryNode, 584  
  ~CategoryNode, 585  
  CategoryNode, 585  
  GetFeatures, 585  
  SetReference, 586  
CategoryNode Class, 98  
causeSpinnakerException  
  ExceptionHandling.cpp, 1341  
causeStandardException  
  ExceptionHandling.cpp, 1341  
CBasePtr  
  Pointer Class, 159  
CBooleanPtr  
  Pointer Class, 159  
CBooleanRef  
  Spinnaker::GenApi, 344  
cbPostInsideLock  
  Spinnaker::GenApi, 348  
cbPostOutsideLock  
  Spinnaker::GenApi, 348  
CCategoryPtr  
  Pointer Class, 160  
CCategoryRef  
  Spinnaker::GenApi, 344  
CChunkAdapter, 586  
  ~CChunkAdapter, 587  
  AttachBuffer, 587  
  AttachNodeMap, 587  
  CChunkAdapter, 587  
  CheckBufferLayout, 588  
  ClearCaches, 588  
  DetachBuffer, 588  
  DetachNodeMap, 588  
  m\_pChunkAdapter, 589  
  UpdateBuffer, 588  
CChunkAdapterDcam, 589  
  ~CChunkAdapterDcam, 590

AttachBuffer, 590  
 CChunkAdapterDcam, 590  
 CheckBufferLayout, 591  
 CheckCRC, 591  
 HasCRC, 591  
 CChunkAdapterGeneric, 592  
   ~CChunkAdapterGeneric, 593  
   AttachBuffer, 593  
   CChunkAdapterGeneric, 592  
   CheckBufferLayout, 593  
 CChunkAdapterGEV, 594  
   ~CChunkAdapterGEV, 595  
   AttachBuffer, 595  
   CChunkAdapterGEV, 595  
   CheckBufferLayout, 596  
 CChunkAdapterU3V, 596  
   ~CChunkAdapterU3V, 597  
   AttachBuffer, 598  
   CChunkAdapterU3V, 597  
   CheckBufferLayout, 598  
 CChunkPort, 599  
   ~CChunkPort, 600  
   AttachChunk, 601  
   AttachPort, 601  
   CChunkPort, 600  
   CheckChunkID, 601  
   ClearCache, 601  
   DetachChunk, 602  
   DetachPort, 602  
   GetAccessMode, 602  
   GetChunkIDLength, 602  
   GetPrincipallInterfaceType, 602  
   GetSwapEndianess, 602  
   InvalidateNode, 603  
   m\_pChunkPort, 604  
   m\_pPort, 604  
   m\_pPortAdapter, 604  
   Read, 603  
   SetPortImpl, 603  
   UpdateBuffer, 603  
   Write, 603  
 CChunkPortPtr  
   Pointer Class, 160  
 CCITTFAX3  
   TIFFOption, 1062  
 CCITTFAX4  
   TIFFOption, 1062  
 CCommandPtr  
   Pointer Class, 160  
 CCommandRef  
   Spinnaker::GenApi, 345  
 CDeviceInfoPtr  
   Pointer Class, 160  
 centerXCoord  
   InferenceBoxCircle, 918  
 centerYCoord  
   InferenceBoxCircle, 918  
 CEnumEntryPtr  
   Pointer Class, 160  
 CEnumEntryRef  
   Spinnaker::GenApi, 345  
 CEnumerationPtr  
   Pointer Class, 160  
 CEnumerationRef  
   Spinnaker::GenApi, 345  
 CEnumerationTRef  
   CEnumerationTRef< EnumT >, 606  
 CEnumerationTRef< EnumT >, 604  
   ~CEnumerationTRef, 606  
   CEnumerationTRef, 606  
   GetCurrentEntry, 606  
   GetEntry, 607  
   GetValue, 607  
   operator(), 607  
   operator=, 608  
   SetEnumReference, 608  
   SetNumEnums, 608  
   SetReference, 608  
   SetValue, 609  
 CEventAdapter, 609  
   ~CEventAdapter, 610  
   AttachNodeMap, 610  
   CEventAdapter, 610  
   DeliverMessage, 610  
   DetachNodeMap, 611  
   m\_pEventAdapter, 611  
 CEventAdapter1394, 611  
   ~CEventAdapter1394, 612  
   CEventAdapter1394, 612  
   DeliverEventMessage, 613  
   DeliverMessage, 613  
 CEventAdapterGeneric, 613  
   ~CEventAdapterGeneric, 614  
   CEventAdapterGeneric, 614  
   DeliverMessage, 615  
 CEventAdapterGEV, 616  
   ~CEventAdapterGEV, 617  
   CEventAdapterGEV, 617  
   DeliverEventMessage, 617  
   DeliverMessage, 617  
 CEventAdapterU3V, 618  
   ~CEventAdapterU3V, 619  
   CEventAdapterU3V, 619  
   DeliverEventMessage, 619  
   DeliverMessage, 619  
 CEventPort, 620  
   ~CEventPort, 622  
   AttachEvent, 622  
   AttachNode, 622  
   CEventPort, 621  
   CheckEventID, 622  
   DetachEvent, 623  
   DetachNode, 623  
   GetAccessMode, 623  
   GetEventIDLength, 623  
   GetPrincipallInterfaceType, 623

GetSwapEndianess, 623  
InvalidateNode, 624  
m\_pEventPort, 624  
m\_pNode, 625  
m\_pPortAdapter, 625  
Read, 624  
SetPortImpl, 624  
Write, 624  
CFeatureBag, 625  
  ~CFeatureBag, 626  
  CFeatureBag, 626  
  GetFeatureBagHandle, 626  
  LoadFromBag, 626  
  operator==, 627  
  PersistFeature, 627  
  SetInfo, 627  
  StoreToBag, 627  
CFloatPtr, 628  
  CFloatPtr, 629  
  GetEnumAlias, 629  
  GetIntAlias, 629  
  operator=, 629  
CFloatRef  
  Spinnaker::GenApi, 345  
CGeneric\_XMLLoaderParams, 630  
  \_initialize, 630  
CGlobalLock, 631  
  ~CGlobalLock, 632  
  CGlobalLock, 631, 632  
  IsValid, 632  
  Lock, 632  
  m\_DebugCount, 633  
  TryLock, 632  
  Unlock, 633  
CGlobalLockUnlocker, 633  
  ~CGlobalLockUnlocker, 634  
  CGlobalLockUnlocker, 634  
  m\_enabled, 635  
  m\_Lock, 635  
  UnlockEarly, 634  
ChangeHeightAndGain  
  NodeMapCallback.cpp, 1376  
CheckBufferLayout  
  CChunkAdapter, 588  
  CChunkAdapterDcam, 591  
  CChunkAdapterGeneric, 593  
  CChunkAdapterGEV, 596  
  CChunkAdapterU3V, 598  
CheckChunkID  
  CChunkPort, 601  
CheckCRC  
  CChunkAdapterDcam, 591  
  IImage, 825  
  Image, 862  
CheckEventID  
  CEventPort, 622  
CheckGevEnabled  
  EnumerationEvents.cpp, 1340  
CheckNodeAccessibility  
  HighDynamicRange.cpp, 1357  
chosenChunkData  
  ChunkData.cpp, 1334  
chosenEvent  
  DeviceEvents.cpp, 1338  
chosenException  
  ExceptionHandling.cpp, 1342  
chosenFileUploadPersistence  
  Inference.cpp, 1370  
chosenInferenceNetworkType  
  Inference.cpp, 1370  
chosenRead  
  NodeMapInfo.cpp, 1380  
chosenTrigger  
  Trigger.cpp, 1394  
  Trigger\_QuickSpin.cpp, 1396  
chosenVideoType  
  SaveToAvi.cpp, 1385  
Chunk Data Inference Class, 48  
CHUNK\_BASE\_ADDRESS\_REGISTER  
  IChunkPort.h, 1215  
CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN  
  IChunkPort.h, 1215  
CHUNK\_LENGTH\_REGISTER  
  IChunkPort.h, 1215  
CHUNK\_LENGTH\_REGISTER\_LEN  
  IChunkPort.h, 1215  
ChunkAdapter Class, 99  
  ChunkAdapterDcam Class, 100  
  ChunkAdapterGeneric Class, 101  
  ChunkAdapterGEV Class, 102  
  ChunkAdapterU3V Class, 181  
ChunkBlackLevel  
  Camera, 451  
ChunkBlackLevelSelector  
  Camera, 451  
  ChunkBlackLevelSelector\_All  
    Spinnaker, 237  
  ChunkBlackLevelSelectorEnums  
    Spinnaker, 237  
  ChunkCounterSelector  
    Camera, 451  
  ChunkCounterSelector\_Counter0  
    Spinnaker, 237  
  ChunkCounterSelector\_Counter1  
    Spinnaker, 237  
  ChunkCounterSelector\_Counter2  
    Spinnaker, 237  
  ChunkCounterSelectorEnums  
    Spinnaker, 237  
  ChunkCounterValue  
    Camera, 451  
  ChunkCRC  
    Camera, 452  
  ChunkData, 635  
    ~ChunkData, 638  
    ChunkData, 637

GetBlackLevel, 638  
GetCounterValue, 638  
GetCRC, 638  
GetEncoderValue, 638  
GetExposureEndLineStatusAll, 639  
GetExposureTime, 639  
GetFrameID, 639  
GetGain, 639  
GetHeight, 640  
GetImage, 640  
GetInferenceBoundingBoxResult, 640  
GetInferenceConfidence, 640  
GetInferenceFrameId, 641  
GetInferenceResult, 641  
GetLinePitch, 641  
GetLineStatusAll, 641  
GetOffsetX, 642  
GetOffsetY, 642  
GetPartSelector, 642  
GetPixelDynamicRangeMax, 642  
GetPixelDynamicRangeMin, 643  
GetScan3dAxisMax, 643  
GetScan3dAxisMin, 643  
GetScan3dCoordinateOffset, 643  
GetScan3dCoordinateReferenceValue, 644  
GetScan3dCoordinateScale, 644  
GetScan3dInvalidDataValue, 644  
GetScan3dTransformValue, 644  
GetScanLineSelector, 645  
GetSequencerSetActive, 645  
GetSerialDataLength, 645  
GetStreamChannelID, 645  
GetTimerValue, 646  
GetTimestamp, 646  
GetTimestampLatchValue, 646  
GetTransferBlockID, 646  
GetTransferQueueCurrentBlockCount, 647  
GetWidth, 647  
SetChunks, 647  
ChunkData Class, 47  
ChunkData.cpp  
    AcquireImages, 1333  
    chosenChunkData, 1334  
    chunkDataType, 1332  
    ConfigureChunkData, 1333  
    DisableChunkData, 1333  
    DisplayChunkData, 1333  
    IMAGE, 1332  
    main, 1333  
    NODEMAP, 1332  
    PrintDeviceInfo, 1333  
    RunSingleCamera, 1334  
chunkDataType  
    ChunkData.cpp, 1332  
ChunkEnable  
    Camera, 452  
ChunkEncoderSelector  
    Camera, 452  
    ChunkEncoderSelector\_Encoder0  
        Spinnaker, 238  
    ChunkEncoderSelector\_Encoder1  
        Spinnaker, 238  
    ChunkEncoderSelector\_Encoder2  
        Spinnaker, 238  
    ChunkEncoderSelectorEnums  
        Spinnaker, 237  
    ChunkEncoderStatus  
        Camera, 452  
    ChunkEncoderStatus\_EncoderDown  
        Spinnaker, 238  
    ChunkEncoderStatus\_EncoderIdle  
        Spinnaker, 238  
    ChunkEncoderStatus\_EncoderStatic  
        Spinnaker, 238  
    ChunkEncoderStatus\_EncoderUp  
        Spinnaker, 238  
    ChunkEncoderStatusEnums  
        Spinnaker, 238  
    ChunkEncoderValue  
        Camera, 452  
    ChunkExposureEndLineStatusAll  
        Camera, 452  
    ChunkExposureTime  
        Camera, 453  
    ChunkExposureTimeSelector  
        Camera, 453  
    ChunkExposureTimeSelector\_Blue  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Common  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Cyan  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Green  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Infrared  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Magenta  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Red  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Stage1  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Stage2  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Ultraviolet  
        Spinnaker, 238  
    ChunkExposureTimeSelector\_Yellow  
        Spinnaker, 238  
    ChunkExposureTimeSelectorEnums  
        Spinnaker, 238  
    ChunkFrameID  
        Camera, 453  
    ChunkGain  
        Camera, 453  
    ChunkGainSelector  
        Camera, 453

ChunkGainSelector\_All  
    Spinnaker, 239

ChunkGainSelector\_Blue  
    Spinnaker, 239

ChunkGainSelector\_Green  
    Spinnaker, 239

ChunkGainSelector\_Red  
    Spinnaker, 239

ChunkGainSelectorEnums  
    Spinnaker, 238

ChunkHeight  
    Camera, 453

ChunkID  
    DCAM\_CHUNK\_TRAILER, 702  
    GVCP\_CHUNK\_TRAILER, 772  
    SingleChunkData\_t, 1029  
    SingleChunkDataStr\_t, 1030  
    U3V\_CHUNK\_TRAILER, 1099

ChunkImage  
    Camera, 454

ChunkImageComponent  
    Camera, 454

ChunkImageComponent\_Color  
    Spinnaker, 239

ChunkImageComponent\_Confidence  
    Spinnaker, 239

ChunkImageComponent\_Disparity  
    Spinnaker, 239

ChunkImageComponent\_Infrared  
    Spinnaker, 239

ChunkImageComponent\_Intensity  
    Spinnaker, 239

ChunkImageComponent\_Range  
    Spinnaker, 239

ChunkImageComponent\_Scatter  
    Spinnaker, 239

ChunkImageComponent\_Ultraviolet  
    Spinnaker, 239

ChunkImageComponentEnums  
    Spinnaker, 239

ChunkInferenceBoundingBoxResult  
    Camera, 454

ChunkInferenceConfidence  
    Camera, 454

ChunkInferenceFrameId  
    Camera, 454

ChunkInferenceResult  
    Camera, 454

ChunkLength  
    DCAM\_CHUNK\_TRAILER, 702  
    GVCP\_CHUNK\_TRAILER, 772  
    SingleChunkData\_t, 1029  
    SingleChunkDataStr\_t, 1030  
    U3V\_CHUNK\_TRAILER, 1099

ChunkLinePitch  
    Camera, 455

ChunkLineStatusAll  
    Camera, 455

ChunkModeActive  
    Camera, 455

ChunkOffset  
    SingleChunkData\_t, 1029  
    SingleChunkDataStr\_t, 1030

ChunkOffsetX  
    Camera, 455

ChunkOffsetY  
    Camera, 455

ChunkPartSelector  
    Camera, 455

ChunkPixelDynamicRangeMax  
    Camera, 456

ChunkPixelDynamicRangeMin  
    Camera, 456

ChunkPixelFormat  
    Camera, 456

ChunkPixelFormat\_BayerBG8  
    Spinnaker, 240

ChunkPixelFormat\_BayerGB8  
    Spinnaker, 240

ChunkPixelFormat\_BayerGR8  
    Spinnaker, 240

ChunkPixelFormat\_BayerRG8  
    Spinnaker, 240

ChunkPixelFormat\_Mono12Packed  
    Spinnaker, 239

ChunkPixelFormat\_Mono16  
    Spinnaker, 239

ChunkPixelFormat\_Mono8  
    Spinnaker, 239

ChunkPixelFormat\_RGB8Packed  
    Spinnaker, 239

ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY  
    Spinnaker, 240

ChunkPixelFormat\_YUV422Packed  
    Spinnaker, 240

ChunkPixelFormatEnums  
    Spinnaker, 239

ChunkPort Class, 103

ChunkRegionID  
    Camera, 456

ChunkRegionID\_Region0  
    Spinnaker, 240

ChunkRegionID\_Region1  
    Spinnaker, 240

ChunkRegionID\_Region2  
    Spinnaker, 240

ChunkRegionIDEnums  
    Spinnaker, 240

ChunkScan3dAxisMax  
    Camera, 456

ChunkScan3dAxisMin  
    Camera, 456

ChunkScan3dCoordinateOffset  
    Camera, 457

ChunkScan3dCoordinateReferenceSelector  
    Camera, 457

ChunkScan3dCoordinateReferenceSelector\_RotationX  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelector\_RotationY  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelector\_RotationZ  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelector\_TranslationX  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelector\_TranslationY  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelector\_TranslationZ  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceSelectorEnums  
    Spinnaker, 240

ChunkScan3dCoordinateReferenceValue  
    Camera, 457

ChunkScan3dCoordinateScale  
    Camera, 457

ChunkScan3dCoordinateSelector  
    Camera, 457

ChunkScan3dCoordinateSelector\_CoordinateA  
    Spinnaker, 241

ChunkScan3dCoordinateSelector\_CoordinateB  
    Spinnaker, 241

ChunkScan3dCoordinateSelector\_CoordinateC  
    Spinnaker, 241

ChunkScan3dCoordinateSelectorEnums  
    Spinnaker, 240

ChunkScan3dCoordinateSystem  
    Camera, 457

ChunkScan3dCoordinateSystem\_Cartesian  
    Spinnaker, 241

ChunkScan3dCoordinateSystem\_Cylindrical  
    Spinnaker, 241

ChunkScan3dCoordinateSystem\_Spherical  
    Spinnaker, 241

ChunkScan3dCoordinateSystemEnums  
    Spinnaker, 241

ChunkScan3dCoordinateSystemReference  
    Camera, 458

ChunkScan3dCoordinateSystemReference\_Anchor  
    Spinnaker, 241

ChunkScan3dCoordinateSystemReference\_Transformed  
    Spinnaker, 241

ChunkScan3dCoordinateSystemReferenceEnums  
    Spinnaker, 241

ChunkScan3dCoordinateTransformSelector  
    Camera, 458

ChunkScan3dCoordinateTransformSelector\_RotationX  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelector\_RotationY  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelector\_RotationZ  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelector\_TranslationX  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelector\_TranslationY  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelector\_TranslationZ  
    Spinnaker, 242

ChunkScan3dCoordinateTransformSelectorEnums  
    Spinnaker, 241

ChunkScan3dDistanceUnit  
    Camera, 458

ChunkScan3dDistanceUnit\_Inch  
    Spinnaker, 242

ChunkScan3dDistanceUnit\_Millimeter  
    Spinnaker, 242

ChunkScan3dDistanceUnitEnums  
    Spinnaker, 242

ChunkScan3dInvalidDataFlag  
    Camera, 458

ChunkScan3dInvalidHeaderValue  
    Camera, 458

ChunkScan3dOutputMode  
    Camera, 458

ChunkScan3dOutputMode\_CalibratedABC\_Grid  
    Spinnaker, 242

ChunkScan3dOutputMode\_CalibratedABC\_PointCloud  
    Spinnaker, 242

ChunkScan3dOutputMode\_CalibratedAC  
    Spinnaker, 243

ChunkScan3dOutputMode\_CalibratedAC\_Linescan  
    Spinnaker, 243

ChunkScan3dOutputMode\_CalibratedC  
    Spinnaker, 243

ChunkScan3dOutputMode\_CalibratedC\_Linescan  
    Spinnaker, 243

ChunkScan3dOutputMode\_DisparityC  
    Spinnaker, 243

ChunkScan3dOutputMode\_DisparityC\_Linescan  
    Spinnaker, 243

ChunkScan3dOutputMode\_RectifiedC  
    Spinnaker, 243

ChunkScan3dOutputMode\_RectifiedC\_Linescan  
    Spinnaker, 243

ChunkScan3dOutputMode\_UncalibratedC  
    Spinnaker, 242

ChunkScan3dOutputModeEnums  
    Spinnaker, 242

ChunkScan3dTransformValue  
    Camera, 459

ChunkScanLineSelector  
    Camera, 459

ChunkSelector  
    Camera, 459

ChunkSelector\_BlackLevel  
    Spinnaker, 244

ChunkSelector\_CRC  
    Spinnaker, 243

ChunkSelector\_ExposureEndLineStatusAll  
    Spinnaker, 244

ChunkSelector\_ExposureTime  
    Spinnaker, 243

ChunkSelector\_FrameID  
    Spinnaker, 243

ChunkSelector\_Gain  
    Spinnaker, 244

ChunkSelector\_Height  
    Spinnaker, 243

ChunkSelector\_Image  
    Spinnaker, 243

ChunkSelector\_OffsetX  
    Spinnaker, 243

ChunkSelector\_OffsetY  
    Spinnaker, 243

ChunkSelector\_PixelFormat  
    Spinnaker, 244

ChunkSelector\_SequencerSetActive  
    Spinnaker, 244

ChunkSelector\_SerialData  
    Spinnaker, 244

ChunkSelector\_Timestamp  
    Spinnaker, 244

ChunkSelector\_Width  
    Spinnaker, 243

ChunkSelectorEnums  
    Spinnaker, 243

ChunkSequencerSetActive  
    Camera, 459

ChunkSerialData  
    Camera, 459

ChunkSerialDataLength  
    Camera, 459

ChunkSerialReceiveOverflow  
    Camera, 460

ChunkSourceID  
    Camera, 460

ChunkSourceID\_Source0  
    Spinnaker, 244

ChunkSourceID\_Source1  
    Spinnaker, 244

ChunkSourceID\_Source2  
    Spinnaker, 244

ChunkSourceIDEnums  
    Spinnaker, 244

ChunkStreamChannelID  
    Camera, 460

ChunkTimerSelector  
    Camera, 460

ChunkTimerSelector\_Timer0  
    Spinnaker, 244

ChunkTimerSelector\_Timer1  
    Spinnaker, 244

ChunkTimerSelector\_Timer2  
    Spinnaker, 244

ChunkTimerSelectorEnums  
    Spinnaker, 244

ChunkTimerValue  
    Camera, 460

ChunkTimestamp  
    Camera, 460

ChunkTimestampLatchValue  
    Camera, 461

ChunkTransferBlockID  
    Camera, 461

ChunkTransferQueueCurrentBlockCount  
    Camera, 461

ChunkTransferStreamID  
    Camera, 461

ChunkTransferStreamID\_Stream0  
    Spinnaker, 245

ChunkTransferStreamID\_Stream1  
    Spinnaker, 245

ChunkTransferStreamID\_Stream2  
    Spinnaker, 245

ChunkTransferStreamID\_Stream3  
    Spinnaker, 245

ChunkTransferStreamIDEnums  
    Spinnaker, 244

ChunkWidth  
    Camera, 461

CIntegerPtr  
    Pointer Class, 161

CIntegerRef  
    Spinnaker::GenApi, 345

circle  
    InferenceBoundingBox, 914

CL  
    Spinnaker::GenApi, 352

classId  
    InferenceBoundingBox, 914

CLASSIFICATION  
    Inference.cpp, 1366

CIConfiguration  
    Camera, 461

CIConfiguration\_Base  
    Spinnaker, 245

CIConfiguration\_DualBase  
    Spinnaker, 245

CIConfiguration\_EightyBit  
    Spinnaker, 245

CIConfiguration\_Full  
    Spinnaker, 245

CIConfiguration\_Medium  
    Spinnaker, 245

CIConfigurationEnums  
    Spinnaker, 245

CleanUp  
    SerialRxTx.cpp, 1390

CleanupChunkAdapter  
    IDataStream, 807

Clear  
    CameraList, 578  
    ICameraList, 794  
    IInterfaceList, 852  
    InterfaceList, 945

ClearCache  
    CChunkPort, 601  
    CNodeMapFactory, 661

ClearCaches  
    CChunkAdapter, 588

**ClearXMLCache**  
 NodeMap, 996  
**CLock**, 648, 649  
 ~CLock, 648, 650  
 CLock, 648, 650  
 Lock, 649, 651  
 m\_bOwnLock, 651  
 m\_lock, 652  
 NodeMap, 651  
 TryLock, 649, 651  
 Unlock, 649, 651  
**CLockEx**, 652, 653  
 m\_lockEx, 654  
**Close**  
 SpinVideo, 1033  
**close**  
 IDevFileStreamBase< CharType, Traits >, 813  
 IDevFileStreamBuf< CharType, Traits >, 816  
 ODevFileStreamBase< CharType, Traits >, 1004  
 ODevFileStreamBuf< CharType, Traits >, 1006  
**CloseFile**  
 FileAccess\_QuickSpin.cpp, 1345  
**closeFile**  
 FileProtocolAdapter, 748  
**CITimeSlotsCount**  
 Camera, 462  
**CITimeSlotsCount\_One**  
 Spinnaker, 245  
**CITimeSlotsCount\_Three**  
 Spinnaker, 245  
**CITimeSlotsCount\_Two**  
 Spinnaker, 245  
**CITimeSlotsCountEnums**  
 Spinnaker, 245  
**CNodeCallback**, 654  
 ~CNodeCallback, 655  
 CNodeCallback, 655  
 Destroy, 655  
 GetCallbackType, 656  
 GetNode, 656  
 m\_CallbackType, 656  
 m\_pNode, 656  
 operator(), 656  
**CNodeMapDynPtr**  
 Pointer Class, 161  
**CNodeMapFactory**, 657  
 ~CNodeMapFactory, 659  
 AddInjectionData, 661  
 ApplyStyleSheet, 661  
 ClearCache, 661  
 CNodeMapFactory, 659, 660  
 CreateEmptyNodeMap, 662  
 CreateNodeDataFromNodeMap, 662  
 CreateNodeMap, 662  
 ExtractSubtree, 662  
 GetNodeStatistics, 663  
 GetSupportedSchemaVersions, 663  
 IsCameraDescriptionFileDataReleased, 663  
 IsEmpty, 663  
 IsLoaded, 663  
 IsPreprocessed, 664  
 LoadAndInject, 664  
 operator=, 664  
 Preprocess, 664  
 ReleaseCameraDescriptionFileData, 664  
 ToString, 665  
 ToXml, 665  
**CNodeMapFactory::NodeStatistics\_t**, 1002  
 NumLinks, 1002  
 NumNodes, 1002  
 NumProperties, 1002  
 NumStrings, 1002  
**CNodeMapPtr**  
 Pointer Class, 161  
**CNodeMapRef**, 665  
 CNodeMapRef, 666, 667  
 operator=, 667  
 Spinnaker::GenApi, 345  
**CNodeMapRefT**  
 CNodeMapRefT< TCameraParams >, 670  
**CNodeMapRefT< TCameraParams >**, 668  
 \_ClearXMLCache, 670  
 \_Connect, 671  
 \_Destroy, 671  
 \_GetDeviceName, 671  
 \_GetNode, 671  
 \_GetNodes, 671  
 \_GetSupportedSchemaVersions, 672  
 \_InvalidateNodes, 672  
 \_LoadXMLFromFile, 672  
 \_LoadXMLFromFileInject, 672  
 \_LoadXMLFromString, 672  
 \_LoadXMLFromStringInject, 673  
 \_LoadXMLFromZIPData, 673  
 \_LoadXMLFromZIPFile, 673  
 \_Poll, 673  
 \_Ptr, 674  
 ~CNodeMapRefT, 670  
 CNodeMapRefT, 670  
 operator=, 673, 674  
**CNodePtr**  
 Pointer Class, 161  
**CNodeRef**  
 Spinnaker::GenApi, 345  
**ColorProcessingAlgorithm**  
 Spinnaker, 245  
**ColorTransformationEnable**  
 Camera, 462  
**ColorTransformationSelector**  
 Camera, 462  
**ColorTransformationSelector\_RGBtoRGB**  
 Spinnaker, 246  
**ColorTransformationSelector\_RGBtoYUV**  
 Spinnaker, 246  
**ColorTransformationSelectorEnums**  
 Spinnaker, 246

ColorTransformationValue  
    Camera, 462

ColorTransformationValueSelector  
    Camera, 462

ColorTransformationValueSelector\_Gain00  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain01  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain02  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain10  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain11  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain12  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain20  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain21  
    Spinnaker, 247

ColorTransformationValueSelector\_Gain22  
    Spinnaker, 247

ColorTransformationValueSelector\_Offset0  
    Spinnaker, 247

ColorTransformationValueSelector\_Offset1  
    Spinnaker, 247

ColorTransformationValueSelector\_Offset2  
    Spinnaker, 247

ColorTransformationValueSelectorEnums  
    Spinnaker, 246

COM\_PORT\_COUNT\_MAX  
    SerialRxTx.cpp, 1388

Combine  
    INode Interface, 136, 137

Command  
    GVCP\_REQUEST\_HEADER, 781

COMMAND\_MAGIC  
    Spinnaker::GenApi, 379

CommandHeader  
    U3V\_EVENT\_MESSAGE, 1102

CommandId  
    U3V\_COMMAND\_HEADER, 1100

CommandNode, 674

- ~CommandNode, 676
- CommandNode, 675, 676
- Execute, 676
- IsDone, 676
- operator(), 677
- SetReference, 677

CommandNode Class, 104

compare  
    gcstring, 765

Compatibility.h  
    FMT\_I64, 1188

compression  
    TIFFOption, 1062

compressionLevel  
    PNGOption, 1009

CompressionMethod  
    TIFFOption, 1061

CompressionRatio  
    Camera, 463

confidence  
    InferenceBoundingBox, 914

ConfigureActionControl  
    ActionCommand.cpp, 1328

ConfigureAdapter  
    AdapterConfig, 184

ConfigureCallbacks  
    NodeMapCallback.cpp, 1376

ConfigureCamera  
    AcquisitionMultipleCameraRecovery.cpp, 1324

ConfigureChunkData  
    ActionCommand.cpp, 1328

- ChunkData.cpp, 1333
- Inference.cpp, 1367

ConfigureCustomImageSettings  
    ImageFormatControl.cpp, 1362

- ImageFormatControl\_QuickSpin.cpp, 1363

ConfigureDevice  
    SerialRxTx.cpp, 1390

ConfigureDeviceEvents  
    DeviceEvents.cpp, 1337

ConfigureDigitalIO  
    CounterAndTimer.cpp, 1335

ConfigureExposure  
    Exposure.cpp, 1342

- Exposure\_QuickSpin.cpp, 1344

ConfigureExposureandTrigger  
    CounterAndTimer.cpp, 1335

ConfigureEEE1588  
    ActionCommand.cpp, 1328

ConfigureImageEvents  
    ImageEvents.cpp, 1360

ConfigureInference  
    Inference.cpp, 1367

ConfigureInterface  
    ActionCommand.cpp, 1328

ConfigureLogicBlock  
    LogicBlock.cpp, 1372

ConfigureLookupTables  
    LookupTable.cpp, 1374

ConfigureOtherNodes  
    ActionCommand.cpp, 1328

ConfigureSequencerPartOne  
    Sequencer.cpp, 1386

ConfigureSequencerPartTwo  
    Sequencer.cpp, 1386

ConfigureStream  
    Polarization.cpp, 1381

ConfigureTestPattern  
    Inference.cpp, 1367

ConfigureTrigger  
    ActionCommand.cpp, 1328

- BufferHandling.cpp, 1330
- Inference.cpp, 1367

LogicBlock.cpp, 1372  
Trigger.cpp, 1393  
Trigger\_QuickSpin.cpp, 1395  
ConfigureUserSet1  
    AcquisitionMultipleCameraRecovery.cpp, 1324  
Connect  
    NodeMap, 996  
    Spinnaker::GenApi, 357  
Container Class, 105  
ContentType\_Xml  
    Spinnaker::GenApi, 354  
ContentType\_ZippedXml  
    Spinnaker::GenApi, 354  
Conversion, 187  
    NumToCString, 187  
Convert  
    IImage, 825, 826  
    Image, 862, 863  
Counter, 677  
    Counter, 678  
    GetValue, 678  
    IsZero, 678  
    operator unsigned int, 678  
    operator++, 678  
    operator--, 678, 679  
Counter Class, 106  
CounterAndTimer.cpp  
    AcquireImages, 1334  
    ConfigureDigitalIO, 1335  
    ConfigureExposureandTrigger, 1335  
    main, 1335  
    PrintDeviceInfo, 1335  
    ResetTrigger, 1335  
    RunSingleCamera, 1335  
    SetupCounterAndTimer, 1336  
CounterDelay  
    Camera, 463  
CounterDuration  
    Camera, 463  
CounterEventActivation  
    Camera, 463  
CounterEventActivation\_AnyEdge  
    Spinnaker, 247  
CounterEventActivation\_FallingEdge  
    Spinnaker, 247  
CounterEventActivation\_LevelHigh  
    Spinnaker, 247  
CounterEventActivation\_LevelLow  
    Spinnaker, 247  
CounterEventActivation\_RisingEdge  
    Spinnaker, 247  
CounterEventActivationEnums  
    Spinnaker, 247  
CounterEventSource  
    Camera, 463  
CounterEventSource\_Counter0End  
    Spinnaker, 248  
CounterEventSource\_Counter0Start  
    Spinnaker, 248  
CounterEventSource\_Counter1End  
    Spinnaker, 248  
CounterEventSource\_Counter1Start  
    Spinnaker, 248  
CounterEventSource\_ExposureEnd  
    Spinnaker, 248  
CounterEventSource\_ExposureStart  
    Spinnaker, 248  
CounterEventSource\_FrameTriggerWait  
    Spinnaker, 248  
CounterEventSource\_Line0  
    Spinnaker, 247  
CounterEventSource\_Line1  
    Spinnaker, 247  
CounterEventSource\_Line2  
    Spinnaker, 247  
CounterEventSource\_Line3  
    Spinnaker, 247  
CounterEventSource\_LogicBlock0  
    Spinnaker, 248  
CounterEventSource\_LogicBlock1  
    Spinnaker, 248  
CounterEventSource\_MHzTick  
    Spinnaker, 247  
CounterEventSource\_Off  
    Spinnaker, 247  
CounterEventSource\_UserOutput0  
    Spinnaker, 248  
CounterEventSource\_UserOutput1  
    Spinnaker, 248  
CounterEventSource\_UserOutput2  
    Spinnaker, 248  
CounterEventSource\_UserOutput3  
    Spinnaker, 248  
CounterEventSourceEnums  
    Spinnaker, 247  
CounterReset  
    Camera, 464  
CounterResetActivation  
    Camera, 464  
CounterResetActivation\_AnyEdge  
    Spinnaker, 248  
CounterResetActivation\_FallingEdge  
    Spinnaker, 248  
CounterResetActivation\_LevelHigh  
    Spinnaker, 248  
CounterResetActivation\_LevelLow  
    Spinnaker, 248  
CounterResetActivation\_RisingEdge  
    Spinnaker, 248  
CounterResetActivationEnums  
    Spinnaker, 248  
CounterResetSource  
    Camera, 464  
CounterResetSource\_Counter0End  
    Spinnaker, 249  
CounterResetSource\_Counter0Start

Spinnaker, 249  
CounterResetSource\_Counter1End  
    Spinnaker, 249  
CounterResetSource\_Counter1Start  
    Spinnaker, 249  
CounterResetSource\_ExposureEnd  
    Spinnaker, 249  
CounterResetSource\_ExposureStart  
    Spinnaker, 249  
CounterResetSource\_FrameTriggerWait  
    Spinnaker, 249  
CounterResetSource\_Line0  
    Spinnaker, 248  
CounterResetSource\_Line1  
    Spinnaker, 248  
CounterResetSource\_Line2  
    Spinnaker, 248  
CounterResetSource\_Line3  
    Spinnaker, 249  
CounterResetSource\_LogicBlock0  
    Spinnaker, 249  
CounterResetSource\_LogicBlock1  
    Spinnaker, 249  
CounterResetSource\_Off  
    Spinnaker, 248  
CounterResetSource\_UserOutput0  
    Spinnaker, 249  
CounterResetSource\_UserOutput1  
    Spinnaker, 249  
CounterResetSource\_UserOutput2  
    Spinnaker, 249  
CounterResetSource\_UserOutput3  
    Spinnaker, 249  
CounterResetSourceEnums  
    Spinnaker, 248  
CounterSelector  
    Camera, 464  
CounterSelector\_Counter0  
    Spinnaker, 249  
CounterSelector\_Counter1  
    Spinnaker, 249  
CounterSelectorEnums  
    Spinnaker, 249  
CounterStart  
    PerformanceCounter, 189  
CounterStatus  
    Camera, 464  
CounterStatus\_CounterActive  
    Spinnaker, 249  
CounterStatus\_CounterCompleted  
    Spinnaker, 249  
CounterStatus\_CounterIdle  
    Spinnaker, 249  
CounterStatus\_CounterOverflow  
    Spinnaker, 249  
CounterStatus\_CounterTriggerWait  
    Spinnaker, 249  
CounterStatusEnums  
    Spinnaker, 249  
Spinnaker, 249  
CounterTriggerActivation  
    Camera, 464  
CounterTriggerActivation\_AnyEdge  
    Spinnaker, 250  
CounterTriggerActivation\_FallingEdge  
    Spinnaker, 250  
CounterTriggerActivation\_LevelHigh  
    Spinnaker, 250  
CounterTriggerActivation\_LevelLow  
    Spinnaker, 250  
CounterTriggerActivation\_RisingEdge  
    Spinnaker, 250  
CounterTriggerActivationEnums  
    Spinnaker, 250  
CounterTriggerSource  
    Camera, 465  
CounterTriggerSource\_Counter0End  
    Spinnaker, 250  
CounterTriggerSource\_Counter0Start  
    Spinnaker, 250  
CounterTriggerSource\_Counter1End  
    Spinnaker, 250  
CounterTriggerSource\_Counter1Start  
    Spinnaker, 250  
CounterTriggerSource\_ExposureEnd  
    Spinnaker, 250  
CounterTriggerSource\_ExposureStart  
    Spinnaker, 250  
CounterTriggerSource\_FrameTriggerWait  
    Spinnaker, 250  
CounterTriggerSource\_Line0  
    Spinnaker, 250  
CounterTriggerSource\_Line1  
    Spinnaker, 250  
CounterTriggerSource\_Line2  
    Spinnaker, 250  
CounterTriggerSource\_Line3  
    Spinnaker, 250  
CounterTriggerSource\_LogicBlock0  
    Spinnaker, 250  
CounterTriggerSource\_LogicBlock1  
    Spinnaker, 250  
CounterTriggerSource\_Off  
    Spinnaker, 250  
CounterTriggerSource\_UserOutput0  
    Spinnaker, 250  
CounterTriggerSource\_UserOutput1  
    Spinnaker, 250  
CounterTriggerSource\_UserOutput2  
    Spinnaker, 250  
CounterTriggerSource\_UserOutput3  
    Spinnaker, 250  
CounterTriggerSourceEnums  
    Spinnaker, 250  
CounterValue  
    Camera, 465  
CounterValueAtReset

Camera, 465  
**CPointer**  
 CPointer< T, B >, 680  
**CPointer< T, B >**, 679  
 ~CPointer, 680  
 CPointer, 680  
 IsValid, 681  
 m\_pT, 683  
 operator bool, 681  
 operator T\*, 681  
 operator!=, 681, 682  
 operator\*, 682  
 operator(), 682  
 operator->, 682  
 operator=, 682  
 operator==, 683  
**CPortConstructPtr**  
 Pointer Class, 161  
**CPortImpl**, 684  
 ~CPortImpl, 685  
 CPortImpl, 685  
 GetAccessMode, 685  
 GetSwapEndianess, 685  
 InvalidateNode, 686  
 m\_ptrPort, 687  
 Read, 686  
 Replay, 686  
 SetPortImpl, 686  
 Write, 686  
**CPortPtr**  
 Pointer Class, 161  
**CPortRecorderPtr**  
 Pointer Class, 162  
**CPortRecorderRef**  
 Spinnaker::GenApi, 345  
**CPortRef**  
 Spinnaker::GenApi, 346  
**CPortReplayPtr**  
 Pointer Class, 162  
**CPortWriteList**, 687  
 ~CPortWriteList, 688  
 CPortWriteList, 688  
 GetCookie, 689  
 GetPortWriteListHandle, 689  
 m\_pWriteList, 690  
 Replay, 689  
 SetCookie, 689  
 Write, 689  
**CPortWriteListPtr**  
 Pointer Class, 162  
**CpuUsageInfo**, 690  
 dummy, 690  
**cpuUsageInfo**  
 GigEVisionPerformance.cpp, 1355  
**CpuUtil**, 188  
 GetCpuStats, 188  
 StartCpuTracing, 188  
 StopCpuTracing, 188  
**CRCChecksum**  
 DCAM\_CHECKSUM, 701  
**Create**  
 Image, 863, 864  
**CreateAndSaveAolpDolpImages**  
 Polarization.cpp, 1382  
**CreateAndSaveGlareReducedImage**  
 Polarization.cpp, 1382  
**CreateAndSaveStokesImages**  
 Polarization.cpp, 1382  
**CreateAolp**  
 ImageUtilityPolarization, 907, 908  
**CreateDolp**  
 ImageUtilityPolarization, 908, 909  
**CreateEmptyNodeMap**  
 CNodeMapFactory, 662  
**CreateGlareReduced**  
 ImageUtilityPolarization, 909, 910  
**CreateHeatmap**  
 ImageUtilityHeatmap, 903, 904  
**CreateHeatmapImages**  
 Polarization.cpp, 1382  
**CreateNodeDataFromNodeMap**  
 CNodeMapFactory, 662  
**CreateNodeMap**  
 CNodeMapFactory, 662  
**CreateNormalized**  
 ImageUtility, 899–901  
**CreateNormalizedImage**  
 Polarization.cpp, 1382  
**CreateScaled**  
 ImageUtility, 901  
**CreateShared**  
 Image, 864  
**CreateStokesS0**  
 ImageUtilityPolarization, 910  
**CreateStokesS1**  
 ImageUtilityPolarization, 911  
**CreateStokesS2**  
 ImageUtilityPolarization, 912  
**CRegisterPortImpl**, 691  
 ~CRegisterPortImpl, 692  
 CRegisterPortImpl, 692  
 GetAccessMode, 692  
 Read, 693  
 ReadRegister, 693  
 SetPortImpl, 693  
 Write, 693  
 WriteRegister, 694  
**CRegisterPtr**  
 Pointer Class, 162  
**CRegisterRef**  
 Spinnaker::GenApi, 346  
**CSelectorPtr**  
 Pointer Class, 162  
**CSelectorRef**  
 Spinnaker::GenApi, 346  
**CSelectorSet**, 694

~CSelectorSet, 696  
CSelectorSet, 695  
GetSelectorList, 696  
IsEmpty, 696  
Restore, 696  
SetFirst, 696  
SetNext, 696  
ToString, 697  
CStringPtr  
    Pointer Class, 162  
CStringRef  
    Spinnaker::GenApi, 346  
ctDependingNodes  
    Spinnaker::GenApi, 350  
CTestPortStruct  
    CTestPortStruct< CDataStruct >, 699  
CTestPortStruct< CDataStruct >, 697  
    CTestPortStruct, 699  
    GetAccessType, 699  
    GetNumReads, 699  
    GetNumWrites, 699  
    GetPrincipalInterfaceType, 699  
    m\_BaseAddress, 700  
    m\_NumReads, 700  
    m\_NumWrites, 701  
    MemSet, 699  
    Read, 700  
    ResetStatistics, 700  
    Write, 700  
ctlInvalidatingChildren  
    Spinnaker::GenApi, 350  
ctParentNodes  
    Spinnaker::GenApi, 350  
ctReadingChildren  
    Spinnaker::GenApi, 350  
ctTerminalNodes  
    Spinnaker::GenApi, 350  
ctWritingChildren  
    Spinnaker::GenApi, 350  
Custom  
    Spinnaker::GenApi, 351  
CValuePtr  
    Pointer Class, 163  
CValueRef  
    Spinnaker::GenApi, 346  
CxpConnectionSelector  
    Camera, 465  
CxpConnectionTestErrorCount  
    Camera, 465  
CxpConnectionTestMode  
    Camera, 465  
CxpConnectionTestMode\_Mode1  
    Spinnaker, 251  
CxpConnectionTestMode\_Off  
    Spinnaker, 251  
CxpConnectionTestModeEnums  
    Spinnaker, 251  
CxpConnectionTestPacketCount  
    Camera, 466  
CxpLinkConfiguration  
    Camera, 466  
CxpLinkConfiguration\_Auto  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP1\_X1  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP1\_X2  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP1\_X3  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP1\_X4  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP1\_X5  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP1\_X6  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP2\_X1  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP2\_X2  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP2\_X3  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP2\_X4  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP2\_X5  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP2\_X6  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP3\_X1  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP3\_X2  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP3\_X3  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP3\_X4  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP3\_X5  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP3\_X6  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP5\_X1  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP5\_X2  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP5\_X3  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP5\_X4  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP5\_X5  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP5\_X6  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP6\_X1  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP6\_X2  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP6\_X3

Spinnaker, 251  
CxpLinkConfiguration\_CXP6\_X4  
    Spinnaker, 251  
CxpLinkConfiguration\_CXP6\_X5  
    Spinnaker, 252  
CxpLinkConfiguration\_CXP6\_X6  
    Spinnaker, 252  
CxpLinkConfigurationEnums  
    Spinnaker, 251  
CxpLinkConfigurationPreferred  
    Camera, 466  
CxpLinkConfigurationPreferred\_CXP1\_X1  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP1\_X2  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP1\_X3  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP1\_X4  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP1\_X5  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP1\_X6  
    Spinnaker, 253  
CxpLinkConfigurationPreferred\_CXP2\_X1  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP2\_X2  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP2\_X3  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP2\_X4  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP2\_X5  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP2\_X6  
    Spinnaker, 253  
CxpLinkConfigurationPreferred\_CXP3\_X1  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP3\_X2  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP3\_X3  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP3\_X4  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP3\_X5  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP3\_X6  
    Spinnaker, 253  
CxpLinkConfigurationPreferred\_CXP5\_X1  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP5\_X2  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP5\_X3  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP5\_X4  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP5\_X5  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP5\_X6  
    Spinnaker, 252  
Spinnaker, 253  
CxpLinkConfigurationPreferred\_CXP6\_X1  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP6\_X2  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP6\_X3  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP6\_X4  
    Spinnaker, 252  
CxpLinkConfigurationPreferred\_CXP6\_X5  
    Spinnaker, 253  
CxpLinkConfigurationPreferred\_CXP6\_X6  
    Spinnaker, 253  
CxpLinkConfigurationPreferredEnums  
    Spinnaker, 252  
CxpLinkConfigurationStatus  
    Camera, 466  
CxpLinkConfigurationStatus\_CXP1\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP1\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP1\_X3  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP1\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP1\_X5  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP1\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP2\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP2\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP2\_X3  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP2\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP2\_X5  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP2\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP3\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP3\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP3\_X3  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP3\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP3\_X5  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP3\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP5\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X3  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X5  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP5\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X3  
    Spinnaker, 253

Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X5  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP5\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP6\_X1  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP6\_X2  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP6\_X3  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP6\_X4  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_CXP6\_X5  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_CXP6\_X6  
    Spinnaker, 254  
CxpLinkConfigurationStatus\_None  
    Spinnaker, 253  
CxpLinkConfigurationStatus\_Pending  
    Spinnaker, 253  
CxpLinkConfigurationStatusEnums  
    Spinnaker, 253  
CxpPoCxpAuto  
    Camera, 466  
CxpPoCxpStatus  
    Camera, 466  
CxpPoCxpStatus\_Auto  
    Spinnaker, 254  
CxpPoCxpStatus\_Off  
    Spinnaker, 254  
CxpPoCxpStatus\_Tripped  
    Spinnaker, 254  
CxpPoCxpStatusEnums  
    Spinnaker, 254  
CxpPoCxpTripReset  
    Camera, 467  
CxpPoCxpTurnOff  
    Camera, 467

Data  
    GVCP\_EVENTDATA\_REQUEST, 779  
    GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID,  
        780

DATA\_BITS  
    SerialRxTx.cpp, 1389

DCAM\_CHECKSUM, 701  
    CRCChecksum, 701

DCAM\_CHUNK\_TRAILER, 701  
    ChunkID, 702  
    ChunkLength, 702  
    InverseChunkLength, 702

DDR  
    Inference.cpp, 1365

DecimationHorizontal  
    Camera, 467

DecimationHorizontalMode  
    Camera, 467

DecimationHorizontalMode\_Discard  
    Spinnaker, 254

DecimationHorizontalModeEnums  
    Spinnaker, 254

DecimationSelector  
    Camera, 467

DecimationSelector\_All  
    Spinnaker, 255

DecimationSelector\_Sensor  
    Spinnaker, 255

DecimationSelectorEnums  
    Spinnaker, 254

DecimationVertical  
    Camera, 468

DecimationVerticalMode  
    Camera, 468

DecimationVerticalMode\_Discard  
    Spinnaker, 255

DecimationVerticalModeEnums  
    Spinnaker, 255

Decreasing  
    Spinnaker::GenApi, 352

DeepCopy  
    IImage, 826  
    Image, 864

DEFAULT  
    Spinnaker, 246

DefectCorrectionMode  
    Camera, 468

DefectCorrectionMode\_Average  
    Spinnaker, 255

DefectCorrectionMode\_Highlight  
    Spinnaker, 255

DefectCorrectionMode\_Zero  
    Spinnaker, 255

DefectCorrectionModeEnums  
    Spinnaker, 255

DefectCorrectStaticEnable  
    Camera, 468

DefectTableApply  
    Camera, 469

DefectTableCoordinateX  
    Camera, 469

DefectTableCoordinateY  
    Camera, 469

DefectTableFactoryRestore  
    Camera, 469

DefectTableIndex  
    Camera, 470

DefectTablePixelCount  
    Camera, 470

DefectTableSave  
    Camera, 470

DEFLATE  
    TIFFOption, 1062

Delnit  
    CameraBase, 564

ICameraBase, 786  
 Deinterlacing  
     Camera, 470  
 Deinterlacing\_LineDuplication  
     Spinnaker, 256  
 Deinterlacing\_Off  
     Spinnaker, 256  
 Deinterlacing\_Weave  
     Spinnaker, 256  
 DeinterlacingEnums  
     Spinnaker, 255  
 deleteFile  
     FileProtocolAdapter, 748  
 DeleteFileOnCamera  
     Inference.cpp, 1367  
 DeliverEventMessage  
     CEventAdapter1394, 613  
     CEventAdapterGEV, 617  
     CEventAdapterU3V, 619  
 DeliverMessage  
     CEventAdapter, 610  
     CEventAdapter1394, 613  
     CEventAdapterGeneric, 615  
     CEventAdapterGEV, 617  
     CEventAdapterU3V, 619  
 DEPRECATED\_CLASS  
     Spinnaker, 327  
 Deregister  
     Spinnaker::GenApi, 357  
 DeregisterCallback  
     Node, 985  
     Spinnaker::GenApi, 357  
 Destroy  
     CNodeCallback, 655  
     Function\_NodeCallback< Function >, 761  
     Member\_NodeCallback< Client, Member >, 980  
     NodeMap, 996  
 DetachBuffer  
     CChunkAdapter, 588  
 DetachChunk  
     CChunkPort, 602  
 DetachEvent  
     CEventPort, 623  
 DetachNode  
     CEventPort, 623  
 DetachNodeMap  
     CChunkAdapter, 588  
     CEventAdapter, 611  
 DetachPort  
     CChunkPort, 602  
 DETECTION  
     Inference.cpp, 1366  
 DeviceAccessStatus  
     TransportLayerDevice, 1066  
     TransportLayerInterface, 1076  
 DeviceAccessStatus\_Busy  
     Spinnaker, 256  
 DeviceAccessStatus\_NoAccess  
     Spinnaker, 256  
 DeviceAccessStatus\_OpenReadOnly  
     Spinnaker, 256  
 DeviceAccessStatus\_OpenReadWrite  
     Spinnaker, 256  
 DeviceAccessStatus\_ReadOnly  
     Spinnaker, 256  
 DeviceAccessStatus\_ReadWrite  
     Spinnaker, 256  
 DeviceAccessStatus\_Uncertain  
     Spinnaker, 256  
 DeviceAccessStatusUnknown  
     Spinnaker, 256  
 DeviceAccessStatusEnum  
     Spinnaker, 256  
 DeviceAddress  
     ActionCommandResult, 393  
 DeviceArrivalEventHandler, 702  
     ~DeviceArrivalEventHandler, 704  
     DeviceArrivalEventHandler, 703  
     OnDeviceArrival, 704  
     operator=, 704  
 DeviceArrivalEventHandler Class, 50  
 DeviceCharacterSet  
     Camera, 471  
 DeviceCharacterSet\_ASCII  
     Spinnaker, 256  
 DeviceCharacterSet\_UTF8  
     Spinnaker, 256  
 DeviceCharacterSetEnums  
     Spinnaker, 256  
 DeviceClockFrequency  
     Camera, 471  
 DeviceClockSelector  
     Camera, 471  
 DeviceClockSelector\_CameraLink  
     Spinnaker, 257  
 DeviceClockSelector\_Sensor  
     Spinnaker, 257  
 DeviceClockSelector\_SensorDigitization  
     Spinnaker, 257  
 DeviceClockSelectorEnums  
     Spinnaker, 256  
 DeviceConnectionSelector  
     Camera, 471  
 DeviceConnectionSpeed  
     Camera, 471  
 DeviceConnectionStatus  
     Camera, 472  
 DeviceConnectionStatus\_Active  
     Spinnaker, 257  
 DeviceConnectionStatus\_Inactive  
     Spinnaker, 257  
 DeviceConnectionStatusEnums  
     Spinnaker, 257  
 DeviceCount  
     TransportLayerInterface, 1077  
 DeviceCurrentSpeed  
     TransportLayerDevice, 1066  
 DeviceCurrentSpeed\_FullSpeed

Spinnaker, 257  
DeviceCurrentSpeed\_HighSpeed  
    Spinnaker, 257  
DeviceCurrentSpeed\_LowSpeed  
    Spinnaker, 257  
DeviceCurrentSpeed\_SuperSpeed  
    Spinnaker, 257  
DeviceCurrentSpeed\_UnknownSpeed  
    Spinnaker, 257  
DeviceCurrentSpeedEnum  
    Spinnaker, 257  
DeviceDisplayName  
    TransportLayerDevice, 1066  
DeviceDriverVersion  
    TransportLayerDevice, 1066  
DeviceEndianessMechanism  
    TransportLayerDevice, 1066  
DeviceEndianessMechanism\_Legacy  
    Spinnaker, 258  
DeviceEndianessMechanism\_Standard  
    Spinnaker, 258  
DeviceEndianessMechanismEnum  
    Spinnaker, 257  
DeviceEventChannelCount  
    Camera, 472  
DeviceEventHandler, 705  
    ~DeviceEventHandler, 706  
    DeviceEventHandler, 706  
    GetDeviceEventId, 706  
    GetDeviceEventName, 707  
    OnDeviceEvent, 707  
    operator=, 707  
DeviceEventHandler Class, 51  
DeviceEventHandlerImpl, 708  
    ~DeviceEventHandlerImpl, 709  
    DeviceEventHandlerImpl, 709  
    OnDeviceEvent, 710  
DeviceEvents.cpp  
    AcquireImages, 1337  
    chosenEvent, 1338  
    ConfigureDeviceEvents, 1337  
    eventType, 1336  
    GENERIC, 1337  
    main, 1337  
    PrintDeviceInfo, 1337  
    ResetDeviceEvents, 1337  
    RunSingleCamera, 1337  
    SPECIFIC, 1337  
DeviceFamilyName  
    Camera, 472  
DeviceFeaturePersistenceEnd  
    Camera, 472  
DeviceFeaturePersistenceStart  
    Camera, 472  
DeviceFirmwareVersion  
    Camera, 473  
DeviceGenCPVersionMajor  
    Camera, 473  
DeviceGenCPVersionMinor  
    Camera, 473  
DeviceID  
    Camera, 473  
    TransportLayerDevice, 1067  
    TransportLayerInterface, 1077  
DeviceIndicatorMode  
    Camera, 473  
DeviceIndicatorMode\_Active  
    Spinnaker, 258  
DeviceIndicatorMode\_ErrorStatus  
    Spinnaker, 258  
DeviceIndicatorMode\_Inactive  
    Spinnaker, 258  
DeviceIndicatorModeEnums  
    Spinnaker, 258  
DeviceInstanceId  
    TransportLayerDevice, 1067  
DeviceUpdater  
    TransportLayerDevice, 1067  
DeviceLinkBandwidthReserve  
    Camera, 473  
DeviceLinkCommandTimeout  
    Camera, 474  
DeviceLinkConnectionCount  
    Camera, 474  
DeviceLinkCurrentThroughput  
    Camera, 474  
DeviceLinkHeartbeatMode  
    Camera, 474  
DeviceLinkHeartbeatMode\_Off  
    Spinnaker, 258  
DeviceLinkHeartbeatMode\_On  
    Spinnaker, 258  
DeviceLinkHeartbeatModeEnums  
    Spinnaker, 258  
DeviceLinkHeartbeatTimeout  
    Camera, 474  
DeviceLinkSelector  
    Camera, 475  
DeviceLinkSpeed  
    Camera, 475  
    TransportLayerDevice, 1067  
DeviceLinkThroughputLimit  
    Camera, 475  
DeviceLinkThroughputLimitMode  
    Camera, 475  
DeviceLinkThroughputLimitMode\_Off  
    Spinnaker, 259  
DeviceLinkThroughputLimitMode\_On  
    Spinnaker, 259  
DeviceLinkThroughputLimitModeEnums  
    Spinnaker, 258  
DeviceLocation  
    TransportLayerDevice, 1067  
DeviceManifestEntrySelector  
    Camera, 475  
DeviceManifestPrimaryURL

Camera, 476  
 DeviceManifestSchemaMajorVersion  
     Camera, 476  
 DeviceManifestSchemaMinorVersion  
     Camera, 476  
 DeviceManifestSecondaryURL  
     Camera, 476  
 DeviceManifestXMLMajorVersion  
     Camera, 476  
 DeviceManifestXMLMinorVersion  
     Camera, 476  
 DeviceManifestXMLSubMinorVersion  
     Camera, 477  
 DeviceManufacturerInfo  
     Camera, 477  
 DeviceMaxThroughput  
     Camera, 477  
 DeviceModelName  
     Camera, 477  
     TransportLayerDevice, 1067  
     TransportLayerInterface, 1077  
 DeviceMulticastMonitorMode  
     TransportLayerDevice, 1068  
 DevicePowerSupplySelector  
     Camera, 477  
 DevicePowerSupplySelector\_External  
     Spinnaker, 259  
 DevicePowerSupplySelectorEnums  
     Spinnaker, 259  
 DeviceRegistersCheck  
     Camera, 478  
 DeviceRegistersEndianness  
     Camera, 478  
 DeviceRegistersEndianness\_Big  
     Spinnaker, 259  
 DeviceRegistersEndianness\_Little  
     Spinnaker, 259  
 DeviceRegistersEndiannessEnums  
     Spinnaker, 259  
 DeviceRegistersStreamingEnd  
     Camera, 478  
 DeviceRegistersStreamingStart  
     Camera, 478  
 DeviceRegistersValid  
     Camera, 478  
 DeviceRemovalEventHandler, 710  
     ~DeviceRemovalEventHandler, 712  
     DeviceRemovalEventHandler, 711  
     OnDeviceRemoval, 712  
     operator=, 712  
 DeviceRemovalEventHandler Class, 52  
 DeviceReset  
     Camera, 479  
 DeviceScanType  
     Camera, 479  
 DeviceScanType\_Areascan  
     Spinnaker, 259  
 DeviceScanTypeEnums  
     Spinnaker, 259  
 Spinnaker, 259  
 DeviceSelector  
     TransportLayerInterface, 1077  
 DeviceSerialNumber  
     Camera, 479  
     TransportLayerDevice, 1068  
     TransportLayerInterface, 1077  
 DeviceSerialPortBaudRate  
     Camera, 479  
 DeviceSerialPortBaudRate\_Baud115200  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud19200  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud230400  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud38400  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud460800  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud57600  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud921600  
     Spinnaker, 260  
 DeviceSerialPortBaudRate\_Baud9600  
     Spinnaker, 260  
 DeviceSerialPortBaudRateEnums  
     Spinnaker, 260  
 DeviceSerialPortSelector  
     Camera, 479  
 DeviceSerialPortSelector\_CameraLink  
     Spinnaker, 260  
 DeviceSerialPortSelectorEnums  
     Spinnaker, 260  
 DeviceSFNCVersionMajor  
     Camera, 480  
 DeviceSFNCVersionMinor  
     Camera, 480  
 DeviceSFNCVersionSubMinor  
     Camera, 480  
 DeviceStreamChannelCount  
     Camera, 480  
 DeviceStreamChannelEndianness  
     Camera, 480  
 DeviceStreamChannelEndianness\_Big  
     Spinnaker, 260  
 DeviceStreamChannelEndianness\_Little  
     Spinnaker, 260  
 DeviceStreamChannelEndiannessEnums  
     Spinnaker, 260  
 DeviceStreamChannelLink  
     Camera, 481  
 DeviceStreamChannelPacketSize  
     Camera, 481  
 DeviceStreamChannelSelector  
     Camera, 481  
 DeviceStreamChannelType  
     Camera, 481  
 DeviceStreamChannelType\_Receiver

Spinnaker, 261  
DeviceStreamChannelType\_Transmitter  
    Spinnaker, 261  
DeviceStreamChannelTypeEnums  
    Spinnaker, 261  
DeviceTapGeometry  
    Camera, 481  
DeviceTapGeometry\_Geometry\_10X  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_10X\_1Y  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X10  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X10\_1Y  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X2  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X2\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X2\_1Y2  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X2\_2YE  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X3  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X3\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X4  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X4\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X8  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_1X\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X\_1Y2  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_1X\_2YE  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_2X  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_2X2  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_2X2\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_2X2E  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_2X2E\_1Y  
    Spinnaker, 261  
DeviceTapGeometry\_Geometry\_2X2E\_2YE  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_2X2M  
    Spinnaker, 262  
DeviceTapGeometry\_Geometry\_2X\_1Y  
    Spinnaker, 261  
Spinnaker, 261  
DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry\_2X2M  
    DeviceTLType\_CameraLink  
        Spinnaker, 263  
    DeviceTLType\_CameraLinkHS  
        Spinnaker, 263  
    DeviceTLType\_CoaxPress  
        Spinnaker, 263  
    DeviceTLType\_Custom

Spinnaker, 263  
**DeviceTLType\_GigEVision**  
 Spinnaker, 263  
**DeviceTLType\_USB3Vision**  
 Spinnaker, 263  
**DeviceTLTypeEnums**  
 Spinnaker, 262  
**DeviceTLVersionMajor**  
 Camera, 482  
**DeviceTLVersionMinor**  
 Camera, 482  
**DeviceTLVersionSubMinor**  
 Camera, 482  
**DeviceType**  
 Camera, 483  
 TransportLayerDevice, 1068  
**DeviceType\_CameraLink**  
 Spinnaker, 263  
**DeviceType\_CameraLinkHS**  
 Spinnaker, 263  
**DeviceType\_CoaxPress**  
 Spinnaker, 263  
**DeviceType\_Custom**  
 Spinnaker, 263  
**DeviceType\_GigEVision**  
 Spinnaker, 263  
**DeviceType\_Peripheral**  
 Spinnaker, 263  
**DeviceType\_Receiver**  
 Spinnaker, 263  
**DeviceType\_Transceiver**  
 Spinnaker, 263  
**DeviceType\_Transmitter**  
 Spinnaker, 263  
**DeviceType\_USB3Vision**  
 Spinnaker, 263  
**DeviceTypeEnum**  
 Spinnaker, 263  
**DeviceTypeEnums**  
 Spinnaker, 263  
**DeviceU3VProtocol**  
 TransportLayerDevice, 1068  
**DeviceUnlock**  
 TransportLayerInterface, 1077  
**DeviceUpdateList**  
 TransportLayerInterface, 1078  
**DeviceUptime**  
 Camera, 483  
**DeviceUserID**  
 Camera, 483  
 TransportLayerDevice, 1068  
**DeviceVendorName**  
 Camera, 483  
 TransportLayerDevice, 1068  
 TransportLayerInterface, 1078  
**DeviceVersion**  
 Camera, 483  
 TransportLayerDevice, 1069

dhcpEnabled  
 AdapterInfo, 397  
**DIRECTIONAL\_FILTER**  
 Spinnaker, 246  
**DisableAll**  
 IImageStatistics, 838  
 ImageStatistics, 892  
**DisableChunkData**  
 ChunkData.cpp, 1333  
 Inference.cpp, 1367  
**DisableTrigger**  
 Inference.cpp, 1368  
**DiscoverMaxPacketSize**  
 CameraBase, 564  
 ICameraBase, 787  
**DisplayChunkData**  
 ChunkData.cpp, 1333  
 Inference.cpp, 1368  
**doc/spindocs/C++/GettingStarted.dox**, 1107  
**doc/spindocs/C++/ProgrammerGuide.dox**, 1107  
**doc/spindocs/shared/Benefits.dox**, 1107  
**doc/spindocs/shared/FlyCapture2Comparison.dox**, 1107  
**doc/spindocs/shared/GenICamGenTL.dox**, 1107  
**doc/spindocs/shared/Licensing.dox**, 1107  
**DoesEnvironmentVariableExist**  
 Spinnaker::GenICam, 387  
**double\_automvector\_t**, 712  
 \_pCount, 715  
 \_pv, 715  
 ~double\_automvector\_t, 713  
 double\_automvector\_t, 713  
 operator delete, 714  
 operator new, 714  
 operator=, 714  
 operator[], 714  
 size, 714  
**DownloadImage**  
 FileAccess\_QuickSpin.cpp, 1346  
**dummy**  
 CpuUsageInfo, 690

**EAccessModeClass**, 715  
 FromString, 715  
 ToString, 716  
**EatComments**  
 Spinnaker::GenApi, 357  
**ECacheUsage\_t**  
 Spinnaker::GenApi, 353  
**ECachingModeClass**, 716  
 FromString, 717  
 ToString, 717  
**EContentType\_t**  
 Spinnaker::GenApi, 354  
**EDGE\_SENSING**  
 Spinnaker, 246  
**EDisplayNotationClass**, 717  
 FromString, 718  
 ToString, 718

EEndianessClass, 718  
  FromString, 719  
  ToString, 719  
EGenApiSchemaVersionClass, 720  
  FromString, 720  
  ToString, 720  
EInputDirectionClass, 721  
  FromString, 721  
  ToString, 721  
empty  
  gcstring, 765  
EnableAll  
  IImageStatistics, 838  
  ImageStatistics, 892  
EnableGreyOnly  
  IImageStatistics, 838  
  ImageStatistics, 892  
EnableHSLOnly  
  IImageStatistics, 838  
  ImageStatistics, 893  
EnableManualFramerate  
  GigEVisionPerformance.cpp, 1352  
EnableRGBOnly  
  IImageStatistics, 839  
  ImageStatistics, 893  
ENameSpaceClass, 722  
  FromString, 722  
  ToString, 722  
EncoderDivider  
  Camera, 483  
EncoderMode  
  Camera, 484  
EncoderMode\_FourPhase  
  Spinnaker, 264  
EncoderMode\_HighResolution  
  Spinnaker, 264  
EncoderModeEnums  
  Spinnaker, 263  
EncoderOutputMode  
  Camera, 484  
EncoderOutputMode\_DirectionDown  
  Spinnaker, 264  
EncoderOutputMode\_DirectionUp  
  Spinnaker, 264  
EncoderOutputMode\_Motion  
  Spinnaker, 264  
EncoderOutputMode\_Off  
  Spinnaker, 264  
EncoderOutputMode\_PositionDown  
  Spinnaker, 264  
EncoderOutputMode\_PositionUp  
  Spinnaker, 264  
EncoderOutputModeEnums  
  Spinnaker, 264  
EncoderReset  
  Camera, 484  
EncoderResetActivation  
  Camera, 484  
EncoderResetActivation\_AnyEdge  
  Spinnaker, 265  
EncoderResetActivation\_FallingEdge  
  Spinnaker, 265  
EncoderResetActivation\_LevelHigh  
  Spinnaker, 265  
EncoderResetActivation\_LevelLow  
  Spinnaker, 265  
EncoderResetActivation\_RisingEdge  
  Spinnaker, 265  
EncoderResetActivationEnums  
  Spinnaker, 264  
EncoderResetSource  
  Camera, 484  
EncoderResetSource\_AcquisitionEnd  
  Spinnaker, 265  
EncoderResetSource\_AcquisitionStart  
  Spinnaker, 265  
EncoderResetSource\_AcquisitionTrigger  
  Spinnaker, 265  
EncoderResetSource\_Action0  
  Spinnaker, 266  
EncoderResetSource\_Action1  
  Spinnaker, 266  
EncoderResetSource\_Action2  
  Spinnaker, 266  
EncoderResetSource\_Counter0End  
  Spinnaker, 265  
EncoderResetSource\_Counter0Start  
  Spinnaker, 265  
EncoderResetSource\_Counter1End  
  Spinnaker, 265  
EncoderResetSource\_Counter1Start  
  Spinnaker, 265  
EncoderResetSource\_Counter2End  
  Spinnaker, 265  
EncoderResetSource\_Counter2Start  
  Spinnaker, 265  
EncoderResetSource\_ExposureEnd  
  Spinnaker, 265  
EncoderResetSource\_ExposureStart  
  Spinnaker, 265  
EncoderResetSource\_FrameEnd  
  Spinnaker, 265  
EncoderResetSource\_FrameStart  
  Spinnaker, 265  
EncoderResetSource\_FrameTrigger  
  Spinnaker, 265  
EncoderResetSource\_Line0  
  Spinnaker, 265  
EncoderResetSource\_Line1  
  Spinnaker, 265  
EncoderResetSource\_Line2  
  Spinnaker, 265  
EncoderResetSource\_LinkTrigger0  
  Spinnaker, 266  
EncoderResetSource\_LinkTrigger1  
  Spinnaker, 266

EncoderResetSource\_LinkTrigger2  
    Spinnaker, 266  
EncoderResetSource\_Off  
    Spinnaker, 265  
EncoderResetSource\_SoftwareSignal0  
    Spinnaker, 265  
EncoderResetSource\_SoftwareSignal1  
    Spinnaker, 265  
EncoderResetSource\_SoftwareSignal2  
    Spinnaker, 266  
EncoderResetSource\_Timer0End  
    Spinnaker, 265  
EncoderResetSource\_Timer0Start  
    Spinnaker, 265  
EncoderResetSource\_Timer1End  
    Spinnaker, 265  
EncoderResetSource\_Timer1Start  
    Spinnaker, 265  
EncoderResetSource\_Timer2End  
    Spinnaker, 265  
EncoderResetSource\_Timer2Start  
    Spinnaker, 265  
EncoderResetSource\_UserOutput0  
    Spinnaker, 265  
EncoderResetSource\_UserOutput1  
    Spinnaker, 265  
EncoderResetSource\_UserOutput2  
    Spinnaker, 265  
EncoderResetSourceEnums  
    Spinnaker, 265  
EncoderSelector  
    Camera, 484  
EncoderSelector\_Encoder0  
    Spinnaker, 266  
EncoderSelector\_Encoder1  
    Spinnaker, 266  
EncoderSelector\_Encoder2  
    Spinnaker, 266  
EncoderSelectorEnums  
    Spinnaker, 266  
EncoderSourceA  
    Camera, 485  
EncoderSourceA\_Line0  
    Spinnaker, 266  
EncoderSourceA\_Line1  
    Spinnaker, 266  
EncoderSourceA\_Line2  
    Spinnaker, 266  
EncoderSourceA\_Off  
    Spinnaker, 266  
EncoderSourceAEnums  
    Spinnaker, 266  
EncoderSourceB  
    Camera, 485  
EncoderSourceB\_Line0  
    Spinnaker, 267  
EncoderSourceB\_Line1  
    Spinnaker, 267  
EncoderSourceB\_Line2  
    Spinnaker, 267  
EncoderSourceB\_Off  
    Spinnaker, 267  
EncoderSourceBEnums  
    Spinnaker, 267  
EncoderStatus  
    Camera, 485  
EncoderStatus\_EncoderDown  
    Spinnaker, 267  
EncoderStatus\_EncoderIdle  
    Spinnaker, 267  
EncoderStatus\_EncoderStatic  
    Spinnaker, 267  
EncoderStatus\_EncoderUp  
    Spinnaker, 267  
EncoderStatusEnums  
    Spinnaker, 267  
EncoderTimeout  
    Camera, 485  
EncoderValue  
    Camera, 485  
EncoderValueAtReset  
    Camera, 485  
EndAcquisition  
    CameraBase, 565  
    ICameraBase, 787  
endTime  
    SecondsCounter, 190  
EnumClasses Class, 107  
EnumEntryNode, 723  
    ~EnumEntryNode, 724  
    EnumEntryNode, 724  
    GetNumericValue, 724  
    GetSymbolic, 725  
    GetValue, 725  
    IsSelfClearing, 725  
    SetReference, 725  
EnumEntryNode Class, 108  
EnumerateGEVInterfaces  
    TransportLayerSystem, 1094  
Enumeration.cpp  
    main, 1338  
    QueryInterface, 1338  
Enumeration\_QuickSpin.cpp  
    main, 1339  
    QueryInterface, 1339  
EnumerationCount  
    Camera, 486  
EnumerationEvents.cpp  
    CheckGevEnabled, 1340  
    main, 1340  
EnumNode, 726  
    ~EnumNode, 728  
    EnumNode, 728  
    GetCurrentEntry, 728  
    GetEntries, 728  
    GetEntry, 729

GetEntryByName, 729  
GetIntValue, 729  
GetSymbolics, 729  
m\_pEnumeration, 731  
operator\*, 730  
operator=, 730  
SetIntValue, 730  
SetReference, 730  
EnumNode Class, 109  
EnumNodeT Class, 110  
ERepresentationClass, 731  
    FromString, 731  
    ToString, 732  
ErrCode  
    AdapterConfigException, 395  
Error  
    Spinnaker, 267  
ESignClass, 732  
    FromString, 733  
    ToString, 733  
ESlopeClass, 733  
    FromString, 734  
    ToString, 734  
EStandardNameSpaceClass, 734  
    FromString, 735  
    ToString, 735  
Event  
    GVCP\_EVENTDATA\_REQUEST, 779  
    GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, 780  
EVENT\_TIMEOUT\_INFINITE  
    Spinnaker, 329  
EVENT\_TIMEOUT\_NONE  
    Spinnaker, 329  
EventAcquisitionEnd  
    Camera, 486  
EventAcquisitionEndFrameID  
    Camera, 486  
EventAcquisitionEndTimestamp  
    Camera, 486  
EventAcquisitionError  
    Camera, 486  
EventAcquisitionErrorHandlerFrameID  
    Camera, 486  
EventAcquisitionErrorTimestamp  
    Camera, 487  
EventAcquisitionStart  
    Camera, 487  
EventAcquisitionStartFrameID  
    Camera, 487  
EventAcquisitionStartTimestamp  
    Camera, 487  
EventAcquisitionTransferEnd  
    Camera, 487  
EventAcquisitionTransferEndFrameID  
    Camera, 487  
EventAcquisitionTransferEndTimestamp  
    Camera, 488  
EventAcquisitionTransferStart  
    Camera, 488  
EventAcquisitionTransferStartFrameID  
    Camera, 488  
EventAcquisitionTransferStartTimestamp  
    Camera, 488  
EventAcquisitionTrigger  
    Camera, 488  
EventAcquisitionTriggerFrameID  
    Camera, 488  
EventAcquisitionTriggerTimestamp  
    Camera, 489  
EventActionLate  
    Camera, 489  
EventActionLateFrameID  
    Camera, 489  
EventActionLateTimestamp  
    Camera, 489  
EventAdapter Class, 111  
EventAdapter1394 Class, 112  
EventAdapterGeneric Class, 113  
EventAdapterGEV Class, 114  
EventAdapterU3V Class, 115  
EventCounter0End  
    Camera, 489  
EventCounter0EndFrameID  
    Camera, 489  
EventCounter0EndTimestamp  
    Camera, 490  
EventCounter0Start  
    Camera, 490  
EventCounter0StartFrameID  
    Camera, 490  
EventCounter0StartTimestamp  
    Camera, 490  
EventCounter1End  
    Camera, 490  
EventCounter1EndFrameID  
    Camera, 490  
EventCounter1EndTimestamp  
    Camera, 491  
EventCounter1Start  
    Camera, 491  
EventCounter1StartFrameID  
    Camera, 491  
EventCounter1StartTimestamp  
    Camera, 491  
EventData  
    U3V\_EVENT\_MESSAGE, 1102  
EventEncoder0Restarted  
    Camera, 491  
EventEncoder0RestartedFrameID  
    Camera, 491  
EventEncoder0RestartedTimestamp  
    Camera, 492  
EventEncoder0Stopped  
    Camera, 492  
EventEncoder0StoppedFrameID

Camera, 492  
 EventEncoder0StoppedTimestamp  
     Camera, 492  
 EventEncoder1Restarted  
     Camera, 492  
 EventEncoder1RestartedFrameID  
     Camera, 492  
 EventEncoder1RestartedTimestamp  
     Camera, 493  
 EventEncoder1Stopped  
     Camera, 493  
 EventEncoder1StoppedFrameID  
     Camera, 493  
 EventEncoder1StoppedTimestamp  
     Camera, 493  
 EventError  
     Camera, 493  
 EventErrorCode  
     Camera, 493  
 EventErrorFrameID  
     Camera, 494  
 EventErrorTimestamp  
     Camera, 494  
 EventExposureEnd  
     Camera, 494  
 EventExposureEndFrameID  
     Camera, 494  
 EventExposureEndTimestamp  
     Camera, 494  
 EventExposureStart  
     Camera, 494  
 EventExposureStartFrameID  
     Camera, 495  
 EventExposureStartTimestamp  
     Camera, 495  
 EventFrameBurstEnd  
     Camera, 495  
 EventFrameBurstEndFrameID  
     Camera, 495  
 EventFrameBurstEndTimestamp  
     Camera, 495  
 EventFrameBurstStart  
     Camera, 495  
 EventFrameBurstStartFrameID  
     Camera, 496  
 EventFrameBurstStartTimestamp  
     Camera, 496  
 EventFrameEnd  
     Camera, 496  
 EventFrameEndFrameID  
     Camera, 496  
 EventFrameEndTimestamp  
     Camera, 496  
 EventFrameStart  
     Camera, 496  
 EventFrameStartFrameID  
     Camera, 497  
 EventFrameStartTimestamp  
     Camera, 497  
 Camera, 497  
 EventFrameTransferEnd  
     Camera, 497  
 EventFrameTransferEndFrameID  
     Camera, 497  
 EventFrameTransferEndTimestamp  
     Camera, 497  
 EventFrameTransferStart  
     Camera, 497  
 EventFrameTransferStartFrameID  
     Camera, 498  
 EventFrameTransferStartTimestamp  
     Camera, 498  
 EventFrameTrigger  
     Camera, 498  
 EventFrameTriggerFrameID  
     Camera, 498  
 EventFrameTriggerTimestamp  
     Camera, 498  
 EventHandler, 736  
     ~EventHandler, 737  
     EventHandler, 737  
     EventProcessor, 738  
     GetEventPayloadData, 737  
     GetEventPayloadDataSize, 737  
     GetEventType, 737  
     IDataStream, 738  
     m\_pEventData, 739  
     operator=, 738  
     SetEventPayload, 738  
     SetEventType, 738  
     Stream, 739  
 EventHandler Class, 53  
 EventId  
     GVCP\_EVENT\_ITEM, 773  
     GVCP\_EVENT\_ITEM\_BASIC, 774  
     GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776  
     U3V\_EVENT\_DATA, 1101  
 EventLine0AnyEdge  
     Camera, 498  
 EventLine0AnyEdgeFrameID  
     Camera, 499  
 EventLine0AnyEdgeTimestamp  
     Camera, 499  
 EventLine0FallingEdge  
     Camera, 499  
 EventLine0FallingEdgeFrameID  
     Camera, 499  
 EventLine0FallingEdgeTimestamp  
     Camera, 499  
 EventLine0RisingEdge  
     Camera, 499  
 EventLine0RisingEdgeFrameID  
     Camera, 500  
 EventLine0RisingEdgeTimestamp  
     Camera, 500  
 EventLine1AnyEdge  
     Camera, 500

EventLine1AnyEdgeFrameID  
    Camera, 500  
EventLine1AnyEdgeTimestamp  
    Camera, 500  
EventLine1FallingEdge  
    Camera, 500  
EventLine1FallingEdgeFrameID  
    Camera, 501  
EventLine1FallingEdgeTimestamp  
    Camera, 501  
EventLine1RisingEdge  
    Camera, 501  
EventLine1RisingEdgeFrameID  
    Camera, 501  
EventLine1RisingEdgeTimestamp  
    Camera, 501  
EventLinkSpeedChange  
    Camera, 501  
EventLinkSpeedChangeFrameID  
    Camera, 502  
EventLinkSpeedChangeTimestamp  
    Camera, 502  
EventLinkTrigger0  
    Camera, 502  
EventLinkTrigger0FrameID  
    Camera, 502  
EventLinkTrigger0Timestamp  
    Camera, 502  
EventLinkTrigger1  
    Camera, 502  
EventLinkTrigger1FrameID  
    Camera, 503  
EventLinkTrigger1Timestamp  
    Camera, 503  
EventNotification  
    Camera, 503  
EventNotification\_Off  
    Spinnaker, 270  
EventNotification\_On  
    Spinnaker, 270  
EventNotificationEnums  
    Spinnaker, 268  
EventPort Class, 116  
EventProcessor  
    EventHandler, 738  
EventSelector  
    Camera, 503  
EventSelector\_Error  
    Spinnaker, 270  
EventSelector\_ExposureEnd  
    Spinnaker, 270  
EventSelector\_SerialPortReceive  
    Spinnaker, 270  
EventSelectorEnums  
    Spinnaker, 270  
EventSequencerSetChange  
    Camera, 503  
EventSequencerSetChangeFrameID  
    Camera, 503  
EventSequencerSetChangeTimestamp  
    Camera, 504  
EventSerialData  
    Camera, 504  
EventSerialDataLength  
    Camera, 504  
EventSerialPortReceive  
    Camera, 504  
EventSerialPortReceiveTimestamp  
    Camera, 504  
EventSerialReceiveOverflow  
    Camera, 504  
EventStream0TransferBlockEnd  
    Camera, 505  
EventStream0TransferBlockEndFrameID  
    Camera, 505  
EventStream0TransferBlockEndTimestamp  
    Camera, 505  
EventStream0TransferBlockStart  
    Camera, 505  
EventStream0TransferBlockStartFrameID  
    Camera, 505  
EventStream0TransferBlockStartTimestamp  
    Camera, 505  
EventStream0TransferBlockTrigger  
    Camera, 506  
EventStream0TransferBlockTriggerFrameID  
    Camera, 506  
EventStream0TransferBlockTriggerTimestamp  
    Camera, 506  
EventStream0TransferBurstEnd  
    Camera, 506  
EventStream0TransferBurstEndFrameID  
    Camera, 506  
EventStream0TransferBurstEndTimestamp  
    Camera, 506  
EventStream0TransferBurstStart  
    Camera, 507  
EventStream0TransferBurstStartFrameID  
    Camera, 507  
EventStream0TransferBurstStartTimestamp  
    Camera, 507  
EventStream0TransferEnd  
    Camera, 507  
EventStream0TransferEndFrameID  
    Camera, 507  
EventStream0TransferEndTimestamp  
    Camera, 507  
EventStream0TransferOverflow  
    Camera, 508  
EventStream0TransferOverflowFrameID  
    Camera, 508  
EventStream0TransferOverflowTimestamp  
    Camera, 508  
EventStream0TransferPause  
    Camera, 508  
EventStream0TransferPauseFrameID  
    Camera, 508

Camera, 508  
 EventStream0TransferPauseTimestamp  
     Camera, 508  
 EventStream0TransferResume  
     Camera, 509  
 EventStream0TransferResumeFrameID  
     Camera, 509  
 EventStream0TransferResumeTimestamp  
     Camera, 509  
 EventStream0TransferStart  
     Camera, 509  
 EventStream0TransferStartFrameID  
     Camera, 509  
 EventStream0TransferStartTimestamp  
     Camera, 509  
 EventTest  
     Camera, 510  
 EventTestTimestamp  
     Camera, 510  
 EventTimer0End  
     Camera, 510  
 EventTimer0EndFrameID  
     Camera, 510  
 EventTimer0EndTimestamp  
     Camera, 510  
 EventTimer0Start  
     Camera, 510  
 EventTimer0StartFrameID  
     Camera, 511  
 EventTimer0StartTimestamp  
     Camera, 511  
 EventTimer1End  
     Camera, 511  
 EventTimer1EndFrameID  
     Camera, 511  
 EventTimer1EndTimestamp  
     Camera, 511  
 EventTimer1Start  
     Camera, 511  
 EventTimer1StartFrameID  
     Camera, 512  
 EventTimer1StartTimestamp  
     Camera, 512  
 EventType  
     Spinnaker, 270  
 eventType  
     DeviceEvents.cpp, 1336  
 EVisibilityClass, 739  
     FromString, 739  
     ToString, 740  
 Exception, 740  
     ~Exception, 743  
     Exception, 742, 743  
     GetBuildDate, 743  
     GetBuildTime, 743  
     GetError, 743  
     GetErrorMessage, 743  
     GetFileName, 744  
     GetFullErrorMessage, 744  
     GetFunctionName, 744  
     GetLineNumber, 744  
     operator!=, 744  
     operator=, 744  
     operator==, 745  
     what, 745  
 Exception Class, 54  
 ExceptionHandling.cpp  
     causeSpinnakerException, 1341  
     causeStandardException, 1341  
     chosenException, 1342  
     exceptionType, 1341  
     main, 1341  
     SPINNAKER\_EXCEPTION, 1341  
     STANDARD\_CAST\_TO\_SPINNAKER, 1341  
     STANDARD\_EXCEPTION, 1341  
 exceptionType  
     ExceptionHandling.cpp, 1341  
 Execute  
     CommandNode, 676  
 ExecuteDeleteCommand  
     FileAccess\_QuickSpin.cpp, 1346  
 ExecuteReadCommand  
     FileAccess\_QuickSpin.cpp, 1346  
 ExecuteWriteCommand  
     FileAccess\_QuickSpin.cpp, 1346  
 EXPAND\_TO\_STRINGISE  
     GCUtilities.h, 1210  
 Expert  
     Spinnaker::GenApi, 352  
 Exposure.cpp  
     AcquireImages, 1342  
     ConfigureExposure, 1342  
     main, 1342  
     PrintDeviceInfo, 1343  
     ResetExposure, 1343  
     RunSingleCamera, 1343  
 Exposure\_QuickSpin.cpp  
     AcquireImages, 1344  
     ConfigureExposure, 1344  
     main, 1344  
     PrintDeviceInfo, 1344  
     ResetExposure, 1344  
     RunSingleCamera, 1344  
 ExposureActiveMode  
     Camera, 512  
 ExposureActiveMode\_AllPixels  
     Spinnaker, 271  
 ExposureActiveMode\_AnyPixels  
     Spinnaker, 271  
 ExposureActiveMode\_Line1  
     Spinnaker, 271  
 ExposureActiveModeEnums  
     Spinnaker, 270  
 ExposureAuto  
     Camera, 512  
 ExposureAuto\_Continuous

Spinnaker, 271  
ExposureAuto\_Off  
    Spinnaker, 271  
ExposureAuto.Once  
    Spinnaker, 271  
ExposureAutoEnums  
    Spinnaker, 271  
ExposureMode  
    Camera, 512  
ExposureMode\_Timed  
    Spinnaker, 271  
ExposureMode\_TriggerWidth  
    Spinnaker, 271  
ExposureModeEnums  
    Spinnaker, 271  
ExposureTime  
    Camera, 512  
ExposureTimeMode  
    Camera, 513  
ExposureTimeMode\_Common  
    Spinnaker, 273  
ExposureTimeMode\_Individual  
    Spinnaker, 273  
ExposureTimeModeEnums  
    Spinnaker, 271  
ExposureTimeSelector  
    Camera, 513  
ExposureTimeSelector\_Blue  
    Spinnaker, 273  
ExposureTimeSelector\_Common  
    Spinnaker, 273  
ExposureTimeSelector\_Cyan  
    Spinnaker, 273  
ExposureTimeSelector\_Green  
    Spinnaker, 273  
ExposureTimeSelector\_Infrared  
    Spinnaker, 273  
ExposureTimeSelector\_Magenta  
    Spinnaker, 273  
ExposureTimeSelector\_Red  
    Spinnaker, 273  
ExposureTimeSelector\_Stage1  
    Spinnaker, 273  
ExposureTimeSelector\_Stage2  
    Spinnaker, 273  
ExposureTimeSelector\_Ultraviolet  
    Spinnaker, 273  
ExposureTimeSelector\_Yellow  
    Spinnaker, 273  
ExposureTimeSelectorEnums  
    Spinnaker, 273  
ExtractAndSavePolarQuadImages  
    Polarization.cpp, 1382  
ExtractIndependentSubtree  
    Spinnaker::GenApi, 358  
ExtractPolarQuadrant  
    ImageUtilityPolarization, 913  
ExtractSubtree  
    CNodeMapFactory, 662  
    EYesNoClass, 745  
        FromString, 746  
        ToString, 746  
    FactoryReset  
        Camera, 513  
    FileAccess\_QuickSpin.cpp  
        \_enableDebug, 1348  
        \_fileSelector, 1348  
        AcquireImages, 1345  
        CloseFile, 1345  
        DownloadImage, 1346  
        ExecuteDeleteCommand, 1346  
        ExecuteReadCommand, 1346  
        ExecuteWriteCommand, 1346  
        InitializeSystem, 1346  
        main, 1346  
        OpenFileToRead, 1346  
        OpenFileToWrite, 1347  
        PrintDebugMessage, 1347  
        PrintDeviceInfo, 1347  
        PrintResultMessage, 1347  
        PrintUsage, 1347  
        UploadImage, 1347  
    FileAccessBuffer  
        Camera, 513  
    FileAccessLength  
        Camera, 513  
    FileAccessOffset  
        Camera, 514  
    filebuf\_type  
        IDevFileStreamBase< CharType, Traits >, 813  
        ODevFileStreamBase< CharType, Traits >, 1004  
FileOpenMode  
    Camera, 514  
FileOpenMode\_Read  
    Spinnaker, 273  
FileOpenMode\_ReadWrite  
    Spinnaker, 273  
FileOpenMode\_Write  
    Spinnaker, 273  
FileOpenModeEnums  
    Spinnaker, 273  
FileOperationExecute  
    Camera, 514  
FileOperationResult  
    Camera, 514  
FileOperationSelector  
    Camera, 514  
FileOperationSelector\_Close  
    Spinnaker, 274  
FileOperationSelector\_Delete  
    Spinnaker, 274  
FileOperationSelector\_Open  
    Spinnaker, 274  
FileOperationSelector\_Read  
    Spinnaker, 274  
FileOperationSelector\_Write

Spinnaker, 274  
**FileOperationSelectorEnums**  
 Spinnaker, 274  
**FileOperationStatus**  
 Camera, 515  
**FileOperationStatus\_Failure**  
 Spinnaker, 274  
**FileOperationStatus\_Overflow**  
 Spinnaker, 274  
**FileOperationStatus\_Success**  
 Spinnaker, 274  
**FileOperationStatusEnums**  
 Spinnaker, 274  
**FileProtocolAdapter**, 746  
 ~FileProtocolAdapter, 747  
 attach, 747  
 closeFile, 748  
 deleteFile, 748  
 FileProtocolAdapter, 747  
 getBufSize, 748  
 openFile, 749  
 read, 749  
 write, 750  
**FileSelector**  
 Camera, 515  
**FileSelector\_SerialPort0**  
 Spinnaker, 274  
**FileSelector\_UserFile1**  
 Spinnaker, 274  
**FileSelector\_UserSet0**  
 Spinnaker, 274  
**FileSelector\_UserSet1**  
 Spinnaker, 274  
**FileSelector\_UserSetDefault**  
 Spinnaker, 274  
**FileSelectorEnums**  
 Spinnaker, 274  
**FileSize**  
 Camera, 515  
**Filestream Class**, 117  
**FileUploadPersistence**  
 Inference.cpp, 1365  
**FilterDriverStatus**  
 TransportLayerInterface, 1078  
**FilterDriverStatus\_Disabled**  
 Spinnaker, 275  
**FilterDriverStatus\_Enabled**  
 Spinnaker, 275  
**FilterDriverStatus\_NotSupported**  
 Spinnaker, 275  
**FilterDriverStatusEnum**  
 Spinnaker, 275  
**find**  
 gcstring, 765, 766  
**find\_first\_not\_of**  
 gcstring, 766  
**find\_first\_of**  
 gcstring, 766  
**fixedIncrement**  
 Spinnaker::GenApi, 349  
**Flags**  
 GVCP\_REQUEST\_HEADER, 781  
 U3V\_COMMAND\_HEADER, 1100  
**FLASH**  
 Inference.cpp, 1365  
**FLIR\_SPINNAKER\_VERSION\_BUILD**  
 System.h, 1275  
**FLIR\_SPINNAKER\_VERSION\_MAJOR**  
 System.h, 1275  
**FLIR\_SPINNAKER\_VERSION\_MINOR**  
 System.h, 1276  
**FLIR\_SPINNAKER\_VERSION\_TYPE**  
 System.h, 1276  
**float32\_t**  
 GCTypes Class, 122  
**float64\_t**  
 GCTypes Class, 122  
**FloatNode**, 750  
 ~FloatNode, 753  
 FloatNode, 753  
 GetDisplayNotation, 753  
 GetDisplayPrecision, 753  
 GetEnumAlias, 753  
 GetInc, 753  
 GetIncMode, 754  
 GetIntAlias, 754  
 GetListOfValidValues, 754  
 GetMax, 754  
 GetMin, 754  
 GetRepresentation, 754  
 GetUnit, 755  
 GetValue, 755  
 HasInc, 755  
 ImposeMax, 755  
 ImposeMin, 755  
 operator\*, 756  
 operator(), 756  
 operator=, 756  
 SetReference, 756  
 SetValue, 756  
**FloatNode Class**, 118  
**FloatRegNode**, 757  
 ~FloatRegNode, 759  
 FloatRegNode, 758, 759  
 SetReference, 759  
**FloatRegNode Class**, 119  
**FlushQueueAllDiscard**  
 IDataStream, 808  
**FMT\_I64**  
 Compatibility.h, 1188  
**fnAutomatic**  
 Spinnaker::GenApi, 348  
**fnFixed**  
 Spinnaker::GenApi, 348  
**fnScientific**  
 Spinnaker::GenApi, 348

ForceIP  
    CameraBase, 565  
    ICameraBase, 787

frameRate  
    AVIOption, 402  
    H264Option, 783  
    MJPGOption, 981

FROM\_FILE\_EXT  
    Spinnaker, 282

FromString  
    EAccessModeClass, 715  
    ECachingModeClass, 717  
    EDisplayNotationClass, 718  
    EEndianessClass, 719  
    EGenApiSchemaVersionClass, 720  
    EInputDirectionClass, 721  
    ENameSpaceClass, 722  
    ERepresentationClass, 731  
    ESignClass, 733  
    ESlopeClass, 734  
    EStandardNameSpaceClass, 735  
    EVisibilityClass, 739  
    EYesNoClass, 746  
    Spinnaker::GenApi, 358  
    ValueNode, 1104

Function\_NodeCallback  
    Function\_NodeCallback< Function >, 761

Function\_NodeCallback< Function >, 760  
    Destroy, 761  
    Function\_NodeCallback, 761  
    operator(), 761

Gain  
    Camera, 515

GainAuto  
    Camera, 515

GainAuto\_Continuous  
    Spinnaker, 275

GainAuto\_Off  
    Spinnaker, 275

GainAuto\_Once  
    Spinnaker, 275

GainAutoBalance  
    Camera, 516

GainAutoBalance\_Continuous  
    Spinnaker, 275

GainAutoBalance\_Off  
    Spinnaker, 275

GainAutoBalance\_Once  
    Spinnaker, 275

GainAutoBalanceEnums  
    Spinnaker, 275

GainAutoEnums  
    Spinnaker, 275

GainSelector  
    Camera, 516

GainSelector\_All  
    Spinnaker, 276

GainSelectorEnums  
    Spinnaker, 282

Spinnaker, 276

Gamma  
    Camera, 516

GammaEnable  
    Camera, 516

gateway  
    IpInfo, 954

GC\_COUNTOF  
    GCUtilities.h, 1211

GC\_INT32\_MAX  
    GCTypes.h, 1206

GC\_INT32\_MIN  
    GCTypes.h, 1206

GC\_INT64\_MAX  
    GCTypes.h, 1206

GC\_INT64\_MIN  
    GCTypes.h, 1206

GC\_INT8\_MAX  
    GCTypes.h, 1207

GC\_INT8\_MIN  
    GCTypes.h, 1207

GC\_UINT32\_MAX  
    GCTypes.h, 1207

GC\_UINT64\_MAX  
    GCTypes.h, 1207

GC\_UINT8\_MAX  
    GCTypes.h, 1207

gcstring, 762  
    npos, 764  
    ~gcstring, 764  
    append, 764  
    assign, 764, 765  
    c\_str, 765  
    compare, 765  
    empty, 765  
    find, 765, 766  
    find\_first\_not\_of, 766  
    find\_first\_of, 766  
    gcstring, 763  
    length, 767  
    max\_size, 767  
    npos, 771  
    operator const char \*, 767  
    operator delete, 767  
    operator new, 767  
    operator!=, 768  
    operator<, 769  
    operator>, 769  
    operator+, 770  
    operator+=, 768  
    operator=, 769  
    operator==, 769  
    resize, 769  
    size, 769  
    substr, 770  
    swap, 770

GCString Class, 120

GCString.h

GCSTRING\_NPOS, 1202  
 operator<<, 1202  
 operator>>, 1203  
**GCSTRING\_NPOS**  
 GCString.h, 1202  
**GCSynch Class**, 121  
**GCTypes Class**, 122  
 float32\_t, 122  
 float64\_t, 122  
**GCTypes.h**  
 \_\_STDC\_CONSTANT\_MACROS, 1206  
 \_\_STDC\_LIMIT\_MACROS, 1206  
 GC\_INT32\_MAX, 1206  
 GC\_INT32\_MIN, 1206  
 GC\_INT64\_MAX, 1206  
 GC\_INT64\_MIN, 1206  
 GC\_INT8\_MAX, 1207  
 GC\_INT8\_MIN, 1207  
 GC\_UINT32\_MAX, 1207  
 GC\_UINT64\_MAX, 1207  
 GC\_UINT8\_MAX, 1207  
**GCUtilities Utility**, 124  
**GCUtilities.h**  
 \_TO\_STRING, 1210  
 \_\_ERR\_\_, 1209  
 \_\_LINE\_STR\_\_, 1210  
 \_\_LOCATION\_\_, 1210  
 \_\_OUTPUT\_FORMATTER\_\_, 1210  
 \_\_TODO\_\_, 1210  
 \_\_WARN\_\_, 1210  
 EXPAND\_TO\_STRINGISE, 1210  
 GC\_COUNTOF, 1211  
 GENICAM\_DEPRECATED, 1211  
 GENICAM\_UNUSED, 1211  
 USE\_TEMP\_CACHE\_FILE, 1211  
**GENCP\_COMMAND\_HEADER\_SIZE**  
 Spinnaker::GenApi, 379  
**GENCP\_EVENT\_BASIC\_SIZE**  
 Spinnaker::GenApi, 379  
**GENCP\_EVENT\_CMD\_ID**  
 Spinnaker::GenApi, 379  
**GENERIC**  
 DeviceEvents.cpp, 1337  
**GENICAM\_DEPRECATED**  
 GCUtilities.h, 1211  
**GENICAM\_ERR\_ACCESS**  
 Spinnaker, 268  
**GENICAM\_ERR\_BAD\_ALLOCATION**  
 Spinnaker, 268  
**GENICAM\_ERR\_DYNAMIC\_CAST**  
 Spinnaker, 268  
**GENICAM\_ERR\_GENERIC**  
 Spinnaker, 268  
**GENICAM\_ERR\_INVALID\_ARGUMENT**  
 Spinnaker, 268  
**GENICAM\_ERR\_LOGICAL**  
 Spinnaker, 268  
**GENICAM\_ERR\_OUT\_OF\_RANGE**  
 Spinnaker, 268  
**Spinnaker**, 268  
**GENICAM\_ERR\_PROPERTY**  
 Spinnaker, 268  
**GENICAM\_ERR\_RUN\_TIME**  
 Spinnaker, 268  
**GENICAM\_ERR\_TIMEOUT**  
 Spinnaker, 268  
**GENICAM\_UNUSED**  
 GCUtilities.h, 1211  
**GenICamXMLLocation**  
 TransportLayerDevice, 1069  
**GenICamXMLLocation\_Device**  
 Spinnaker, 276  
**GenICamXMLLocation\_Host**  
 Spinnaker, 276  
**GenICamXMLLocationEnum**  
 Spinnaker, 276  
**GenICamXMLPath**  
 TransportLayerDevice, 1069  
**GenTLInfo\_QuickSpin.cpp**  
 main, 1348  
 PrintApplicationLayerDeviceInfo, 1348  
 PrintTransportLayerDeviceInfo, 1349  
 PrintTransportLayerInterfaceInfo, 1349  
 PrintTransportLayerStreamInfo, 1349  
**GenTLSFNCVersionMajor**  
 TransportLayerSystem, 1095  
**GenTLSFNCVersionMinor**  
 TransportLayerSystem, 1095  
**GenTLSFNCVersionSubMinor**  
 TransportLayerSystem, 1095  
**GenTLVersionMajor**  
 TransportLayerSystem, 1095  
**GenTLVersionMinor**  
 TransportLayerSystem, 1095  
**Get**  
 RegisterNode, 1026  
 Spinnaker::GenApi, 358  
**get**  
 BasePtr< T, B >, 404  
**GetAccessMode**  
 CameraBase, 565  
 CChunkPort, 602  
 CEventPort, 623  
 CPortImpl, 685  
 CRegisterPortImpl, 692  
 CTestPortStruct< CDataStruct >, 699  
 ICameraBase, 787  
 Node, 985  
 PortRecorder, 1017  
 PortReplay, 1021  
**GetAddress**  
 RegisterNode, 1028  
 Spinnaker::GenApi, 359  
**GetAlias**  
 Node, 986  
 Spinnaker::GenApi, 359  
**GetAuto10GDesc**

AdapterConfig, 185  
GetAutoGigabitDesc  
    AdapterConfig, 185  
GetAutoStartIp  
    AdapterConfig, 185  
GetAutoSubnetMask  
    AdapterConfig, 185  
GetAutoSubnetMaskLength  
    AdapterConfig, 185  
GetBitsPerPixel  
    IImage, 826  
    Image, 865  
GetBlackLevel  
    ChunkData, 638  
    IChunkData, 799  
GetBoxAt  
    InferenceBoundingBoxResult, 917  
GetBoxCount  
    InferenceBoundingBoxResult, 917  
GetBoxSize  
    InferenceBoundingBoxResult, 917  
GetBufferChunkData  
    IDataStream, 808  
GetBufferInfoBool8Type  
    IDataStream, 808  
GetBufferInfoPtrType  
    IDataStream, 808  
GetBufferInfoSizeType  
    IDataStream, 808  
GetBufferInfoUInt64Type  
    IDataStream, 808  
GetBufferOwnership  
    CameraBase, 566  
    ICameraBase, 787  
GetBufferSize  
    IImage, 826  
    Image, 865  
getBufSize  
    FileProtocolAdapter, 748  
GetBuildDate  
    Exception, 743  
GetBuildTime  
    Exception, 743  
GetByDeviceID  
    CameraList, 578  
    ICameraList, 795  
GetByIndex  
    CameraList, 579  
    ICameraList, 795  
    IInterfaceList, 852  
    InterfaceList, 945  
GetBySerial  
    CameraList, 579  
    ICameraList, 795  
GetCachingMode  
    Node, 986  
    Spinnaker::GenApi, 359  
GetCallbackType  
    CNodeCallback, 656  
getCameraCategory  
    GigEVisionPerformance.cpp, 1352  
GetCameras  
    IInterface, 843  
    Interface, 930  
    ISystem, 957  
    System, 1045  
GetCastAlias  
    Node, 986  
    Spinnaker::GenApi, 359  
GetCategoryName  
    LoggingEventData, 971  
GetChannelStatus  
    IImageStatistics, 839  
    ImageStatistics, 893  
GetChildren  
    Node, 986  
    Spinnaker::GenApi, 359  
GetChunkData  
    IImage, 826  
    Image, 865  
GetChunkID  
    PortNode, 1012  
GetChunkIDLength  
    CChunkPort, 602  
GetChunkLayoutId  
    IImage, 827  
    Image, 866  
GetColorProcessing  
    IImage, 827  
    Image, 866  
GetConfigLogFileName  
    AdapterConfig, 185  
GetCookie  
    CPortWriteList, 689  
    Spinnaker::GenApi, 360  
GetCounterValue  
    ChunkData, 638  
    IChunkData, 799  
GetCpuStats  
    CpuUtil, 188  
GetCRC  
    ChunkData, 638  
    IChunkData, 799  
GetCurrentEntry  
    CEnumerationTRef< EnumT >, 606  
    EnumNode, 728  
    Spinnaker::GenApi, 360  
GetData  
    IImage, 827  
    Image, 866  
GetDataAbsoluteMax  
    IImage, 827  
    Image, 867  
GetDataAbsoluteMin  
    IImage, 827  
    Image, 867

GetDefaultColorProcessing  
     Image, 867

GetDescription  
     Node, 987  
     Spinnaker::GenApi, 360

GetDeviceEventId  
     DeviceEventHandler, 706  
     IDeviceEventHandler, 820

GetDeviceEventName  
     DeviceEventHandler, 707  
     IDeviceEventHandler, 820

GetDeviceName  
     Node, 987  
     NodeMap, 997  
     Spinnaker::GenApi, 360

GetDeviceNodeMap  
     IDataStream, 809

GetDeviceSerial  
     AcquisitionMultipleCameraRecovery.cpp, 1325

GetDeviceVersion  
     NodeMap, 997  
     Spinnaker::GenApi, 360

GetDisplayName  
     Node, 987  
     Spinnaker::GenApi, 360

GetDisplayNotation  
     FloatNode, 753  
     Spinnaker::GenApi, 361

GetDisplayPrecision  
     FloatNode, 753  
     Spinnaker::GenApi, 361

GetDocuURL  
     Node, 987  
     Spinnaker::GenApi, 361

GetEncoderValue  
     ChunkData, 638  
     IChunkData, 799

GetEntries  
     EnumNode, 728  
     Spinnaker::GenApi, 361

GetEntry  
     CEnumerationTRef< EnumT >, 607  
     EnumNode, 729  
     Spinnaker::GenApi, 361

GetEntryByName  
     EnumNode, 729  
     Spinnaker::GenApi, 362

GetEnumAlias  
     CFloatPtr, 629  
     FloatNode, 753

GetEnumerationLogFileName  
     AdapterConfig, 185

GetError  
     Exception, 743

GetErrorMessage  
     Exception, 743  
     SpinUpdate.h, 1271

GetEventID  
     Node, 987  
     Spinnaker::GenApi, 362

GetEventIDLength  
     CEventPort, 623

GetEventPayloadData  
     EventHandler, 737

GetEventPayloadAxisSize  
     EventHandler, 737

GetEventType  
     EventHandler, 737

GetExposureEndLineStatusAll  
     ChunkData, 639  
     IChunkData, 799

GetExposureTime  
     ChunkData, 639  
     IChunkData, 799

GetFeatureBagHandle  
     CFeatureBag, 626

GetFeatures  
     CategoryNode, 585

GetFileName  
     Exception, 744

GetFiles  
     Spinnaker::GenICam, 387

GetFloatAlias  
     IntegerNode, 925

GetFrameID  
     ChunkData, 639  
     IChunkData, 800  
     IImage, 827  
     Image, 868

GetFullErrorMessage  
     Exception, 744

GetFunctionName  
     Exception, 744

GetGain  
     ChunkData, 639  
     IChunkData, 800

GetGenApiVersion  
     NodeMap, 997  
     Spinnaker::GenApi, 362

GetGenICamCacheFolder  
     Spinnaker::GenICam, 387

GetGenICamCLProtocolFolder  
     Spinnaker::GenICam, 388

GetGenICamLogConfig  
     Spinnaker::GenICam, 388

GetGuiXml  
     CameraBase, 566  
     ICameraBase, 787

GetHeatmapColorGradient  
     ImageUtilityHeatmap, 904

GetHeatmapRange  
     ImageUtilityHeatmap, 905

GetHeight  
     ChunkData, 640  
     IChunkData, 800  
     IImage, 828

Image, 868  
GetHistogram  
    IImageStatistics, 839  
    ImageStatistics, 893  
GetID  
    IImage, 828  
    Image, 868  
GetImage  
    ChunkData, 640  
    IChunkData, 800  
getImageCount  
    ImageEventHandlerImpl, 886  
GetImageData  
    IImage, 828  
    Image, 869  
GetImageSize  
    IImage, 828  
    Image, 869  
GetImageStatus  
    IImage, 828  
    Image, 869  
GetImageStatusDescription  
    Image, 869  
GetInc  
    FloatNode, 753  
    IntegerNode, 926  
    Spinnaker::GenApi, 362  
GetIncMode  
    FloatNode, 754  
    IntegerNode, 926  
    Spinnaker::GenApi, 362  
GetInferenceBoundingBoxResult  
    ChunkData, 640  
    IChunkData, 800  
GetInferenceConfidence  
    ChunkData, 640  
    IChunkData, 800  
GetInferenceFrameId  
    ChunkData, 641  
    IChunkData, 801  
GetInferenceResult  
    ChunkData, 641  
    IChunkData, 801  
GetInstance  
    System, 1046  
GetIntAlias  
    CFloatPtr, 629  
    FloatNode, 754  
GetInterfaceId  
    InterfaceEventHandlerImpl, 941  
GetInterfaceName  
    Pointer Class, 163  
GetInterfaces  
    ISystem, 957  
    System, 1046  
GetIntValue  
    EnumNode, 729  
    Spinnaker::GenApi, 362  
    GetLength  
        RegisterNode, 1028  
        Spinnaker::GenApi, 363  
    GetLibraryVersion  
        ISystem, 957  
        System, 1046  
    getline  
        Spinnaker::GenICam, 388  
    GetLineNumber  
        Exception, 744  
    GetLinePitch  
        ChunkData, 641  
        IChunkData, 801  
    GetLineStatusAll  
        ChunkData, 641  
        IChunkData, 801  
    GetListOfValidValues  
        FloatNode, 754  
        IntegerNode, 926  
        Spinnaker::GenApi, 363  
    GetLock  
        LockableObject< Object >, 969  
        NodeMap, 997  
        Spinnaker::GenApi, 363  
    GetLoggingEventPriorityLevel  
        ISystem, 957  
        System, 1047  
    GetLogMessage  
        LoggingEventData, 971  
    GetMax  
        FloatNode, 754  
        IntegerNode, 926  
        Spinnaker::GenApi, 363  
    getMaxImages  
        ImageEventHandlerImpl, 887  
    GetMaxIpAddress  
        AdapterConfig, 186  
    GetMaxLength  
        Spinnaker::GenApi, 363  
        StringNode, 1038  
    GetMean  
        IImageStatistics, 839  
        ImageStatistics, 894  
    GetMin  
        FloatNode, 754  
        IntegerNode, 926  
        Spinnaker::GenApi, 364  
    GetMinIpAddress  
        AdapterConfig, 186  
    GetModelName  
        NodeMap, 997  
    GetModulePathFromFunction  
        Spinnaker::GenICam, 388  
    GetName  
        Node, 987  
    GetNamespace  
        Node, 987  
        Spinnaker::GenApi, 364

GetNDC  
    LoggingEventData, 971

GetNextImage  
    CameraBase, 566  
    ICameraBase, 788  
    IDataStream, 809

GetNextImageInternal  
    IDataStream, 809

GetNode  
    CNodeCallback, 656  
    NodeMap, 997  
    Spinnaker::GenApi, 364  
    ValueNode, 1104

GetNodeHandle  
    Node, 988

GetNodeMap  
    CameraBase, 567  
    ICameraBase, 788  
    IDataStream, 809  
    Node, 988  
    Spinnaker::GenApi, 364

GetNodeMapHandle  
    NodeMap, 998

GetNodes  
    NodeMap, 998

GetNodeStatistics  
    CNodeMapFactory, 663

GetNumChannels  
    IImage, 828  
    Image, 870

GetNumDataStreams  
    CameraBase, 567  
    ICameraBase, 788

GetNumericValue  
    EnumEntryNode, 724  
    Spinnaker::GenApi, 364

GetNumImagesInUse  
    CameraBase, 568  
    ICameraBase, 788  
    IDataStream, 809

GetNumNodes  
    NodeMap, 998  
    Spinnaker::GenApi, 364

GetNumPixelValues  
    IImageStatistics, 839  
    ImageStatistics, 894

GetNumReads  
    CTestPortStruct< CDataStruct >, 699

GetNumWrites  
    CTestPortStruct< CDataStruct >, 699

GetOffsetX  
    ChunkData, 642  
    IChunkData, 801

GetOffsetY  
    ChunkData, 642  
    IChunkData, 801

GetParamStr  
    AdapterConfigException, 395

GetParents  
    Node, 988  
    Spinnaker::GenApi, 364

GetPartSelector  
    ChunkData, 642  
    IChunkData, 802

GetPayloadType  
    IImage, 829  
    Image, 870

GetPerformanceCounter  
    PerformanceCounter, 189

GetPixelDynamicRangeMax  
    ChunkData, 642  
    IChunkData, 802

GetPixelDynamicRangeMin  
    ChunkData, 643  
    IChunkData, 802

GetPixelFormat  
    IImage, 829  
    Image, 870

GetPixelFormatIntType  
    IImage, 829  
    Image, 871

GetPixelFormatName  
    IImage, 829  
    Image, 871

GetPixelValueRange  
    IImageStatistics, 840  
    ImageStatistics, 894

GetPollingTime  
    Node, 988  
    Spinnaker::GenApi, 365

GetPort  
    IDataStream, 809

GetPortHandle  
    PortNode, 1013

GetPortReplayHandle  
    PortReplay, 1022

GetPortWriteListHandle  
    CPortWriteList, 689

GetPrincipalInterfaceType  
    CChunkPort, 602  
    CEventPort, 623  
    CTestPortStruct< CDataStruct >, 699  
    Node, 988  
    Spinnaker::GenApi, 365

GetPriority  
    LoggingEventData, 971

GetPriorityName  
    LoggingEventData, 972

GetPrivateData  
    IImage, 829  
    Image, 871

GetProductGuid  
    NodeMap, 998  
    Spinnaker::GenApi, 365

GetProperty  
    Node, 989

Spinnaker::GenApi, 365  
GetPropertyNames  
    Node, 989  
        Spinnaker::GenApi, 365  
GetQuadFileNameAppendage  
    Polarization.cpp, 1383  
GetRange  
    IImageStatistics, 840  
        ImageStatistics, 895  
GetRepresentation  
    FloatNode, 754  
        IntegerNode, 926  
        Spinnaker::GenApi, 365  
GetScan3dAxisMax  
    ChunkData, 643  
        IChunkData, 802  
GetScan3dAxisMin  
    ChunkData, 643  
        IChunkData, 802  
GetScan3dCoordinateOffset  
    ChunkData, 643  
        IChunkData, 802  
GetScan3dCoordinateReferenceValue  
    ChunkData, 644  
        IChunkData, 803  
GetScan3dCoordinateScale  
    ChunkData, 644  
        IChunkData, 803  
GetScan3dInvalidHeaderValue  
    ChunkData, 644  
        IChunkData, 803  
GetScan3dTransformValue  
    ChunkData, 644  
        IChunkData, 803  
GetScanLineSelector  
    ChunkData, 645  
        IChunkData, 803  
GetSchemaVersion  
    NodeMap, 998  
        Spinnaker::GenApi, 366  
GetSecondsCounter  
    SecondsCounter, 190  
GetSelectedFeatures  
    Node, 989  
        Spinnaker::GenApi, 366  
GetSelectingFeatures  
    Node, 989  
        Spinnaker::GenApi, 366  
GetSelectorList  
    CSelectorSet, 696  
        Spinnaker::GenApi, 366  
GetSequencerSetActive  
    ChunkData, 645  
        IChunkData, 803  
GetSerialDataLength  
    ChunkData, 645  
        IChunkData, 804  
GetSize  
    CameraList, 580  
        ICameraList, 795  
        IInterfaceList, 852  
        InterfaceList, 945  
    GetStandardNameSpace  
        NodeMap, 998  
            Spinnaker::GenApi, 366  
    GetStatistics  
        IImageStatistics, 840  
            ImageStatistics, 895  
    GetStreamChannelID  
        ChunkData, 645  
            IChunkData, 804  
    GetStreamInfoBool8Type  
        IDataStream, 809  
    GetStreamInfoSizeType  
        IDataStream, 810  
    GetStreamType  
        IDataStream, 810  
    GetStride  
        IImage, 829  
            Image, 872  
    GetSubnetMaskLength  
        AdapterConfig, 186  
    GetSupportedSchemaVersions  
        CNodeMapFactory, 663  
        NodeMap, 999  
            Spinnaker::GenApi, 367  
    GetSwapEndianess  
        CChunkPort, 602  
            CEventPort, 623  
                CPortImpl, 685  
                    PortNode, 1013  
                        Spinnaker::GenApi, 367  
    GetSymbolic  
        EnumEntryNode, 725  
            Spinnaker::GenApi, 367  
    GetSymbolics  
        EnumNode, 729  
    GetThreadName  
        LoggingEventData, 972  
    GetTimerValue  
        ChunkData, 646  
            IChunkData, 804  
    GetTimeStamp  
        IImage, 830  
            Image, 872  
    GetTimestamp  
        ChunkData, 646  
            IChunkData, 804  
                LoggingEventData, 972  
    GetTimestampLatchValue  
        ChunkData, 646  
            IChunkData, 804  
    GetTLDeviceNodeMap  
        CameraBase, 568  
            ICameraBase, 788  
        GetTLNodeMap

IIInterface, 843  
 Interface, 931  
 ISystem, 957  
 System, 1047  
 GetTLPayloadType  
     IImage, 830  
     Image, 872  
 GetTLPixelFormat  
     IImage, 830  
     Image, 873  
 GetTLPixelFormatNamespace  
     IImage, 830  
     Image, 873  
 GetTLStreamNodeMap  
     CameraBase, 568  
     ICameraBase, 788  
 GetToolTip  
     Node, 989  
     NodeMap, 999  
     Spinnaker::GenApi, 367  
 GetTransferBlockID  
     ChunkData, 646  
     IChunkData, 804  
 GetTransferQueueCurrentBlockCount  
     ChunkData, 647  
     IChunkData, 805  
 GetUniqueId  
     CameraBase, 569  
     ICameraBase, 789  
 GetUnit  
     FloatNode, 755  
     IntegerNode, 927  
     Spinnaker::GenApi, 367  
 GetUserBufferCount  
     CameraBase, 569  
     ICameraBase, 789  
 GetUserBufferSize  
     CameraBase, 569  
     ICameraBase, 789  
 GetUserBufferTotalSize  
     CameraBase, 570  
     ICameraBase, 789  
 GetValidPayloadSize  
     IImage, 830  
     Image, 873  
 GetValue  
     BooleanNode, 409  
     CEnumerationTRef< EnumT >, 607  
     Counter, 678  
     EnumEntryNode, 725  
     FloatNode, 755  
     IntegerNode, 927  
     Spinnaker::GenApi, 367  
     StringNode, 1038  
 GetValueOfEnvironmentVariable  
     Spinnaker::GenICam, 389  
 GetVendorName  
     NodeMap, 999  
                 Spinnaker::GenApi, 368  
 GetVersion  
     InferenceBoundingBoxResult, 917  
 GetVersionGuid  
     NodeMap, 999  
     Spinnaker::GenApi, 368  
 GetVisibility  
     Node, 990  
     Spinnaker::GenApi, 368  
 GetWidth  
     ChunkData, 647  
     IChunkData, 805  
     IImage, 830  
     Image, 874  
 GetXOffset  
     IImage, 831  
     Image, 874  
 GetXPadding  
     IImage, 831  
     Image, 874  
 GetYOffset  
     IImage, 831  
     Image, 875  
 GetYPadding  
     IImage, 831  
     Image, 875  
 GEV  
     Spinnaker::GenApi, 352  
 GevActionDeviceKey  
     TransportLayerInterface, 1078  
 GevActionGroupKey  
     TransportLayerInterface, 1078  
 GevActionGroupMask  
     TransportLayerInterface, 1078  
 GevActionTime  
     TransportLayerInterface, 1079  
 GevActiveLinkCount  
     Camera, 516  
 GevCCP  
     Camera, 517  
     TransportLayerDevice, 1069  
 GevCCP\_ControlAccess  
     Spinnaker, 277  
 GevCCP\_EnumEntry\_GevCCP\_ControlAccess  
     Spinnaker, 276  
 GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess  
     Spinnaker, 276  
 GevCCP\_EnumEntry\_GevCCP\_OpenAccess  
     Spinnaker, 276  
 GevCCP\_ExclusiveAccess  
     Spinnaker, 277  
 GevCCP\_OpenAccess  
     Spinnaker, 277  
 GevCCPEnum  
     Spinnaker, 276  
 GevCCPEnums  
     Spinnaker, 276  
 GevCurrentDefaultGateway

Camera, 517  
GevCurrentIPAddress  
    Camera, 517  
GevCurrentIPConfigurationDHCP  
    Camera, 517  
GevCurrentIPConfigurationLLA  
    Camera, 517  
GevCurrentIPConfigurationPersistentIP  
    Camera, 517  
GevCurrentPhysicalLinkConfiguration  
    Camera, 518  
GevCurrentPhysicalLinkConfiguration\_DynamicLAG  
    Spinnaker, 277  
GevCurrentPhysicalLinkConfiguration\_MultiLink  
    Spinnaker, 277  
GevCurrentPhysicalLinkConfiguration\_SingleLink  
    Spinnaker, 277  
GevCurrentPhysicalLinkConfiguration\_StaticLAG  
    Spinnaker, 277  
GevCurrentPhysicalLinkConfigurationEnums  
    Spinnaker, 277  
GevCurrentSubnetMask  
    Camera, 518  
GevDeviceAutoForceIP  
    TransportLayerDevice, 1069  
    TransportLayerInterface, 1079  
GevDeviceDiscoverMaximumPacketSize  
    TransportLayerDevice, 1069  
GevDeviceForceGateway  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1079  
GevDeviceForceIP  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1079  
GevDeviceForceIPAddress  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1079  
GevDeviceForceSubnetMask  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1079  
GevDeviceGateway  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1080  
GevDeviceIPAddress  
    TransportLayerDevice, 1070  
    TransportLayerInterface, 1080  
GevDeviceIsWrongSubnet  
    TransportLayerDevice, 1071  
GevDeviceMACAddress  
    TransportLayerDevice, 1071  
    TransportLayerInterface, 1080  
GevDeviceMaximumPacketSize  
    TransportLayerDevice, 1071  
GevDeviceMaximumRetryCount  
    TransportLayerDevice, 1071  
GevDeviceModelsBigEndian  
    TransportLayerDevice, 1071  
GevDevicePort  
    TransportLayerDevice, 1071  
    TransportLayerInterface, 1072  
GevDeviceReadAndWriteTimeout  
    TransportLayerDevice, 1072  
GevDeviceSubnetMask  
    TransportLayerDevice, 1072  
    TransportLayerInterface, 1080  
GevDiscoveryAckDelay  
    Camera, 518  
GevFailedPacketCount  
    TransportLayerStream, 1087  
GevFirstURL  
    Camera, 518  
GevGVCPExtendedStatusCodes  
    Camera, 518  
GevGVCPExtendedStatusCodesSelector  
    Camera, 518  
GevGVCPExtendedStatusCodesSelector\_Version1\_1  
    Spinnaker, 277  
GevGVCPExtendedStatusCodesSelector\_Version2\_0  
    Spinnaker, 277  
GevGVCPExtendedStatusCodesSelectorEnums  
    Spinnaker, 277  
GevGVCPHeartbeatDisable  
    Camera, 519  
GevGVCPPendingAck  
    Camera, 519  
GevGVCPPendingTimeout  
    Camera, 519  
GevGVSPExtendedIDMode  
    Camera, 519  
GevGVSPExtendedIDMode\_Off  
    Spinnaker, 278  
GevGVSPExtendedIDMode\_On  
    Spinnaker, 278  
GevGVSPExtendedIDModeEnums  
    Spinnaker, 277  
GevHeartbeatTimeout  
    Camera, 519  
GevIEEE1588  
    Camera, 519  
GevIEEE1588ClockAccuracy  
    Camera, 520  
GevIEEE1588ClockAccuracy\_Unknown  
    Spinnaker, 278  
GevIEEE1588ClockAccuracyEnums  
    Spinnaker, 278  
GevIEEE1588Mode  
    Camera, 520  
GevIEEE1588Mode\_Auto  
    Spinnaker, 278  
GevIEEE1588Mode\_SlaveOnly  
    Spinnaker, 278  
GevIEEE1588ModeEnums  
    Spinnaker, 278  
GevIEEE1588Status  
    Camera, 520  
GevIEEE1588Status\_Disabled  
    Spinnaker, 278

GevIEEE1588Status\_Faulty  
     Spinnaker, 278

GevIEEE1588Status\_Initializing  
     Spinnaker, 278

GevIEEE1588Status\_Listening  
     Spinnaker, 279

GevIEEE1588Status\_Master  
     Spinnaker, 279

GevIEEE1588Status\_Passive  
     Spinnaker, 279

GevIEEE1588Status\_PreMaster  
     Spinnaker, 279

GevIEEE1588Status\_Slave  
     Spinnaker, 279

GevIEEE1588Status\_Uncalibrated  
     Spinnaker, 279

GevIEEE1588StatusEnums  
     Spinnaker, 278

GevInterfaceDefaultGateway  
     TransportLayerSystem, 1095

GevInterfaceDefaultIPAddress  
     TransportLayerSystem, 1096

GevInterfaceDefaultSubnetMask  
     TransportLayerSystem, 1096

GevInterfaceGateway  
     TransportLayerInterface, 1080

GevInterfaceGatewaySelector  
     TransportLayerInterface, 1080

GevInterfaceMACAddress  
     TransportLayerInterface, 1081  
     TransportLayerSystem, 1096

GevInterfaceMTU  
     TransportLayerInterface, 1081

GevInterfaceReceiveLinkSpeed  
     TransportLayerInterface, 1081

GevInterfaceSelector  
     Camera, 520

GevInterfaceSubnetIPAddress  
     TransportLayerInterface, 1081

GevInterfaceSubnetMask  
     TransportLayerInterface, 1081

GevInterfaceSubnetSelector  
     TransportLayerInterface, 1081

GevInterfaceTransmitLinkSpeed  
     TransportLayerInterface, 1082

GevIPConfigurationStatus  
     Camera, 520

GevIPConfigurationStatus\_DHCP  
     Spinnaker, 279

GevIPConfigurationStatus\_ForceIP  
     Spinnaker, 279

GevIPConfigurationStatus\_LLA  
     Spinnaker, 279

GevIPConfigurationStatus\_None  
     Spinnaker, 279

GevIPConfigurationStatus\_PersistentIP  
     Spinnaker, 279

GevIPConfigurationStatusEnums

    Spinnaker, 279

GevMACAddress  
     Camera, 520

GevMaximumNumberResendRequests  
     TransportLayerStream, 1087

GevMCDA  
     Camera, 521

GevMCPHostPort  
     Camera, 521

GevMCRC  
     Camera, 521

GevMCSP  
     Camera, 521

GevMCTT  
     Camera, 521

GevNumberOfInterfaces  
     Camera, 521

GevPacketResendMode  
     TransportLayerStream, 1087

GevPacketResendTimeout  
     TransportLayerStream, 1087

GevPAUSEFrameReception  
     Camera, 522

GevPAUSEFrameTransmission  
     Camera, 522

GevPersistentDefaultGateway  
     Camera, 522

GevPersistentIPAddress  
     Camera, 522

GevPersistentSubnetMask  
     Camera, 522

GevPhysicalLinkConfiguration  
     Camera, 522

GevPhysicalLinkConfiguration\_DynamicLAG  
     Spinnaker, 279

GevPhysicalLinkConfiguration\_MultiLink  
     Spinnaker, 279

GevPhysicalLinkConfiguration\_SingleLink  
     Spinnaker, 279

GevPhysicalLinkConfiguration\_StaticLAG  
     Spinnaker, 279

GevPhysicalLinkConfigurationEnums  
     Spinnaker, 279

GevPrimaryApplicationIPAddress  
     Camera, 523

GevPrimaryApplicationSocket  
     Camera, 523

GevPrimaryApplicationSwitchoverKey  
     Camera, 523

GevResendPacketCount  
     TransportLayerStream, 1088

GevResendRequestCount  
     TransportLayerStream, 1088

GevSCCFGAllInTransmission  
     Camera, 523

GevSCCFGExtendedChunkData  
     Camera, 523

GevSCCFGPacketResendDestination

Camera, 523  
GevSCCFGUnconditionalStreaming  
    Camera, 524  
GevSCDA  
    Camera, 524  
GevSCPDir  
    Camera, 524  
GevSCPDir  
    Camera, 524  
GevSCPHostPort  
    Camera, 524  
GevSCPIInterfaceIndex  
    Camera, 524  
GevSCPSBigEndian  
    Camera, 525  
GevSCPSDoNotFragment  
    Camera, 525  
GevSCPSFireTestPacket  
    Camera, 525  
GevSCPSPacketSize  
    Camera, 525  
GevSCSP  
    Camera, 525  
GevSCZoneConfigurationLock  
    Camera, 525  
GevSCZoneCount  
    Camera, 526  
GevSCZoneDirectionAll  
    Camera, 526  
GevSecondURL  
    Camera, 526  
GevStreamChannelSelector  
    Camera, 526  
GevSupportedOption  
    Camera, 526  
GevSupportedOptionSelector  
    Camera, 526  
GevSupportedOptionSelector\_Action  
    Spinnaker, 280  
GevSupportedOptionSelector\_CCPApplicationSocket  
    Spinnaker, 280  
GevSupportedOptionSelector\_CommandsConcatenation  
    Spinnaker, 280  
GevSupportedOptionSelector\_DiscoveryAckDelay  
    Spinnaker, 280  
GevSupportedOptionSelector\_DiscoveryAckDelayWritable  
    Spinnaker, 280  
GevSupportedOptionSelector\_Event  
    Spinnaker, 280  
GevSupportedOptionSelector\_EventData  
    Spinnaker, 280  
GevSupportedOptionSelector\_ExtendedStatusCodes  
    Spinnaker, 280  
GevSupportedOptionSelector\_HeartbeatDisable  
    Spinnaker, 280  
GevSupportedOptionSelector\_IPConfigurationDHCP  
    Spinnaker, 280  
GevSupportedOptionSelector\_IPConfigurationLLA  
    Spinnaker, 280  
GevSupportedOptionSelector\_IPConfigurationPersistentIP  
    Spinnaker, 280  
GevSupportedOptionSelector\_LinkSpeed  
    Spinnaker, 280  
GevSupportedOptionSelector\_ManifestTable  
    Spinnaker, 280  
GevSupportedOptionSelector\_MessageChannelSourceSocket  
    Spinnaker, 280  
GevSupportedOptionSelector\_PacketResend  
    Spinnaker, 280  
GevSupportedOptionSelector\_PendingAck  
    Spinnaker, 280  
GevSupportedOptionSelector\_SerialNumber  
    Spinnaker, 280  
GevSupportedOptionSelector\_StreamChannelSourceSocket  
    Spinnaker, 280  
GevSupportedOptionSelector\_TestData  
    Spinnaker, 280  
GevSupportedOptionSelector\_UserDefinedName  
    Spinnaker, 280  
GevSupportedOptionSelector\_WriteMem  
    Spinnaker, 280  
GevSupportedOptionEnums  
    Spinnaker, 279  
GevTimestampTickFrequency  
    Camera, 527  
GevTotalPacketCount  
    TransportLayerStream, 1088  
GevVersionMajor  
    TransportLayerDevice, 1072  
    TransportLayerSystem, 1096  
GevVersionMinor  
    TransportLayerDevice, 1072  
    TransportLayerSystem, 1096  
GigEVisionPerformance.cpp  
    AcquireImages, 1352  
    argBayerRG, 1354  
    argDuration, 1354  
    argMaxFrames, 1354  
    argNumImages, 1354  
    argPacketDelay, 1354  
    argPacketSize, 1354  
    argPrintUsage, 1355  
    argRelease, 1355  
    argUserSetFrames, 1355  
    cpuUsageInfo, 1355  
    EnableManualFramerate, 1352  
    getCameraCategory, 1352  
    IsRelease, 1355  
    main, 1352  
    NumImagesToGrab, 1355  
    PacketDelayToSet, 1355  
    PacketSizeToSet, 1355  
    ParseArguments, 1353  
    PixelFormatToSet, 1356  
    PrintAllNodes, 1353  
    PrintCPUUsage, 1353

PrintDataStreamInfo, 1353  
 PrintDeviceInfo, 1353  
 PrintUsage, 1353  
 RunSingleCamera, 1353  
 SetFrameRate, 1354  
 TestDuration, 1356  
 UseDuration, 1356  
 UseMaxFramerate, 1356  
 UserSetFramerate, 1356  
 globalCamList  
     AcquisitionMultipleCameraRecovery.cpp, 1326  
 GrabInfo, 771  
     GrabInfo, 771  
     imageEventHandler, 771  
     numImagesGrabbed, 771  
     numIncompleteImages, 772  
     numRemovals, 772  
 GrabNextImageByTrigger  
     BufferHandling.cpp, 1331  
     Trigger.cpp, 1393  
     Trigger\_QuickSpin.cpp, 1395  
 GrabTwoImages  
     LogicBlock.cpp, 1373  
 GREEN  
     Spinnaker, 313  
 GREY  
     Spinnaker, 313  
 GUIXMLLocation  
     TransportLayerDevice, 1072  
 GUIXMLLocation\_Device  
     Spinnaker, 280  
 GUIXMLLocation\_Host  
     Spinnaker, 280  
 GUIXMLLocationEnum  
     Spinnaker, 280  
 GuiXmlManifestAddress  
     Camera, 527  
 GUIXMLPath  
     TransportLayerDevice, 1072  
 Guru  
     Spinnaker::GenApi, 352  
 GVCP\_CHUNK\_TRAILER, 772  
     ChunkID, 772  
     ChunkLength, 772  
 GVCP\_EVENT\_ITEM, 773  
     BlockId, 773  
     EventId, 773  
     ReservedOrEventSize, 773  
     StreamChannelId, 774  
     TimestampHigh, 774  
     TimestampLow, 774  
 GVCP\_EVENT\_ITEM\_BASIC, 774  
     EventId, 774  
     ReservedOrEventSize, 775  
 GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 775  
     BlockId, 775  
     BlockId64High, 775  
     BlockId64Low, 776  
     EventId, 776  
     ReservedOrEventSize, 776  
     StreamChannelId, 776  
     TimestampHigh, 776  
     TimestampLow, 776  
 GVCP\_EVENT\_REQUEST, 777  
     Header, 777  
     Items, 777  
 GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 778  
     Header, 778  
     Items, 778  
 GVCP\_EVENTDATA\_REQUEST, 779  
     Data, 779  
     Event, 779  
     Header, 779  
 GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID, 780  
     Data, 780  
     Event, 780  
     Header, 781  
 GVCP\_MESSAGE\_TAGS  
     Spinnaker::GenApi, 354  
 GVCP\_REQUEST\_HEADER, 781  
     Command, 781  
     Flags, 781  
     Length, 781  
     Magic, 782  
     ReqId, 782  
 H264  
     SaveToAvi.cpp, 1384  
 H264Option, 782  
     bitrate, 783  
     frameRate, 783  
     H264Option, 783  
     height, 783  
     reserved, 783  
     width, 783  
 HARDWARE  
     Trigger.cpp, 1393  
     Trigger\_QuickSpin.cpp, 1395  
 HasCRC  
     CChunkAdapterDcam, 591  
     IImage, 831  
     Image, 875  
 HasInc  
     FloatNode, 755  
     Spinnaker::GenApi, 368  
 Header  
     GVCP\_EVENT\_REQUEST, 777  
     GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 778  
     GVCP\_EVENTDATA\_REQUEST, 779  
     GVCP\_EVENTDATA\_REQUEST\_EXTENDED\_ID,  
         781  
 HEATMAP\_BLACK  
     ImageUtilityHeatmap, 903  
 HEATMAP\_BLUE  
     ImageUtilityHeatmap, 903  
 HEATMAP\_CYAN  
     ImageUtilityHeatmap, 903

HEATMAP\_GREEN  
    ImageUtilityHeatmap, 903

HEATMAP\_RED  
    ImageUtilityHeatmap, 903

HEATMAP\_WHITE  
    ImageUtilityHeatmap, 903

HEATMAP\_YELLOW  
    ImageUtilityHeatmap, 903

HeatmapColor  
    ImageUtilityHeatmap, 903

Height  
    Camera, 527

height  
    H264Option, 783

HeightMax  
    Camera, 527

HexNumber  
    Spinnaker::GenApi, 351

HighDynamicRange.cpp  
    CheckNodeAccessibility, 1357  
    InitializeHDRImages, 1357  
    k\_HDRGain1, 1358  
    k\_HDRGain2, 1358  
    k\_HDRGain3, 1359  
    k\_HDRGain4, 1359  
    k\_HDRShutter1, 1359  
    k\_HDRShutter2, 1359  
    k\_HDRShutter3, 1359  
    k\_HDRShutter4, 1359  
    main, 1357  
    PrintBuildInfo, 1358  
    PrintDeviceInfo, 1358  
    RunSingleCamera, 1358  
    ToggleHDRMode, 1358

HOST\_ADDRESS\_ZERO  
    AdapterConfig, 184

HostAdapterDriverVersion  
    TransportLayerInterface, 1082

HostAdapterName  
    TransportLayerInterface, 1082

HostAdapterVendor  
    TransportLayerInterface, 1082

HQ\_LINEAR  
    Spinnaker, 246

HUE  
    Spinnaker, 313

IBase  
    Spinnaker::GenApi, 380

IBase Interface, 96

IBoolean  
    Spinnaker::GenApi, 380

IBoolean Interface, 125

ICameraBase, 784  
    ~ICameraBase, 786  
    BeginAcquisition, 786  
    CameralInternal, 792  
    DeInit, 786  
    DiscoverMaxPacketSize, 787

EndAcquisition, 787

ForceIP, 787

GetAccessMode, 787

GetBufferOwnership, 787

GetGuiXml, 787

GetNextImage, 788

GetNodeMap, 788

GetNumDataStreams, 788

GetNumImagesInUse, 788

GetTLDeviceNodeMap, 788

GetTlStreamNodeMap, 788

GetUniqueId, 789

GetUserBufferCount, 789

GetUserBufferSize, 789

GetUserBufferTotalSize, 789

ICameraBase, 786  
    Init, 789  
    InterfaceImpl, 792  
    IsInitialized, 789  
    IsStreaming, 790  
    IsValid, 790  
    m\_pCameraBaseData, 792  
    operator=, 790  
    ReadPort, 790  
    RegisterEventHandler, 790  
    SetBufferOwnership, 791  
    SetUserBuffers, 791  
    TLDevice, 792  
    TlStream, 792  
    TransportLayerDevice, 1066  
    TransportLayerStream, 1087  
    UnregisterEventHandler, 791  
    WritePort, 791

ICameraList, 793  
    ~ICameraList, 794  
    Append, 794  
    CameraListImpl, 796  
    Clear, 794  
    GetByDeviceID, 795  
    GetByIndex, 795  
    GetBySerial, 795  
    GetSize, 795  
    ICameraList, 794  
    InterfaceImpl, 796  
    m\_pCameraListData, 797  
    operator=, 795  
    operator[], 795  
    RemoveByDeviceID, 796  
    RemoveByIndex, 796  
    RemoveBySerial, 796

ICategory  
    Spinnaker::GenApi, 380

ICategory Interfaces, 126

IChunkData, 797  
    ~IChunkData, 798  
    GetBlackLevel, 799  
    GetCounterValue, 799  
    GetCRC, 799

GetEncoderValue, 799  
 GetExposureEndLineStatusAll, 799  
 GetExposureTime, 799  
 GetFrameID, 800  
 GetGain, 800  
 GetHeight, 800  
 GetImage, 800  
 GetInferenceBoundingBoxResult, 800  
 GetInferenceConfidence, 800  
 GetInferenceFrameId, 801  
 GetInferenceResult, 801  
 GetLinePitch, 801  
 GetLineStatusAll, 801  
 GetOffsetX, 801  
 GetOffsetY, 801  
 GetPartSelector, 802  
 GetPixelDynamicRangeMax, 802  
 GetPixelDynamicRangeMin, 802  
 GetScan3dAxisMax, 802  
 GetScan3dAxisMin, 802  
 GetScan3dCoordinateOffset, 802  
 GetScan3dCoordinateReferenceValue, 803  
 GetScan3dCoordinateScale, 803  
 GetScan3dInvalidDataValue, 803  
 GetScan3dTParameterValue, 803  
 GetScanLineSelector, 803  
 GetSequencerSetActive, 803  
 GetSerialDataLength, 804  
 GetStreamChannelID, 804  
 GetTimerValue, 804  
 GetTimestamp, 804  
 GetTimestampLatchValue, 804  
 GetTransferBlockID, 804  
 GetTransferQueueCurrentBlockCount, 805  
 GetWidth, 805  
 IChunkData, 798  
 SetChunks, 805  
 IChunkData Class, 87  
 IChunkPort  
     Spinnaker::GenApi, 380  
 IChunkPort Interface, 127  
 IChunkPort.h  
     CHUNK\_BASE\_ADDRESS\_REGISTER, 1215  
     CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN, 1215  
     CHUNK\_LENGTH\_REGISTER, 1215  
     CHUNK\_LENGTH\_REGISTER\_LEN, 1215  
 ICommand  
     Spinnaker::GenApi, 380  
 ICommand Interface, 128  
 IDataStream, 806  
     ~IDataStream, 806  
     AnnounceImage, 807  
     AttachBuffer, 807  
     CleanupChunkAdapter, 807  
     EventHandler, 738  
     FlushQueueAllDiscard, 808  
     GetBufferChunkData, 808  
     GetBufferInfoBool8Type, 808  
     GetBufferInfoPtrType, 808  
     GetBufferInfoSizeType, 808  
     GetBufferInfoUInt64Type, 808  
     GetDeviceNodeMap, 809  
     GetNextImage, 809  
     GetNextImageInternal, 809  
     GetNodeMap, 809  
     GetNumImagesInUse, 809  
     GetPort, 809  
     GetStreamInfoBool8Type, 809  
     GetStreamInfoSizeType, 810  
     GetStreamType, 810  
     IDataStream, 807  
     Image, 881  
     InitChunkAdapter, 810  
     IsCRCCheckEnabled, 810  
     IsImageInUse, 810  
     IsStreaming, 810  
     KillBufferEvent, 810  
     RegisterImageEventHandler, 811  
     ReleaseImage, 811  
     RevokelImages, 811  
     StartStream, 811  
     StopStream, 811  
     TransportLayerStreamInfo, 811  
     UnregisterImageEventHandler, 811  
     WaitOnImageEvent, 812  
 IDestroy  
     Spinnaker::GenApi, 380  
 IDestroy Interface, 129  
 IDevFileStream  
     Spinnaker::GenApi, 346  
 IDevFileStreamBase< CharType, Traits >, 812  
     close, 813  
     filebuf\_type, 813  
     ios\_type, 813  
     is\_open, 813  
     istream\_type, 813  
     open, 814  
     rdbuf, 814  
 IDevFileStreamBuf  
     IDevFileStreamBuf< CharType, Traits >, 815  
 IDevFileStreamBuf< CharType, Traits >, 814  
     ~IDevFileStreamBuf, 815  
     close, 816  
     IDevFileStreamBuf, 815  
     is\_open, 816  
     open, 816  
     pbackfail, 816  
     underflow, 816  
 IDeviceArrivalEventHandler, 817  
     ~IDeviceArrivalEventHandler, 818  
     IDeviceArrivalEventHandler, 818  
     OnDeviceArrival, 818  
     operator=, 818  
 IDeviceEventHandler, 819  
     ~IDeviceEventHandler, 820

GetDeviceEventId, 820  
GetDeviceEventName, 820  
IDeviceEventHandler, 820  
OnDeviceEvent, 821  
operator=, 821  
IDeviceInfo  
    Spinnaker::GenApi, 381  
IDeviceInfo Interface, 130  
IDeviceRemovalEventHandler, 821  
    ~IDeviceRemovalEventHandler, 822  
    IDeviceRemovalEventHandler, 822  
    OnDeviceRemoval, 823  
    operator=, 823  
idFrom  
    Spinnaker::GenApi, 349  
idNone  
    Spinnaker::GenApi, 349  
idTo  
    Spinnaker::GenApi, 349  
IEnumEntry  
    Spinnaker::GenApi, 381  
IEnumEntry Interface, 131  
IEnumeration  
    Spinnaker::GenApi, 381  
IEnumeration Interface, 132  
IEnumerationT  
    Spinnaker::GenApi, 381  
IEnumerationT Interface, 133  
IEnumReference  
    Spinnaker::GenApi, 381  
IFloat  
    Spinnaker::GenApi, 382  
IFloat Interface, 134  
IIDC  
    Spinnaker::GenApi, 352  
IIImage, 823  
    ~IIImage, 825  
    CalculateStatistics, 825  
    CheckCRC, 825  
    Convert, 825, 826  
    DeepCopy, 826  
    GetBitsPerPixel, 826  
    GetBufferSize, 826  
    GetChunkData, 826  
    GetChunkLayoutId, 827  
    GetColorProcessing, 827  
    GetData, 827  
    GetDataAbsoluteMax, 827  
    GetDataAbsoluteMin, 827  
    GetFrameID, 827  
    GetHeight, 828  
    GetID, 828  
    GetImageData, 828  
    GetImageSize, 828  
    GetImageStatus, 828  
    GetNumChannels, 828  
    GetPayloadType, 829  
    GetPixelFormat, 829  
    GetPixelFormatIntType, 829  
    GetPixelFormatName, 829  
    GetPrivateData, 829  
    GetStride, 829  
    GetTimeStamp, 830  
    GetTLPayloadType, 830  
    GetTLPixelFormat, 830  
    GetTLPixelFormatNamespace, 830  
    GetValidPayloadSize, 830  
    GetWidth, 830  
    GetXOffset, 831  
    GetXPadding, 831  
    GetYOffset, 831  
    GetYPadding, 831  
    HasCRC, 831  
    IIImage, 825  
    IsIncomplete, 831  
    IsInUse, 832  
    Release, 832  
    ResetImage, 832  
    Save, 832–834  
    Stream, 834  
IIImage Class, 88  
IIImageEventHandler, 835  
    ~IIImageEventHandler, 836  
    IIImageEventHandler, 836  
    OnImageEvent, 836  
    operator=, 836  
IIImageStatistics, 837  
    ~IIImageStatistics, 838  
    DisableAll, 838  
    EnableAll, 838  
    EnableGreyOnly, 838  
    EnableHSOnly, 838  
    EnableRGBOnly, 839  
    GetChannelStatus, 839  
    GetHistogram, 839  
    GetMean, 839  
    GetNumPixelValues, 839  
    GetPixelValueRange, 840  
    GetRange, 840  
    GetStatistics, 840  
    IIImageStatistics, 838  
    SetChannelStatus, 840  
IIImageStatistics Class, 89  
IIInteger  
    Spinnaker::GenApi, 382  
IIInteger Interface, 135  
IIInterface, 841  
    ~IIInterface, 842  
    GetCameras, 843  
    GetTLNodeMap, 843  
    IIInterface, 842, 843  
    InterfaceInternal, 844  
    IsInUse, 843  
    IsValid, 843  
    m\_pInterfaceData, 845  
    operator=, 843

RegisterEventHandler, 844  
 SendActionCommand, 844  
 SystemImpl, 845  
 TLInterface, 845  
 TransportLayerInterface, 1076  
 UnregisterEventHandler, 844  
 UpdateCameras, 844  
**IInterface Class**, 90  
**IInterfaceArrivalEventHandler**, 846  
 ~IInterfaceArrivalEventHandler, 847  
**IInterfaceArrivalEventHandler**, 847  
 OnInterfaceArrival, 847  
 operator=, 847  
**IInterfaceEventHandler**, 848  
 ~IInterfaceEventHandler, 849  
**IInterfaceEventHandler**, 849  
 OnDeviceArrival, 850  
 OnDeviceRemoval, 850  
 operator=, 850  
**IInterfaceList**, 851  
 ~IInterfaceList, 851  
 Clear, 852  
 GetByIndex, 852  
 GetSize, 852  
**IInterfaceList**, 852  
 m\_pInterfaceListData, 853  
 operator=, 852  
 operator[], 853  
**IInterfaceList Class**, 91  
**IInterfaceRemovalEventHandler**, 853  
 ~IInterfaceRemovalEventHandler, 854  
**IInterfaceRemovalEventHandler**, 854  
 OnInterfaceRemoval, 855  
 operator=, 855  
**ILoggingEventHandler**, 855  
 ~ILoggingEventHandler, 856  
**ILoggingEventHandler**, 856  
 OnLogEvent, 857  
 operator=, 857  
**IMAGE**  
 ChunkData.cpp, 1332  
**Image**, 857  
 ~Image, 861  
 CalculateStatistics, 862  
 CheckCRC, 862  
 Convert, 862, 863  
 Create, 863, 864  
 CreateShared, 864  
 DeepCopy, 864  
 GetBitsPerPixel, 865  
 GetBufferSize, 865  
 GetChunkData, 865  
 GetChunkLayoutId, 866  
 GetColorProcessing, 866  
 GetData, 866  
 GetDataAbsoluteMax, 867  
 GetDataAbsoluteMin, 867  
 GetDefaultColorProcessing, 867  
 GetFrameID, 868  
 GetHeight, 868  
 GetID, 868  
 GetImageData, 869  
 GetImageSize, 869  
 GetImageStatus, 869  
 GetImageStatusDescription, 869  
 GetNumChannels, 870  
 GetPayloadType, 870  
 GetPixelFormat, 870  
 GetPixelFormatIntType, 871  
 GetPixelFormatName, 871  
 GetPrivateData, 871  
 GetStride, 872  
 GetTimeStamp, 872  
 GetTLPayloadType, 872  
 GetTLPixelFormat, 873  
 GetTLPixelFormatNamespace, 873  
 GetValidPayloadSize, 873  
 GetWidth, 874  
 GetXOffset, 874  
 GetXPadding, 874  
 GetYOffset, 875  
 GetYPadding, 875  
 HasCRC, 875  
 IDataStream, 881  
 Image, 861  
 ImageConverter, 881  
 ImageFiler, 881  
 ImageStatsCalculator, 881  
 ImageUtilityImpl, 881  
 ImageUtilityPolarizationImpl, 881  
 IsCompressed, 876  
 IsIncomplete, 876  
 IsInUse, 876  
 Release, 876  
 ResetImage, 877  
 Save, 878–880  
 SetDefaultColorProcessing, 880  
 Stream, 882  
**Image Class**, 55  
**Image Utility Class**, 59  
**Image Utility Heatmap Class**, 60  
**Image Utility Polarization Class**, 61  
**IMAGE\_CHUNK\_DATA\_INVALID**  
 Spinnaker, 283  
**IMAGE\_CRC\_CHECK\_FAILED**  
 Spinnaker, 283  
**IMAGE\_DATA\_INCOMPLETE**  
 Spinnaker, 283  
**IMAGE\_DATA\_OVERFLOW**  
 Spinnaker, 283  
**IMAGE\_DATA\_RANGE**  
 ImageUtility, 898  
**IMAGE\_FILE\_FORMAT\_FORCE\_32BITS**  
 Spinnaker, 283  
**IMAGE\_INFO\_INCONSISTENT**  
 Spinnaker, 283

IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT  
    Spinnaker, 283

IMAGE\_MIN\_ABSOLUTE\_MAX  
    ImageUtility, 898

IMAGE\_MISSING\_LEADER  
    Spinnaker, 283

IMAGE\_MISSING\_PACKETS  
    Spinnaker, 283

IMAGE\_MISSING\_TRAILER  
    Spinnaker, 283

IMAGE\_NO\_ERROR  
    Spinnaker, 283

IMAGE\_NO\_SYSTEM\_RESOURCES  
    Spinnaker, 283

IMAGE\_PACKETID\_INCONSISTENT  
    Spinnaker, 283

IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT  
    Spinnaker, 283

IMAGE\_UNKNOWN\_ERROR  
    Spinnaker, 283

ImageComponentEnable  
    Camera, 527

ImageComponentSelector  
    Camera, 528

ImageComponentSelector\_Color  
    Spinnaker, 281

ImageComponentSelector\_Confidence  
    Spinnaker, 281

ImageComponentSelector\_Disparity  
    Spinnaker, 281

ImageComponentSelector\_Infrared  
    Spinnaker, 281

ImageComponentSelector\_Intensity  
    Spinnaker, 281

ImageComponentSelector\_Range  
    Spinnaker, 281

ImageComponentSelector\_Scatter  
    Spinnaker, 281

ImageComponentSelector\_Ultraviolet  
    Spinnaker, 281

ImageComponentSelectorEnums  
    Spinnaker, 280

ImageCompressionBitrate  
    Camera, 528

ImageCompressionJPEGFormatOption  
    Camera, 528

ImageCompressionJPEGFormatOption\_BaselineOptimized  
    Spinnaker, 281

ImageCompressionJPEGFormatOption\_BaselineStandard  
    Spinnaker, 281

ImageCompressionJPEGFormatOption\_Lossless  
    Spinnaker, 281

ImageCompressionJPEGFormatOption\_Progressive  
    Spinnaker, 281

ImageCompressionJPEGFormatOptionEnums  
    Spinnaker, 281

ImageCompressionMode  
    Camera, 528

ImageCompressionMode\_Lossless  
    Spinnaker, 282

ImageCompressionMode\_Off  
    Spinnaker, 282

ImageCompressionModeEnums  
    Spinnaker, 282

ImageCompressionQuality  
    Camera, 528

ImageCompressionRateOption  
    Camera, 528

ImageCompressionRateOption\_FixBitrate  
    Spinnaker, 282

ImageCompressionRateOption\_FixQuality  
    Spinnaker, 282

ImageCompressionRateOptionEnums  
    Spinnaker, 282

ImageConverter  
    Image, 881

ImageEventHandler, 882

    ~ImageEventHandler, 884

    ImageEventHandler, 883

    OnImageEvent, 884

    operator=, 884

imageEventHandler  
    GrabInfo, 771

ImageEventHandler Class, 56

ImageEventHandlerImpl, 885

    ~ImageEventHandlerImpl, 886

    getImageCount, 886

    getMaxImages, 887

    ImageEventHandlerImpl, 886

    OnImageEvent, 887

ImageEvents.cpp

    AcquireImages, 1360

    ConfigureImageEvents, 1360

    main, 1360

    PrintDeviceInfo, 1361

    ResetImageEvents, 1361

    RunSingleCamera, 1361

    SleepyWrapper, 1361

    WaitForImages, 1361

ImageFileFormat  
    Spinnaker, 282

ImageFiler  
    Image, 881

ImageFormatControl.cpp

    AcquireImages, 1362

    ConfigureCustomImageSettings, 1362

    main, 1362

    PrintDeviceInfo, 1362

    RunSingleCamera, 1363

ImageFormatControl\_QuickSpin.cpp

    AcquireImages, 1363

    ConfigureCustomImageSettings, 1363

    main, 1364

    PrintDeviceInfo, 1364

    RunSingleCamera, 1364

ImagePtr, 888

~ImagePtr, 889  
 ImagePtr, 889  
 operator=, 890  
**ImagePtr Class**, 57  
**ImageScalingAlgorithm**  
 ImageUtility, 898  
**ImageStatistics**, 890  
 ~ImageStatistics, 892  
 DisableAll, 892  
 EnableAll, 892  
 EnableGreyOnly, 892  
 EnableHSLOnly, 893  
 EnableRGBOnly, 893  
 GetChannelStatus, 893  
 GetHistogram, 893  
 GetMean, 894  
 GetNumPixelValues, 894  
 GetPixelValueRange, 894  
 GetRange, 895  
 GetStatistics, 895  
 ImageStatistics, 892  
 ImageStatsCalculator, 897  
 operator=, 896  
 SetChannelStatus, 896  
**ImageStatistics Class**, 58  
**ImageStatsCalculator**  
 Image, 881  
 ImageStatistics, 897  
**ImageStatus**  
 Spinnaker, 283  
**ImageUtility**, 897  
 ABSOLUTE\_DATA\_RANGE, 898  
 ABSOLUTE\_MIN\_IMAGE\_MAX, 898  
 CreateNormalized, 899–901  
 CreateScaled, 901  
 IMAGE\_DATA\_RANGE, 898  
 IMAGE\_MIN\_ABSOLUTE\_MAX, 898  
 ImageScalingAlgorithm, 898  
 NEAREST\_NEIGHBOR, 898  
 SourceDataRange, 898  
**ImageUtilityHeatmap**, 902  
 CreateHeatmap, 903, 904  
 GetHeatmapColorGradient, 904  
 GetHeatmapRange, 905  
 HEATMAP\_BLACK, 903  
 HEATMAP\_BLUE, 903  
 HEATMAP\_CYAN, 903  
 HEATMAP\_GREEN, 903  
 HEATMAP\_RED, 903  
 HEATMAP\_WHITE, 903  
 HEATMAP\_YELLOW, 903  
 HeatmapColor, 903  
 SetHeatmapColorGradient, 905  
 SetHeatmapRange, 905  
**ImageUtilityImpl**  
 Image, 881  
**ImageUtilityPolarization**, 906  
 CreateAolp, 907, 908  
 CreateDolp, 908, 909  
 CreateGlareReduced, 909, 910  
 CreateStokesS0, 910  
 CreateStokesS1, 911  
 CreateStokesS2, 912  
 ExtractPolarQuadrant, 913  
 PolarizationQuadrant, 907  
 QUADRANT\_I0, 907  
 QUADRANT\_I135, 907  
 QUADRANT\_I45, 907  
 QUADRANT\_I90, 907  
**ImageUtilityPolarizationImpl**  
 Image, 881  
**ImposeAccessMode**  
 Node, 990  
 Spinnaker::GenApi, 368  
**ImposeMax**  
 FloatNode, 755  
 IntegerNode, 927  
 Spinnaker::GenApi, 368, 369  
**ImposeMin**  
 FloatNode, 755  
 IntegerNode, 927  
 Spinnaker::GenApi, 369  
**ImposeVisibility**  
 Node, 990  
 Spinnaker::GenApi, 369  
**include/AdapterConfig.h**, 1107  
**include/AVIRecorder.h**, 1109  
**include/BasePtr.h**, 1110  
**include/Camera.h**, 1111  
**include/CameraBase.h**, 1111  
**include/CameraDefs.h**, 1112  
**include/CameraList.h**, 1144  
**include/CameraPtr.h**, 1145  
**include/ChunkData.h**, 1145  
**include/ChunkDataInference.h**, 1146  
**include/DeviceArrivalEventHandler.h**, 1147  
**include/DeviceEventHandler.h**, 1148  
**include/DeviceRemovalEventHandler.h**, 1149  
**include/EventHandler.h**, 1150  
**include/Exception.h**, 1150  
**include/Image.h**, 1151  
**include/ImageEventHandler.h**, 1152  
**include/ImagePtr.h**, 1152  
**include/ImageStatistics.h**, 1153  
**include/ImageUtility.h**, 1154  
**include/ImageUtilityHeatmap.h**, 1155  
**include/ImageUtilityPolarization.h**, 1156  
**include/Interface.h**, 1157  
**include/Interface/ICameraBase.h**, 1157  
**include/Interface/ICameraList.h**, 1158  
**include/Interface/IChunkData.h**, 1159  
**include/Interface/IDeviceArrivalEventHandler.h**, 1160  
**include/Interface/IDeviceEventHandler.h**, 1161  
**include/Interface/IDeviceRemovalEventHandler.h**, 1162  
**include/Interface/IImage.h**, 1163  
**include/Interface/IImageEventHandler.h**, 1163

include/Interface/IImageStatistics.h, 1164  
include/Interface/IInterface.h, 1165  
include/Interface/IInterfaceArrivalEventHandler.h, 1166  
include/Interface/IInterfaceEventHandler.h, 1167  
include/Interface/IInterfaceList.h, 1167  
include/Interface/IInterfaceRemovalEventHandler.h,  
    1168  
include/Interface/ILoggingEventHandler.h, 1169  
include/Interface/IStream.h, 1169  
include/Interface/ISystem.h, 1170  
include/Interface/ISystemEventHandler.h, 1171  
include/InterfaceArrivalEventHandler.h, 1172  
include/InterfaceEventHandler.h, 1173  
include/InterfaceList.h, 1174  
include/InterfacePtr.h, 1174  
include/InterfaceRemovalEventHandler.h, 1175  
include/LoggingEventData.h, 1176  
include/LoggingEventDataPtr.h, 1176  
include/LoggingEventHandler.h, 1177  
include/SpinGenApi/Autovector.h, 1178  
include/SpinGenApi/Base.h, 1179  
include/SpinGenApi/BooleanNode.h, 1180  
include/SpinGenApi/CategoryNode.h, 1181  
include/SpinGenApi/ChunkAdapter.h, 1182  
include/SpinGenApi/ChunkAdapterDcam.h, 1183  
include/SpinGenApi/ChunkAdapterGeneric.h, 1184  
include/SpinGenApi/ChunkAdapterGEV.h, 1185  
include/SpinGenApi/ChunkAdapterU3V.h, 1186  
include/SpinGenApi/ChunkPort.h, 1187  
include/SpinGenApi/CommandNode.h, 1187  
include/SpinGenApi/Compatibility.h, 1188  
include/SpinGenApi/Container.h, 1189  
include/SpinGenApi/Counter.h, 1189  
include/SpinGenApi/EnumClasses.h, 1190  
include/SpinGenApi/EnumEntryNode.h, 1191  
include/SpinGenApi/EnumNode.h, 1192  
include/SpinGenApi/EnumNodeT.h, 1193  
include/SpinGenApi/EventAdapter.h, 1193  
include/SpinGenApi/EventAdapter1394.h, 1194  
include/SpinGenApi/EventAdapterGeneric.h, 1195  
include/SpinGenApi/EventAdapterGEV.h, 1195  
include/SpinGenApi/EventAdapterU3V.h, 1196  
include/SpinGenApi/EventPort.h, 1197  
include/SpinGenApi/Filestream.h, 1198  
include/SpinGenApi/FloatNode.h, 1199  
include/SpinGenApi/FloatRegNode.h, 1200  
include/SpinGenApi/GCBase.h, 1201  
include/SpinGenApi/GCString.h, 1201  
include/SpinGenApi/GCStringVector.h, 1203  
include/SpinGenApi/GCSynch.h, 1204  
include/SpinGenApi/GCTypes.h, 1205  
include/SpinGenApi/GCUtilities.h, 1208  
include/SpinGenApi/IBoolean.h, 1212  
include/SpinGenApi/ICategory.h, 1213  
include/SpinGenApi/IChunkPort.h, 1214  
include/SpinGenApi/ICommand.h, 1216  
include/SpinGenApi/IDestroy.h, 1217  
include/SpinGenApi/IDeviceInfo.h, 1218  
include/SpinGenApi/IEnumEntry.h, 1219  
include/SpinGenApi/IEnumeration.h, 1220  
include/SpinGenApi/IEnumerationT.h, 1221  
include/SpinGenApi/IFloat.h, 1222  
include/SpinGenApi/IInteger.h, 1224  
include/SpinGenApi/INode.h, 1225  
include/SpinGenApi/INodeMap.h, 1228  
include/SpinGenApi/INodeMapDyn.h, 1230  
include/SpinGenApi/IntegerNode.h, 1231  
include/SpinGenApi/IntRegNode.h, 1232  
include/SpinGenApi/IPort.h, 1233  
include/SpinGenApi/IPortConstruct.h, 1234  
include/SpinGenApi/IPortRecorder.h, 1235  
include/SpinGenApi/IRegister.h, 1236  
include/SpinGenApi/ISelector.h, 1237  
include/SpinGenApi/ISelectorDigit.h, 1238  
include/SpinGenApi/IString.h, 1239  
include/SpinGenApi/IValue.h, 1240  
include/SpinGenApi/Node.h, 1241  
include/SpinGenApi/NodeCallback.h, 1242  
include/SpinGenApi/NodeCallbackImpl.h, 1244  
include/SpinGenApi/NodeMap.h, 1244  
include/SpinGenApi/NodeMapFactory.h, 1245  
include/SpinGenApi/NodeMapRef.h, 1246  
include/SpinGenApi/Persistence.h, 1247  
include/SpinGenApi/Pointer.h, 1248  
include/SpinGenApi/PortImpl.h, 1250  
include/SpinGenApi/PortNode.h, 1251  
include/SpinGenApi/PortRecorder.h, 1252  
include/SpinGenApi/PortReplay.h, 1252  
include/SpinGenApi/PortWriteList.h, 1253  
include/SpinGenApi/Reference.h, 1254  
include/SpinGenApi/RegisterNode.h, 1255  
include/SpinGenApi/RegisterPortImpl.h, 1256  
include/SpinGenApi/SelectorSet.h, 1256  
include/SpinGenApi/SpinnakerGenApi.h, 1257  
include/SpinGenApi/SpinTestCamera.h, 1257  
include/SpinGenApi/StringNode.h, 1258  
include/SpinGenApi/StringRegNode.h, 1259  
include/SpinGenApi/StructPort.h, 1259  
include/SpinGenApi/Synch.h, 1260  
include/SpinGenApi/Types.h, 1261  
include/SpinGenApi/ValueNode.h, 1264  
include/Spinnaker.h, 1265  
include/SpinnakerDefs.h, 1265  
include/SpinnakerPlatform.h, 1269  
include/SpinUpdate.h, 1270  
include/SpinVideo.h, 1273  
include/SpinVideoDefs.h, 1274  
include/System.h, 1275  
include/SystemEventHandler.h, 1276  
include/SystemPtr.h, 1277  
include/TransportLayerDefs.h, 1277  
include/TransportLayerDevice.h, 1279  
include/TransportLayerInterface.h, 1280  
include/TransportLayerStream.h, 1280  
include/TransportLayerSystem.h, 1281  
IncompatibleDeviceCount

TransportLayerInterface, 1082  
 IncompatibleDeviceID  
     TransportLayerInterface, 1082  
 IncompatibleDeviceModelName  
     TransportLayerInterface, 1083  
 IncompatibleDeviceSelector  
     TransportLayerInterface, 1083  
 IncompatibleDeviceVendorName  
     TransportLayerInterface, 1083  
 IncompatibleGevDeviceIPAddress  
     TransportLayerInterface, 1083  
 IncompatibleGevDeviceMACAddress  
     TransportLayerInterface, 1083  
 IncompatibleGevDeviceSubnetMask  
     TransportLayerInterface, 1083  
 Increasing  
     Spinnaker::GenApi, 352  
 Indent  
     NodeMapInfo.cpp, 1378  
 indexedColor\_8bit  
     BMPOption, 407  
 INDIVIDUAL  
     NodeMapInfo.cpp, 1378  
 Inference.cpp  
     AcquireImages, 1366  
     arrayLabelClassification, 1369  
     arrayLabelDetection, 1369  
     CameraCloseFile, 1366  
     CameraDeleteFile, 1366  
     CameraOpenFile, 1366  
     CameraWriteToFile, 1366  
     chosenFileUploadPersistence, 1370  
     chosenInferenceNetworkType, 1370  
     CLASSIFICATION, 1366  
     ConfigureChunkData, 1367  
     ConfigureInference, 1367  
     ConfigureTestPattern, 1367  
     ConfigureTrigger, 1367  
     DDR, 1365  
     DeleteFileOnCamera, 1367  
     DETECTION, 1366  
     DisableChunkData, 1367  
     DisableTrigger, 1368  
     DisplayChunkData, 1368  
     FileUploadPersistence, 1365  
     FLASH, 1365  
     InferenceNetworkType, 1365  
     injectedImageFilePath, 1370  
     injectedImageHeight, 1370  
     injectedImageWidth, 1370  
     labelClassification, 1368  
     labelDetection, 1368  
     LoadFileIntoMemory, 1368  
     main, 1368  
     networkFilePath, 1370  
     PrintDeviceInfo, 1369  
     RunSingleCamera, 1369  
     SetChunkEnable, 1369  
 UploadFileToCamera, 1369  
 INFERENCE\_BOX\_TYPE\_CIRCLE  
     Spinnaker, 284  
 INFERENCE\_BOX\_TYPE\_RECTANGLE  
     Spinnaker, 284  
 INFERENCE\_BOX\_TYPE\_ROTATED\_RECTANGLE  
     Spinnaker, 284  
 InferenceBoundingBox, 913  
     boxType, 914  
     circle, 914  
     classId, 914  
     confidence, 914  
     rect, 915  
     rotatedRect, 915  
 InferenceBoundingBoxResult, 915  
     ~InferenceBoundingBoxResult, 916  
     GetBoxAt, 917  
     GetBoxCount, 917  
     GetBoxSize, 917  
     GetVersion, 917  
     InferenceBoundingBoxResult, 916  
     operator=, 917  
 InferenceBoxCircle, 918  
     centerXCoord, 918  
     centerYCoord, 918  
     radius, 918  
 InferenceBoxRect, 918  
     bottomRightXCoord, 919  
     bottomRightYCoord, 919  
     topLeftXCoord, 919  
     topLeftYCoord, 919  
 InferenceBoxRotatedRect, 919  
     bottomRightXCoord, 920  
     bottomRightYCoord, 920  
     rotationAngle, 920  
     topLeftXCoord, 920  
     topLeftYCoord, 920  
 InferenceBoxType  
     Spinnaker, 283  
 InferenceNetworkType  
     Inference.cpp, 1365  
 Init  
     Camera, 437  
     CameraBase, 570  
     ICameraBase, 789  
 InitChunkAdapter  
     IDataStream, 810  
 InitializeHDRImages  
     HighDynamicRange.cpp, 1357  
 InitializeSystem  
     FileAccess\_QuickSpin.cpp, 1346  
 injectedImageFilePath  
     Inference.cpp, 1370  
 injectedImageHeight  
     Inference.cpp, 1370  
 injectedImageWidth  
     Inference.cpp, 1370  
 INode

Spinnaker::GenApi, 382  
INode Interface, 136  
Combine, 136, 137  
IsAvailable, 137  
IsCacheable, 138  
IsImplemented, 138  
IsReadable, 138, 139  
IsVisible, 139  
IsWritable, 139  
INodeMap  
  Spinnaker::GenApi, 382  
INodeMap Interface, 140  
INodeMapDyn  
  Spinnaker::GenApi, 382  
INodeMapDyn Interface, 141  
int64\_autovector\_t, 920  
  \_pCount, 923  
  \_pv, 923  
  ~int64\_autovector\_t, 921  
int64\_autovector\_t, 921  
operator delete, 922  
operator new, 922  
operator=, 922  
operator[], 922  
size, 922  
IntegerNode, 923  
  ~IntegerNode, 925  
  GetFloatAlias, 925  
  GetInc, 926  
  GetIncMode, 926  
  GetListOfValidValues, 926  
  GetMax, 926  
  GetMin, 926  
  GetRepresentation, 926  
  GetUnit, 927  
  GetValue, 927  
  ImposeMax, 927  
  ImposeMin, 927  
  IntegerNode, 925  
  operator\*, 928  
  operator(), 927  
  operator=, 928  
  SetReference, 928  
  SetValue, 928  
IntegerNode Class, 142  
INTEGRAL\_CAST  
  Spinnaker::GenICam, 389  
INTEGRAL\_CAST2  
  Spinnaker::GenICam, 389  
Interface, 929  
  ~Interface, 930  
  GetCameras, 930  
  GetTLNodeMap, 931  
  InterfaceInternal, 934  
  IsInUse, 931  
  IsValid, 931  
  RegisterEventHandler, 932  
  SendActionCommand, 932  
TransportLayerInterface, 1076  
UnregisterEventHandler, 933  
UpdateCameras, 933  
interface  
  Types.h, 1264  
Interface Class, 62  
InterfaceArrivalEventHandler, 934  
  ~InterfaceArrivalEventHandler, 935  
  InterfaceArrivalEventHandler, 935  
  OnInterfaceArrival, 935  
  operator=, 936  
InterfaceArrivalEventHandler Class, 63  
InterfaceDisplayName  
  TransportLayerInterface, 1084  
  TransportLayerSystem, 1096  
InterfaceEventHandler, 936  
  ~InterfaceEventHandler, 938  
  InterfaceEventHandler, 937  
  OnDeviceArrival, 938  
  OnDeviceRemoval, 938  
  operator=, 938  
InterfaceEventHandler Class, 64  
InterfaceEventHandlerImpl, 939  
  ~InterfaceEventHandlerImpl, 941  
  GetInterfaceId, 941  
  InterfaceEventHandlerImpl, 940, 941  
  OnDeviceArrival, 941, 942  
  OnDeviceRemoval, 942  
  PrintGenericHandlerMessage, 943  
InterfaceID  
  TransportLayerInterface, 1084  
  TransportLayerSystem, 1097  
InterfaceImpl  
  CameraBase, 575  
  ICameraBase, 792  
  ICameraList, 796  
InterfaceInternal  
  IInterface, 844  
  Interface, 934  
  TransportLayerInterface, 1076  
InterfaceList, 943  
  ~InterfaceList, 944  
  Clear, 945  
  GetByIndex, 945  
  GetSize, 945  
  InterfaceList, 944  
  operator=, 946  
  operator[], 946  
  SystemImpl, 946  
InterfaceList Class, 65  
InterfacePtr, 947  
  InterfacePtr, 948  
InterfacePtr Class, 66  
InterfaceRemovalEventHandler, 949  
  ~InterfaceRemovalEventHandler, 950  
  InterfaceRemovalEventHandler, 950  
  OnInterfaceRemoval, 950  
  operator=, 951

InterfaceRemovalEventHandler Class, 67  
 InterfaceSelector  
     TransportLayerSystem, 1097  
 InterfaceType  
     TransportLayerInterface, 1084  
 InterfaceType\_CameraLink  
     Spinnaker, 284  
 InterfaceType\_CameraLinkHS  
     Spinnaker, 284  
 InterfaceType\_CoaXPress  
     Spinnaker, 284  
 InterfaceType\_Custom  
     Spinnaker, 284  
 InterfaceType\_GigEVision  
     Spinnaker, 284  
 InterfaceType\_USB3Vision  
     Spinnaker, 284  
 InterfaceTypeEnum  
     Spinnaker, 284  
 InterfaceUpdateList  
     TransportLayerSystem, 1097  
 interlaced  
     PNGOption, 1009  
 intfIBase  
     Spinnaker::GenApi, 350  
 intfIBoolean  
     Spinnaker::GenApi, 350  
 intfICategory  
     Spinnaker::GenApi, 350  
 intfICommand  
     Spinnaker::GenApi, 350  
 intfIEnumEntry  
     Spinnaker::GenApi, 350  
 intfIEnumeration  
     Spinnaker::GenApi, 350  
 intfIFloat  
     Spinnaker::GenApi, 350  
 intfIInteger  
     Spinnaker::GenApi, 350  
 intfIPort  
     Spinnaker::GenApi, 350  
 intfIRegister  
     Spinnaker::GenApi, 350  
 intfIString  
     Spinnaker::GenApi, 350  
 intfIValue  
     Spinnaker::GenApi, 350  
 IntRegNode, 951  
     ~IntRegNode, 953  
     IntRegNode, 952, 953  
     SetReference, 953  
 IntRegNode Class, 143  
 IntType\_FLOAT32  
     Spinnaker, 301  
 IntType\_INT16  
     Spinnaker, 301  
 IntType\_INT8  
     Spinnaker, 301  
     IntType\_UINT10  
         Spinnaker, 301  
     IntType\_UINT10P  
         Spinnaker, 301  
     IntType\_UINT10p  
         Spinnaker, 301  
     IntType\_UINT12  
         Spinnaker, 301  
     IntType\_UINT12P  
         Spinnaker, 301  
     IntType\_UINT12p  
         Spinnaker, 301  
     IntType\_UINT14  
         Spinnaker, 301  
     IntType\_UINT16  
         Spinnaker, 301  
     IntType\_UINT8  
         Spinnaker, 301  
     IntType\_UNKNOWN  
         Spinnaker, 301  
 Invalidate  
     Spinnaker::GenApi, 383  
 InvalidateNode  
     CChunkPort, 603  
     CEventPort, 624  
     CPortImpl, 686  
     Node, 990  
     Spinnaker::GenApi, 369  
 InvalidateNodes  
     NodeMap, 1000  
     Spinnaker::GenApi, 369  
 InverseChunkLength  
     DCAM\_CHUNK\_TRAILER, 702  
 Invisible  
     Spinnaker::GenApi, 352  
 ios\_type  
     IDevFileStreamBase< CharType, Traits >, 813  
     ODevFileStreamBase< CharType, Traits >, 1004  
 IP\_ADDRESS\_INVALID  
     AdapterConfig, 184  
 IP\_ADDRESS\_IS\_NOT\_V4  
     AdapterConfig, 184  
 IP\_ADDRESS\_TOO\_LARGE  
     AdapterConfig, 184  
 IP\_ADDRESS\_TOO\_SMALL  
     AdapterConfig, 184  
 ipAddress  
     IpInfo, 954  
 IPersistScript  
     Spinnaker::GenApi, 383  
 IpInfo, 953  
     gateway, 954  
     ipAddress, 954  
     IpInfo, 954  
     subnetLength, 954  
     subnetMask, 954  
 ipInfo  
     AdapterInfo, 397

IPort  
    Spinnaker::GenApi, 383  
IPort Interface, 144  
IPortConstruct  
    Spinnaker::GenApi, 383  
IPortConstruct Interface, 145  
IPortRecorder  
    Spinnaker::GenApi, 383  
IPortRecorder Interface, 146  
IPortReplay  
    Spinnaker::GenApi, 383  
IPortWriteList  
    Spinnaker::GenApi, 384  
IPP  
    Spinnaker, 246  
IPV4Address  
    Spinnaker::GenApi, 351  
IReference  
    Spinnaker::GenApi, 384  
IRegister  
    Spinnaker::GenApi, 384  
IRegister Interfaces, 147  
is\_open  
    IDevFileStreamBase< CharType, Traits >, 813  
    IDevFileStreamBuf< CharType, Traits >, 816  
    ODevFileStreamBase< CharType, Traits >, 1004  
    ODevFileStreamBuf< CharType, Traits >, 1006  
IsAccessModeCacheable  
    Node, 990  
    Spinnaker::GenApi, 370  
IsAvailable  
    INode Interface, 137  
    Pointer Class, 163  
IsCachable  
    Node, 990  
    Spinnaker::GenApi, 370  
IsCacheable  
    INode Interface, 138  
IsCameraDescriptionFileDataReleased  
    CNodeMapFactory, 663  
IsCompressed  
    Image, 876  
IsCRCCheckEnabled  
    IDataStream, 810  
IsDeprecated  
    Node, 991  
    Spinnaker::GenApi, 370  
IsDone  
    CommandNode, 676  
    Spinnaker::GenApi, 370  
ISelector  
    Spinnaker::GenApi, 384  
ISelector Interface, 148  
ISelectorDigit  
    Spinnaker::GenApi, 384  
ISelectorDigit Interface, 149  
IsEmpty  
    CNodeMapFactory, 663  
                CSelectorSet, 696  
IsFeature  
    Node, 991  
    Spinnaker::GenApi, 370  
IsImageInUse  
    IDataStream, 810  
IsImplemented  
    INode Interface, 138  
    Pointer Class, 163  
IsIncomplete  
    IImage, 831  
    Image, 876  
IsInitialized  
    CameraBase, 571  
    ICameraBase, 789  
IsInUse  
    IImage, 832  
    IInterface, 843  
    Image, 876  
    Interface, 931  
    ISystem, 958  
    System, 1047  
IsLoaded  
    CNodeMapFactory, 663  
IsOnSameSubnet  
    AdapterConfig, 186  
IsPEnable  
    Camera, 529  
isPixelFormatColor  
    Polarization.cpp, 1383  
IsPreprocessed  
    CNodeMapFactory, 664  
IsReadable  
    INode Interface, 138, 139  
    Pointer Class, 163  
IsRelease  
    GigEVisionPerformance.cpp, 1355  
IsSelector  
    Node, 991  
IsSelfClearing  
    EnumEntryNode, 725  
    Spinnaker::GenApi, 371  
IsStreamable  
    Node, 991  
    Spinnaker::GenApi, 371  
IsStreaming  
    CameraBase, 571  
    ICameraBase, 790  
    IDataStream, 810  
istream\_type  
    IDevFileStreamBase< CharType, Traits >, 813  
IString  
    Spinnaker::GenApi, 385  
IString Class, 150  
IsValid  
    BasePtr< T, B >, 404  
    CameraBase, 571  
    CGlobalLock, 632

CPointer< T, B >, 681  
 ICameraBase, 790  
 IIInterface, 843  
 Interface, 931  
 IsValidIpAddress  
     AdapterConfig, 186  
 IsValidSubnetMask  
     AdapterConfig, 186  
 IsValueCacheValid  
     Spinnaker::GenApi, 371  
     ValueNode, 1104  
 IsVisible  
     INode Interface, 139  
 IsWritable  
     INode Interface, 139  
     Pointer Class, 163  
 ISystem, 955  
     ~ISystem, 956  
     GetCameras, 957  
     GetInterfaces, 957  
     GetLibraryVersion, 957  
     GetLoggingEventPriorityLevel, 957  
     GetTLNodeMap, 957  
     IsInUse, 958  
     ISystem, 956  
     operator=, 958  
     RegisterEventHandler, 958  
     RegisterInterfaceEventHandler, 958  
     RegisterLoggingEventHandler, 958  
     ReleaseInstance, 958  
     SendActionCommand, 959  
     SetLoggingEventPriorityLevel, 959  
     SystemPtrInternal, 960  
     TLSYSTEM, 960  
     TransportLayerSystem, 1094  
     UnregisterAllLoggingEventHandlers, 959  
     UnregisterEventHandler, 959  
     UnregisterInterfaceEventHandler, 959  
     UnregisterLoggingEventHandler, 960  
     UpdateCameras, 960  
     UpdateInterfaceList, 960  
 ISystem Class, 92  
 ISystemEventHandler, 961  
     ~ISystemEventHandler, 962  
     ISystemEventHandler, 962  
     OnInterfaceArrival, 962  
     OnInterfaceRemoval, 962  
     operator=, 963  
 IsZero  
     Counter, 678  
 Items  
     GVCP\_EVENT\_REQUEST, 777  
     GVCP\_EVENT\_REQUEST\_EXTENDED\_ID, 778  
 IValue  
     Spinnaker::GenApi, 385  
 IValue Class, 151  
 JPEG  
     Spinnaker, 282  
     TIFFOption, 1062  
 JPEG12\_C  
     Spinnaker, 283  
 JPEG2000  
     Spinnaker, 282  
 JPEGOption, 963  
     JPEGOption, 963  
     progressive, 964  
     quality, 964  
     reserved, 964  
 JPG2Option, 964  
     JPG2Option, 965  
     quality, 965  
     reserved, 965  
 jumboPackets  
     AdapterInfo, 397  
 jumboPacketsRegKey  
     AdapterInfo, 397  
 jumboPacketValidValues  
     AdapterInfo, 397  
 k\_HDRGain1  
     HighDynamicRange.cpp, 1358  
 k\_HDRGain2  
     HighDynamicRange.cpp, 1358  
 k\_HDRGain3  
     HighDynamicRange.cpp, 1359  
 k\_HDRGain4  
     HighDynamicRange.cpp, 1359  
 k\_HDRShutter1  
     HighDynamicRange.cpp, 1359  
 k\_HDRShutter2  
     HighDynamicRange.cpp, 1359  
 k\_HDRShutter3  
     HighDynamicRange.cpp, 1359  
 k\_HDRShutter4  
     HighDynamicRange.cpp, 1359  
 k\_LoggingLevel  
     Logging.cpp, 1371  
 k\_numLoops  
     BufferHandling.cpp, 1330  
 KillBufferEvent  
     IDataStream, 810  
 labelClassification  
     Inference.cpp, 1368  
 labelDetection  
     Inference.cpp, 1368  
 Length  
     GVCP\_REQUEST\_HEADER, 781  
     Spinnaker::GenApi, 385  
     U3V\_COMMAND\_HEADER, 1100  
 length  
     gcstring, 767  
 LibraryVersion, 966  
     build, 966  
     major, 966  
     minor, 966  
     type, 966

LIGHTNESS  
    Spinnaker, 313

Linear  
    Spinnaker::GenApi, 351

LineFilterWidth  
    Camera, 529

LineFormat  
    Camera, 529

LineFormat\_LVDS  
    Spinnaker, 284

LineFormat\_NoConnect  
    Spinnaker, 284

LineFormat\_OpenDrain  
    Spinnaker, 284

LineFormat\_OptoCoupled  
    Spinnaker, 284

LineFormat\_RS422  
    Spinnaker, 284

LineFormat\_TriState  
    Spinnaker, 284

LineFormat\_TTL  
    Spinnaker, 284

LineFormatEnums  
    Spinnaker, 284

LineInputFilterSelector  
    Camera, 529

LineInputFilterSelector\_Debounce  
    Spinnaker, 285

LineInputFilterSelector\_Deglitch  
    Spinnaker, 285

LineInputFilterSelectorEnums  
    Spinnaker, 285

LineInverter  
    Camera, 529

LineMode  
    Camera, 530

LineMode\_Input  
    Spinnaker, 285

LineMode\_Output  
    Spinnaker, 285

LineModeEnums  
    Spinnaker, 285

LinePitch  
    Camera, 530

LineSelector  
    Camera, 530

LineSelector\_Line0  
    Spinnaker, 285

LineSelector\_Line1  
    Spinnaker, 285

LineSelector\_Line2  
    Spinnaker, 285

LineSelector\_Line3  
    Spinnaker, 285

LineSelectorEnums  
    Spinnaker, 285

LineSource  
    Camera, 530

    LineSource\_AllPixel  
        Spinnaker, 286

    LineSource\_AnyPixel  
        Spinnaker, 286

    LineSource\_Counter0Active  
        Spinnaker, 286

    LineSource\_Counter1Active  
        Spinnaker, 286

    LineSource\_ExposureActive  
        Spinnaker, 286

    LineSource\_FrameTriggerWait  
        Spinnaker, 286

    LineSource\_Line0  
        Spinnaker, 286

    LineSource\_Line1  
        Spinnaker, 286

    LineSource\_Line2  
        Spinnaker, 286

    LineSource\_Line3  
        Spinnaker, 286

    LineSource\_LogicBlock0  
        Spinnaker, 286

    LineSource\_LogicBlock1  
        Spinnaker, 286

    LineSource\_Off  
        Spinnaker, 286

    LineSource\_PPSSignal  
        Spinnaker, 286

    LineSource\_SerialPort0  
        Spinnaker, 286

    LineSource\_UserOutput0  
        Spinnaker, 286

    LineSource\_UserOutput1  
        Spinnaker, 286

    LineSource\_UserOutput2  
        Spinnaker, 286

    LineSource\_UserOutput3  
        Spinnaker, 286

    LineSourceEnums  
        Spinnaker, 285

    LineStatus  
        Camera, 530

    LineStatusAll  
        Camera, 530

    LinkErrorCount  
        Camera, 531

    LinkUptime  
        Camera, 531

    listIncrement  
        Spinnaker::GenApi, 349

    LittleEndian  
        Spinnaker::GenApi, 349

    LoadAndInject  
        CNodeMapFactory, 664

    LoadFileIntoMemory  
        Inference.cpp, 1368

    LoadFromBag  
        CFeatureBag, 626

**LoadXMLFromFile**  
 NodeMap, 1000  
 Spinnaker::GenApi, 371  
**LoadXMLFromFileInject**  
 NodeMap, 1000  
 Spinnaker::GenApi, 371  
**LoadXMLFromString**  
 NodeMap, 1000  
 Spinnaker::GenApi, 371  
**LoadXMLFromStringInject**  
 NodeMap, 1000  
 Spinnaker::GenApi, 372  
**LoadXMLFromZIPData**  
 NodeMap, 1001  
 Spinnaker::GenApi, 372  
**LoadXMLFromZIPFile**  
 NodeMap, 1001  
 Spinnaker::GenApi, 372  
**Lock**  
 CGlobalLock, 632  
 CLock, 649, 651  
 LockableObject< Object >, 969  
 LockableObject< Object >::Lock, 967  
**LockableObject< Object >, 968**  
 GetLock, 969  
 Lock, 969  
 m\_Lock, 969  
 LockableObject< Object >::Lock, 967  
 ~Lock, 967  
 Lock, 967  
**LockEventHandlerMutex**  
 SystemEventHandlerImpl, 1057  
**LOG\_LEVEL\_ALERT**  
 Spinnaker, 313  
**LOG\_LEVEL\_CRIT**  
 Spinnaker, 313  
**LOG\_LEVEL\_DEBUG**  
 Spinnaker, 313  
**LOG\_LEVEL\_ERROR**  
 Spinnaker, 313  
**LOG\_LEVEL\_FATAL**  
 Spinnaker, 313  
**LOG\_LEVEL\_INFO**  
 Spinnaker, 313  
**LOG\_LEVEL\_NOTICE**  
 Spinnaker, 313  
**LOG\_LEVEL\_NOTSET**  
 Spinnaker, 313  
**LOG\_LEVEL\_OFF**  
 Spinnaker, 313  
**LOG\_LEVEL\_WARN**  
 Spinnaker, 313  
**Logarithmic**  
 Spinnaker::GenApi, 351  
**Logging EventHandler Class, 68**  
**Logging.cpp**  
 k\_LoggingLevel, 1371  
 main, 1371  
**LoggingEventData, 969**  
 ~LoggingEventData, 970  
 GetCategoryName, 971  
 GetLogMessage, 971  
 GetNDC, 971  
 GetPriority, 971  
 GetPriorityName, 972  
 GetThreadName, 972  
 GetTimestamp, 972  
 LoggingEventData, 970  
 SystemImpl, 972  
**LoggingEventDataPtr, 973**  
 LoggingEventDataPtr, 974  
**LoggingEventDataPtr Class, 69**  
**LoggingEventHandler, 975**  
 ~LoggingEventHandler, 976  
 LoggingEventHandler, 976  
 OnLogEvent, 976  
 operator=, 977  
**LoggingEventHandler Class, 70**  
**LoggingEventHandlerImpl, 977**  
**LogicBlock.cpp**  
 AcquireImages, 1372  
 ConfigureLogicBlock, 1372  
 ConfigureTrigger, 1372  
 GrabTwoImages, 1373  
 main, 1373  
 PrintDeviceInfo, 1373  
 ResetExposure, 1373  
 ResetTrigger, 1373  
 RunSingleCamera, 1373  
**LogicBlockLUTInputActivation**  
 Camera, 531  
**LogicBlockLUTInputActivation\_AnyEdge**  
 Spinnaker, 286  
**LogicBlockLUTInputActivation\_FallingEdge**  
 Spinnaker, 286  
**LogicBlockLUTInputActivation\_LevelHigh**  
 Spinnaker, 286  
**LogicBlockLUTInputActivation\_LevelLow**  
 Spinnaker, 286  
**LogicBlockLUTInputActivation\_RisingEdge**  
 Spinnaker, 286  
**LogicBlockLUTInputActivationEnums**  
 Spinnaker, 286  
**LogicBlockLUTInputSelector**  
 Camera, 531  
**LogicBlockLUTInputSelector\_Input0**  
 Spinnaker, 287  
**LogicBlockLUTInputSelector\_Input1**  
 Spinnaker, 287  
**LogicBlockLUTInputSelector\_Input2**  
 Spinnaker, 287  
**LogicBlockLUTInputSelector\_Input3**  
 Spinnaker, 287  
**LogicBlockLUTInputSelectorEnums**  
 Spinnaker, 286  
**LogicBlockLUTInputSource**

Camera, 531  
LogicBlockLUTInputSource\_AcquisitionActive  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Counter0End  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Counter0Start  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Counter1End  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Counter1Start  
    Spinnaker, 287  
LogicBlockLUTInputSource\_ExposureEnd  
    Spinnaker, 287  
LogicBlockLUTInputSource\_ExposureStart  
    Spinnaker, 287  
LogicBlockLUTInputSource\_FrameTriggerWait  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Line0  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Line1  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Line2  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Line3  
    Spinnaker, 287  
LogicBlockLUTInputSource\_LogicBlock0  
    Spinnaker, 287  
LogicBlockLUTInputSource\_LogicBlock1  
    Spinnaker, 287  
LogicBlockLUTInputSource\_UserOutput0  
    Spinnaker, 287  
LogicBlockLUTInputSource\_UserOutput1  
    Spinnaker, 287  
LogicBlockLUTInputSource\_UserOutput2  
    Spinnaker, 287  
LogicBlockLUTInputSource\_UserOutput3  
    Spinnaker, 287  
LogicBlockLUTInputSource\_Zero  
    Spinnaker, 287  
LogicBlockLUTInputSourceEnums  
    Spinnaker, 287  
LogicBlockLUTOutputValue  
    Camera, 531  
LogicBlockLUTOutputValueAll  
    Camera, 532  
LogicBlockLUTRowIndex  
    Camera, 532  
LogicBlockLUTSelector  
    Camera, 532  
LogicBlockLUTSelector\_Enable  
    Spinnaker, 288  
LogicBlockLUTSelector\_Value  
    Spinnaker, 288  
LogicBlockLUTSelectorEnums  
    Spinnaker, 287  
LogicBlockSelector  
    Camera, 532  
LogicBlockSelector\_LogicBlock0  
    Spinnaker, 288  
LogicBlockSelector\_Language  
    Spinnaker, 288  
LogicBlockSelectorEnums  
    Spinnaker, 288  
LookupTable.cpp  
    AcquireImages, 1374  
    ConfigureLookupTables, 1374  
    main, 1374  
    PrintDeviceInfo, 1374  
    PrintRetrieveNodeFailure, 1375  
    ResetLookupTables, 1375  
    RunSingleCamera, 1375  
LUTEnable  
    Camera, 532  
LUTIndex  
    Camera, 532  
LUTSelector  
    Camera, 533  
LUTSelector\_LUT1  
    Spinnaker, 288  
LUTSelectorEnums  
    Spinnaker, 288  
LUTValue  
    Camera, 533  
LUTValueAll  
    Camera, 533  
LZW  
    TIFFOption, 1062  
m\_BaseAddress  
    CTestPortStruct< CDataStruct >, 700  
m\_bOwnLock  
    Clock, 651  
m\_Callbacks  
    Node, 993  
m\_CallbackType  
    CNodeCallback, 656  
m\_DebugCount  
    CGlobalLock, 633  
m\_enabled  
    CGlobalLockUnlocker, 635  
m\_Lock  
    CGlobalLockUnlocker, 635  
    LockableObject< Object >, 969  
m\_lock  
    Clock, 652  
m\_lockEx  
    ClockEx, 654  
m\_NumReads  
    CTestPortStruct< CDataStruct >, 700  
m\_NumWrites  
    CTestPortStruct< CDataStruct >, 701  
m\_pCameraBaseData  
    ICameraBase, 792  
m\_pCameraListData  
    ICameraList, 797  
m\_pChunkAdapter  
    CChunkAdapter, 589

m\_pChunkPort  
     CChunkPort, 604

m\_pEnumeration  
     EnumNode, 731

m\_pEventAdapter  
     CEventAdapter, 611

m\_pEventData  
     EventHandler, 739

m\_pEventPort  
     CEventPort, 624

m\_pInterfaceData  
     IInterface, 845

m\_pInterfaceListData  
     IInterfaceList, 853

m\_pNode  
     CEventPort, 625  
     CNodeCallback, 656

m\_pNodeData  
     Node, 993

m\_pNodeMap  
     Node, 993

m\_pPort  
     CChunkPort, 604

m\_pPortAdapter  
     CChunkPort, 604  
     CEventPort, 625

m\_pT  
     BasePtr< T, B >, 406  
     CPointer< T, B >, 683

m\_ptrPort  
     CPortImpl, 687

m\_pWriteList  
     CPortWriteList, 690

MACAddress  
     Spinnaker::GenApi, 351

Magic  
     GVCP\_REQUEST\_HEADER, 782

main  
     Acquisition.cpp, 1282  
     AcquisitionMultipleCameraRecovery.cpp, 1325  
     AcquisitionMultipleThread.cpp, 1326  
     ActionCommand.cpp, 1329  
     BufferHandling.cpp, 1331  
     ChunkData.cpp, 1333  
     CounterAndTimer.cpp, 1335  
     DeviceEvents.cpp, 1337  
     Enumeration.cpp, 1338  
     Enumeration\_QuickSpin.cpp, 1339  
     EnumerationEvents.cpp, 1340  
     ExceptionHandling.cpp, 1341  
     Exposure.cpp, 1342  
     Exposure\_QuickSpin.cpp, 1344  
     FileAccess\_QuickSpin.cpp, 1346  
     GenTLInfo\_QuickSpin.cpp, 1348  
     GigEVisionPerformance.cpp, 1352  
     HighDynamicRange.cpp, 1357  
     ImageEvents.cpp, 1360  
     ImageFormatControl.cpp, 1362

    ImageFormatControl\_QuickSpin.cpp, 1364  
     Inference.cpp, 1368  
     Logging.cpp, 1371  
     LogicBlock.cpp, 1373  
     LookupTable.cpp, 1374  
     NodeMapCallback.cpp, 1376  
     NodeMapInfo.cpp, 1378  
     Polarization.cpp, 1383  
     SaveToAvi.cpp, 1385  
     Sequencer.cpp, 1387  
     SerialRxTx.cpp, 1390  
     Trigger.cpp, 1393  
     Trigger\_QuickSpin.cpp, 1395

Major  
     Version\_t, 1106

major  
     LibraryVersion, 966

make\_NodeCallback  
     Spinnaker::GenApi, 372

max\_size  
     gcstring, 767

maxChars  
     NodeMapInfo.cpp, 1380

MaxDeviceResetTime  
     Camera, 533

Member\_NodeCallback  
     Member\_NodeCallback< Client, Member >, 980

Member\_NodeCallback< Client, Member >, 978  
     Destroy, 980  
     Member\_NodeCallback, 980  
     operator(), 980  
     PMEMBERFUNC, 979

MemSet  
     CTestPortStruct< CDataStruct >, 699

MergeXMLFiles  
     Spinnaker::GenApi, 373

MILLISECOND  
     SerialRxTx.cpp, 1389

Minor  
     Version\_t, 1106

minor  
     LibraryVersion, 966

MJPEG  
     SaveToAvi.cpp, 1384

MJPGOption, 980  
     frameRate, 981  
     MJPGOption, 981  
     quality, 981  
     reserved, 981

NA  
     Spinnaker::GenApi, 347

NEAREST\_NEIGHBOR  
     ImageUtility, 898  
     Spinnaker, 246

NEAREST\_NEIGHBOR\_AVG  
     Spinnaker, 246

networkFilePath  
     Inference.cpp, 1370

NI  
    Spinnaker::GenApi, 347

No  
    Spinnaker::GenApi, 353

NO\_COLOR\_PROCESSING  
    Spinnaker, 246

NoCache  
    Spinnaker::GenApi, 348

Node, 982  
    ~Node, 985  
    DeregisterCallback, 985  
    GetAccessMode, 985  
    GetAlias, 986  
    GetCachingMode, 986  
    GetCastAlias, 986  
    GetChildren, 986  
    GetDescription, 987  
    GetDeviceName, 987  
    GetDisplayName, 987  
    GetDocuURL, 987  
    GetEventID, 987  
    GetName, 987  
    GetNameSpace, 987  
    GetNodeHandle, 988  
    GetNodeMap, 988  
    GetParents, 988  
    GetPollingTime, 988  
    GetPrincipallInterfaceType, 988  
    GetProperty, 989  
    GetPropertyNames, 989  
    GetSelectedFeatures, 989  
    GetSelectingFeatures, 989  
    GetToolTip, 989  
    GetVisibility, 990  
    ImposeAccessMode, 990  
    ImposeVisibility, 990  
    InvalidateNode, 990  
    IsAccessModeCacheable, 990  
    IsCachable, 990  
    IsDeprecated, 991  
    IsFeature, 991  
    IsSelector, 991  
    IsStreamable, 991  
    m\_Callbacks, 993  
    m\_pNodeData, 993  
    m\_pNodeMap, 993  
    Node, 985  
    operator!=, 991  
    operator==, 991  
    RegisterCallback, 992  
    SetNodeHandle, 992  
    SetNodeMap, 992  
    SetReference, 992

Node Class, 152

NodeCallback Class, 153

NodeList\_t  
    Spinnaker::GenApi, 346

NODEMAP

    ChunkData.cpp, 1332

NodeMap, 993  
    \_Ptr, 1001  
    ~NodeMap, 996  
    ClearXMLCache, 996  
    Clock, 651  
    Connect, 996  
    Destroy, 996  
    GetDeviceName, 997  
    GetDeviceVersion, 997  
    GetGenApiVersion, 997  
    GetLock, 997  
    GetmodelName, 997  
    GetNode, 997  
    GetNodeMapHandle, 998  
    GetNodes, 998  
    GetNumNodes, 998  
    GetProductGuid, 998  
    GetSchemaVersion, 998  
    GetStandardNameSpace, 998  
    GetSupportedSchemaVersions, 999  
    GetToolTip, 999  
    GetVendorName, 999  
    GetVersionGuid, 999  
    InvalidateNodes, 1000  
    LoadXMLFromFile, 1000  
    LoadXMLFromFileInject, 1000  
    LoadXMLFromString, 1000  
    LoadXMLFromStringInject, 1000  
    LoadXMLFromZIPData, 1001  
    LoadXMLFromZIPFile, 1001  
    NodeMap, 995  
    Poll, 1001

NodeMap Class, 154

NodeMapCallback.cpp  
    ChangeHeightAndGain, 1376  
    ConfigureCallbacks, 1376  
    main, 1376  
    OnGainNodeUpdate, 1376  
    OnHeightNodeUpdate, 1376  
    PrintDeviceInfo, 1376  
    ResetCallbacks, 1376  
    RunSingleCamera, 1377

NodeMapFactory Class, 155

NodeMapInfo.cpp  
    chosenRead, 1380  
    Indent, 1378  
    INDIVIDUAL, 1378  
    main, 1378  
    maxChars, 1380  
    PrintBooleanNode, 1378  
    PrintCategoryNodeAndAllFeatures, 1378  
    PrintCommandNode, 1379  
    PrintEnumerationNodeAndCurrentEntry, 1379  
    PrintEnumerationSelector, 1379  
    PrintFloatNode, 1379  
    PrintIntegerNode, 1379  
    PrintNode, 1379

PrintStringNode, 1380  
 PrintValueNode, 1380  
 readType, 1378  
 RunSingleCamera, 1380  
 VALUE, 1378  
 NodeMapRef Class, 156  
 noIncrement  
     Spinnaker::GenApi, 349  
 NONE  
     TIFFOption, 1062  
 None  
     Spinnaker::GenApi, 352  
 npos  
     gcstring, 771  
 NUM\_ACQUISITIONMODE  
     Spinnaker, 230  
 NUM\_ACQUISITIONSTATUSSELECTOR  
     Spinnaker, 231  
 NUM\_ACTIONUNCONDITIONALMODE  
     Spinnaker, 231  
 NUM\_ADCBITDEPTH  
     Spinnaker, 232  
 NUM\_AUTOALGORITHMSELECTOR  
     Spinnaker, 232  
 NUM\_AUTOEXPOSURECONTROLPRIORITY  
     Spinnaker, 232  
 NUM\_AUTOEXPOSURELIGHTINGMODE  
     Spinnaker, 233  
 NUM\_AUTOEXPOSUREMETERINGMODE  
     Spinnaker, 233  
 NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO  
     Spinnaker, 233  
 NUM\_BALANCERATIOSELECTOR  
     Spinnaker, 234  
 NUM\_BALANCEWHITEAUTO  
     Spinnaker, 234  
 NUM\_BALANCEWHITEAUTOPROFILE  
     Spinnaker, 234  
 NUM\_BINNINGHORIZONTALMODE  
     Spinnaker, 235  
 NUM\_BINNINGSELECTOR  
     Spinnaker, 235  
 NUM\_BINNINGVERTICALMODE  
     Spinnaker, 235  
 NUM\_BLACKLEVELAUTO  
     Spinnaker, 236  
 NUM\_BLACKLEVELAUTOBALANCE  
     Spinnaker, 236  
 NUM\_BLACKLEVELSELECTOR  
     Spinnaker, 236  
 NUM\_CHUNKBLACKLEVELSELECTOR  
     Spinnaker, 237  
 NUM\_CHUNKCOUNTERSELECTOR  
     Spinnaker, 237  
 NUM\_CHUNKENCODERSELECTOR  
     Spinnaker, 238  
 NUM\_CHUNKENCODERSTATUS  
     Spinnaker, 238  
 NUM\_CHUNKEXPOSURETIMESELECTOR  
     Spinnaker, 238  
 NUM\_CHUNKGAINSELECTOR  
     Spinnaker, 239  
 NUM\_CHUNKIMAGECOMPONENT  
     Spinnaker, 239  
 NUM\_CHUNKPIXELFORMAT  
     Spinnaker, 240  
 NUM\_CHUNKREGIONID  
     Spinnaker, 240  
 NUM\_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR  
     Spinnaker, 240  
 NUM\_CHUNKSCAN3DCOORDINATESELECTOR  
     Spinnaker, 241  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEM  
     Spinnaker, 241  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE  
     Spinnaker, 241  
 NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR  
     Spinnaker, 242  
 NUM\_CHUNKSCAN3DDISTANCEUNIT  
     Spinnaker, 242  
 NUM\_CHUNKSCAN3DOUTPUTMODE  
     Spinnaker, 243  
 NUM\_CHUNKSELECTOR  
     Spinnaker, 244  
 NUM\_CHUNKSOURCEID  
     Spinnaker, 244  
 NUM\_CHUNKTIMERSELECTOR  
     Spinnaker, 244  
 NUM\_CHUNKTRANSFERSTREAMID  
     Spinnaker, 245  
 NUM\_CLCONFIGURATION  
     Spinnaker, 245  
 NUM\_CLTIMESLOTSCOUNT  
     Spinnaker, 245  
 NUM\_COLORTRANSFORMATIONSELECTOR  
     Spinnaker, 246  
 NUM\_COLORTRANSFORMATIONVALUESELECTOR  
     Spinnaker, 247  
 NUM\_COUNTEREVENTACTIVATION  
     Spinnaker, 247  
 NUM\_COUNTEREVENTSOURCE  
     Spinnaker, 248  
 NUM\_COUNTERRESETACTIVATION  
     Spinnaker, 248  
 NUM\_COUNTERRESETSOURCE  
     Spinnaker, 249  
 NUM\_COUNTERSELECTOR  
     Spinnaker, 249  
 NUM\_COUNTERSTATUS  
     Spinnaker, 249  
 NUM\_COUNTERTRIGGERACTIVATION  
     Spinnaker, 250  
 NUM\_COUNTERTRIGGERSOURCE  
     Spinnaker, 250  
 NUM\_CXPNECTIONTESTMODE  
     Spinnaker, 251

NUM\_CXLINKCONFIGURATION  
    Spinnaker, 252

NUM\_CXLINKCONFIGURATIONPREFERRED  
    Spinnaker, 253

NUM\_CXLINKCONFIGURATIONSTATUS  
    Spinnaker, 254

NUM\_CXPPOCXPSTATUS  
    Spinnaker, 254

NUM\_DECIMATIONHORIZONTALMODE  
    Spinnaker, 254

NUM\_DECIMATIONSELECTOR  
    Spinnaker, 255

NUM\_DECIMATIONVERTICALMODE  
    Spinnaker, 255

NUM\_DEFECTCORRECTIONMODE  
    Spinnaker, 255

NUM\_DEINTERLACING  
    Spinnaker, 256

NUM\_DEVICECHARACTERSET  
    Spinnaker, 256

NUM\_DEVICECLOCKSELECTOR  
    Spinnaker, 257

NUM\_DEVICECONNECTIONSTATUS  
    Spinnaker, 257

NUM\_DEVICEINDICATORMODE  
    Spinnaker, 258

NUM\_DEVICELINKHEARTBEATMODE  
    Spinnaker, 258

NUM\_DEVICELINKTHROUGHPUTLIMITMODE  
    Spinnaker, 259

NUM\_DEVICEPOWERSUPPLYSELECTOR  
    Spinnaker, 259

NUM\_DEVICEREGISTERSENDIANCESS  
    Spinnaker, 259

NUM\_DEVICESCANTYPE  
    Spinnaker, 259

NUM\_DEVICESERIALPORTBAUDRATE  
    Spinnaker, 260

NUM\_DEVICESERIALPORTSELECTOR  
    Spinnaker, 260

NUM\_DEVICESTREAMCHANNELENDIANCESS  
    Spinnaker, 260

NUM\_DEVICESTREAMCHANNELTYPE  
    Spinnaker, 261

NUM\_DEVICETAPGEOMETRY  
    Spinnaker, 262

NUM\_DEVICETEMPERATURESELECTOR  
    Spinnaker, 262

NUM\_DEVICETLTTYPE  
    Spinnaker, 263

NUM\_DEVICETYPE  
    Spinnaker, 263

NUM\_ENCODERMODE  
    Spinnaker, 264

NUM\_ENCODEROUTPUTMODE  
    Spinnaker, 264

NUM\_ENCODERRESETACTIVATION  
    Spinnaker, 265

NUM\_ENCODERRESETSOURCE  
    Spinnaker, 266

NUM\_ENCODERSELECTOR  
    Spinnaker, 266

NUM\_ENCODERSOURCEA  
    Spinnaker, 266

NUM\_ENCODERSOURCEB  
    Spinnaker, 267

NUM\_ENCODERSTATUS  
    Spinnaker, 267

NUM\_EVENTNOTIFICATION  
    Spinnaker, 270

NUM\_EVENTSELECTOR  
    Spinnaker, 270

NUM\_EXPOSUREACTIVEMODE  
    Spinnaker, 271

NUM\_EXPOSUREAUTO  
    Spinnaker, 271

NUM\_EXPOSUREMODE  
    Spinnaker, 271

NUM\_EXPOSURETIMEMODE  
    Spinnaker, 273

NUM\_EXPOSURETIMESELECTOR  
    Spinnaker, 273

NUM\_FILEOPENMODE  
    Spinnaker, 273

NUM\_FILEOPERATIONSELECTOR  
    Spinnaker, 274

NUM\_FILEOPERATIONSTATUS  
    Spinnaker, 274

NUM\_FILESELECTOR  
    Spinnaker, 274

NUM\_GAINAUTO  
    Spinnaker, 275

NUM\_GAINAUTOBALANCE  
    Spinnaker, 275

NUM\_GAINSELECTOR  
    Spinnaker, 276

NUM\_GEVCCP  
    Spinnaker, 277

NUM\_GEVCURRENTPHYSICALLINKCONFIGURATION  
    Spinnaker, 277

NUM\_GEVGVCPEXTENDEDSTATUSCODESELECTOR  
    Spinnaker, 277

NUM\_GEVGVSPEXTENDEDIDMODE  
    Spinnaker, 278

NUM\_GEVIEEE1588CLOCKACCURACY  
    Spinnaker, 278

NUM\_GEVIEEE1588MODE  
    Spinnaker, 278

NUM\_GEVIEEE1588STATUS  
    Spinnaker, 279

NUM\_GEVIPCONFIGURATIONSTATUS  
    Spinnaker, 279

NUM\_GEVPHYSICALLINKCONFIGURATION  
    Spinnaker, 279

NUM\_GEVSUPPORTEDOPTIONSELECTOR  
    Spinnaker, 280

NUM\_IMAGECOMPONENTSELECTOR  
Spinnaker, 281

NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION  
Spinnaker, 281

NUM\_IMAGECOMPRESSIONMODE  
Spinnaker, 282

NUM\_IMAGECOMPRESSIONRATEOPTION  
Spinnaker, 282

NUM\_LINEFORMAT  
Spinnaker, 284

NUM\_LINEINPUTFILTERSELECTOR  
Spinnaker, 285

NUM\_LINEMODE  
Spinnaker, 285

NUM\_LINESELECTOR  
Spinnaker, 285

NUM\_LINESOURCE  
Spinnaker, 286

NUM\_LOGICBLOCKLUTINPUTACTIVATION  
Spinnaker, 286

NUM\_LOGICBLOCKLUTINPUTSELECTOR  
Spinnaker, 287

NUM\_LOGICBLOCKLUTINPUTSOURCE  
Spinnaker, 287

NUM\_LOGICBLOCKLUTSELECTOR  
Spinnaker, 288

NUM\_LOGICBLOCKSELECTOR  
Spinnaker, 288

NUM\_LUTSELECTOR  
Spinnaker, 288

NUM\_PIXELCOLORFILTER  
Spinnaker, 289

NUM\_PIXELFORMAT  
Spinnaker, 295

NUM\_PIXELFORMATINFOSELECTOR  
Spinnaker, 301

NUM\_PIXELSIZE  
Spinnaker, 302

NUM\_REGIONDESTINATION  
Spinnaker, 303

NUM\_REGIONMODE  
Spinnaker, 303

NUM\_REGIONSELECTOR  
Spinnaker, 303

NUM\_RGBTRANSFORMLIGHTSOURCE  
Spinnaker, 304

NUM\_SCAN3DCOORDINATEREFERENCESELECTOR  
Spinnaker, 304

NUM\_SCAN3DCOORDINATESELECTOR  
Spinnaker, 305

NUM\_SCAN3DCORDINATESYSTEM  
Spinnaker, 305

NUM\_SCAN3DCORDINATESYSTEMREFERENCE  
Spinnaker, 305

NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR  
Spinnaker, 306

NUM\_SCAN3DDISTANCEUNIT  
Spinnaker, 306

NUM\_SCAN3DOUTPUTMODE  
Spinnaker, 307

NUM\_SENORDIGITIZATIONTAPS  
Spinnaker, 307

NUM\_SENSORSHUTTERMODE  
Spinnaker, 308

NUM\_SENSORTAPS  
Spinnaker, 308

NUMSEQUENCERCONFIGURATIONMODE  
Spinnaker, 308

NUMSEQUENCERCONFIGURATIONVALID  
Spinnaker, 309

NUMSEQUENCERMODE  
Spinnaker, 309

NUMSEQUENCERSETVALID  
Spinnaker, 309

NUMSEQUENCERTRIGGERACTIVATION  
Spinnaker, 310

NUMSEQUENCERTRIGGERSOURCE  
Spinnaker, 310

NUM\_SERIALPORTBAUDRATE  
Spinnaker, 310

NUM\_SERIALPORTPARITY  
Spinnaker, 311

NUM\_SERIALPORTSELECTOR  
Spinnaker, 311

NUM\_SERIALPORTSOURCE  
Spinnaker, 311

NUM\_SERIALPORTSTOPBITS  
Spinnaker, 312

NUM\_SOFTWARESIGNALSELECTOR  
Spinnaker, 312

NUM\_SOURCESELECTOR  
Spinnaker, 312

NUM\_STATISTICS\_CHANNELS  
Spinnaker, 313

NUM\_TESTPATTERN  
Spinnaker, 315

NUM\_TESTPATTERNGENERATORSELECTOR  
Spinnaker, 315

NUM\_TIMERSELECTOR  
Spinnaker, 317

NUM\_TIMERSTATUS  
Spinnaker, 317

NUM\_TIMERTRIGGERACTIVATION  
Spinnaker, 317

NUM\_TIMERTRIGGERSOURCE  
Spinnaker, 319

NUM\_TRANSFERCOMPONENTSELECTOR  
Spinnaker, 319

NUM\_TRANSFERCONTROLMODE  
Spinnaker, 320

NUM\_TRANSFEROPERATIONMODE  
Spinnaker, 320

NUM\_TRANSFERQUEUEMODE  
Spinnaker, 320

NUM\_TRANSFERSELECTOR  
Spinnaker, 321

NUM\_TRANSFERSTATUSSELECTOR  
    Spinnaker, 321

NUM\_TRANSFERTRIGGERACTIVATION  
    Spinnaker, 322

NUM\_TRANSFERTRIGGERMODE  
    Spinnaker, 322

NUM\_TRANSFERTRIGGERSELECTOR  
    Spinnaker, 322

NUM\_TRANSFERTRIGGERSOURCE  
    Spinnaker, 323

NUM\_TRIGGERACTIVATION  
    Spinnaker, 324

NUM\_TRIGGERMODE  
    Spinnaker, 324

NUM\_TRIGGEROVERLAP  
    Spinnaker, 324

NUM\_TRIGGERSELECTOR  
    Spinnaker, 325

NUM\_TRIGGERSOURCE  
    Spinnaker, 325

NUM\_USEROUTPUTSELECTOR  
    Spinnaker, 326

NUM\_USERSETDEFAULT  
    Spinnaker, 326

NUM\_USERSETSELECTOR  
    Spinnaker, 326

NUM\_WHITECLIPSELECTOR  
    Spinnaker, 327

NumAttachedChunks  
    AttachStatistics\_t, 399

numBuffers  
    BufferHandling.cpp, 1330

NumChunkPorts  
    AttachStatistics\_t, 399

NumChunks  
    AttachStatistics\_t, 399

NUMDEVICEACCESSSTATUS  
    Spinnaker, 256

NUMDEVICECURRENTSPEED  
    Spinnaker, 257

NUMDEVICEENDIANESSMECHANISM  
    Spinnaker, 258

NUMDEVICETYPE  
    Spinnaker, 263

NUMFILTERDRIVERSTATUS  
    Spinnaker, 275

NUMGENICAMXMLLOCATION  
    Spinnaker, 276

NUMGEVCCP  
    Spinnaker, 276

NUMGUIMXMLLOCATION  
    Spinnaker, 280

numImagesGrabbed  
    GrabInfo, 771

NumImagesToGrab  
    GigEVisionPerformance.cpp, 1355

numIncompleteImages  
    GrabInfo, 772

NUMINTERFACETYPE  
    Spinnaker, 284

NumLinks  
    CNodeMapFactory::NodeStatistics\_t, 1002

NumNodes  
    CNodeMapFactory::NodeStatistics\_t, 1002

NUMPOESTATUS  
    Spinnaker, 302

NumProperties  
    CNodeMapFactory::NodeStatistics\_t, 1002

numRemovals  
    GrabInfo, 772

NUMSTREAMBUFFERCOUNTMODE  
    Spinnaker, 314

NUMSTREAMBUFFERHANDLINGMODE  
    Spinnaker, 314

NUMSTREAMTYPE  
    Spinnaker, 315

NumStrings  
    CNodeMapFactory::NodeStatistics\_t, 1002

NUMTLLTYPE  
    Spinnaker, 319

NumToString  
    Conversion, 187

ODevFileStream  
    Spinnaker::GenApi, 347

ODevFileStreamBase< CharType, Traits >, 1003  
    close, 1004  
    filebuf\_type, 1004  
    ios\_type, 1004  
    is\_open, 1004  
    open, 1004  
    ostream\_type, 1004  
    rdbuf, 1005

ODevFileStreamBuf  
    ODevFileStreamBuf< CharType, Traits >, 1006

ODevFileStreamBuf< CharType, Traits >, 1005  
    ~ODevFileStreamBuf, 1006  
    close, 1006  
    is\_open, 1006  
    ODevFileStreamBuf, 1006  
    open, 1006  
    overflow, 1007  
    sync, 1007  
    xputn, 1007

OffsetX  
    Camera, 534

OffsetY  
    Camera, 534

OnDeviceArrival  
    DeviceArrivalEventHandler, 704  
    IDeviceArrivalEventHandler, 818  
    IInterfaceEventHandler, 850  
    InterfaceEventHandler, 938  
    InterfaceEventHandlerImpl, 941, 942

OnDeviceEvent  
    DeviceEventHandler, 707  
    DeviceEventHandlerImpl, 710

IDeviceEventHandler, 821  
 OnDeviceRemoval  
     DeviceRemovalEventHandler, 712  
     IDeviceRemovalEventHandler, 823  
     IInterfaceEventHandler, 850  
     InterfaceEventHandler, 938  
     InterfaceEventHandlerImpl, 942  
 OnGainNodeUpdate  
     NodeMapCallback.cpp, 1376  
 OnHeightNodeUpdate  
     NodeMapCallback.cpp, 1376  
 OnImageEvent  
     IImageEventHandler, 836  
     ImageEventHandler, 884  
     ImageEventHandlerImpl, 887  
 OnInterfaceArrival  
     IInterfaceArrivalEventHandler, 847  
     InterfaceArrivalEventHandler, 935  
     ISystemEventHandler, 962  
     SystemEventHandler, 1054  
     SystemEventHandlerImpl, 1057  
 OnInterfaceRemoval  
     IInterfaceRemovalEventHandler, 855  
     InterfaceRemovalEventHandler, 950  
     ISystemEventHandler, 962  
     SystemEventHandler, 1055  
     SystemEventHandlerImpl, 1058  
 OnLogEvent  
     ILoggingEventHandler, 857  
     LoggingEventHandler, 976  
 Open  
     SpinVideo, 1033, 1035  
 open  
     IDevFileStreamBase< CharType, Traits >, 814  
     IDevFileStreamBuf< CharType, Traits >, 816  
     ODevFileStreamBase< CharType, Traits >, 1004  
     ODevFileStreamBuf< CharType, Traits >, 1006  
 openFile  
     FileProtocolAdapter, 749  
 OpenFileToRead  
      FileAccess\_QuickSpin.cpp, 1346  
 OpenFileToWrite  
      FileAccess\_QuickSpin.cpp, 1347  
 operator bool  
     BasePtr< T, B >, 404  
     CPointer< T, B >, 681  
 operator const char \*  
     gcstring, 767  
 operator delete  
     double\_avector\_t, 714  
     gcstring, 767  
     int64\_avector\_t, 922  
 operator new  
     double\_avector\_t, 714  
     gcstring, 767  
     int64\_avector\_t, 922  
 operator T\*  
     BasePtr< T, B >, 404  
     CPointer< T, B >, 681  
     operator unsigned int  
         Counter, 678  
 operator!=  
     CPointer< T, B >, 681, 682  
     Exception, 744  
     gcstring, 768  
     Node, 991  
     Spinnaker::GenApi, 373  
 operator<  
     gcstring, 769  
 operator<<  
     GCString.h, 1202  
     Spinnaker::GenApi, 373  
 operator>  
     gcstring, 769  
 operator>>  
     GCString.h, 1203  
     Spinnaker::GenApi, 375  
 operator\*  
     CPointer< T, B >, 682  
     EnumNode, 730  
     FloatNode, 756  
     IntegerNode, 928  
     Spinnaker::GenApi, 373  
     StringNode, 1039  
 operator()  
     CEnumerationTRef< EnumT >, 607  
     CNodeCallback, 656  
     CommandNode, 677  
     CPointer< T, B >, 682  
     FloatNode, 756  
     Function\_NodeCallback< Function >, 761  
     IntegerNode, 927  
     Member\_NodeCallback< Client, Member >, 980  
     Spinnaker::GenApi, 373  
     StringNode, 1039  
 operator+  
     gcstring, 770  
 operator++  
     Counter, 678  
 operator+=  
     gcstring, 768  
 operator->  
     BasePtr< T, B >, 404  
     CPointer< T, B >, 682  
 operator--  
     Counter, 678, 679  
 operator=  
     BasePtr< T, B >, 405  
     BooleanNode, 410  
     CameraBase, 572  
     CameraList, 580  
     CEnumerationTRef< EnumT >, 608  
     CFloatPtr, 629  
     CNodeMapFactory, 664  
     CNodeMapRef, 667  
     CNodeMapRefT< TCameraParams >, 673, 674

CPointer< T, B >, 682  
DeviceArrivalEventHandler, 704  
DeviceEventHandler, 707  
DeviceRemovalEventHandler, 712  
double\_vector\_t, 714  
EnumNode, 730  
EventHandler, 738  
Exception, 744  
FloatNode, 756  
gcstring, 769  
ICameraBase, 790  
ICameraList, 795  
IDeviceArrivalEventHandler, 818  
IDeviceEventHandler, 821  
IDeviceRemovalEventHandler, 823  
IImageEventHandler, 836  
IInterface, 843  
IInterfaceArrivalEventHandler, 847  
IInterfaceEventHandler, 850  
IInterfaceList, 852  
IInterfaceRemovalEventHandler, 855  
ILoggingEventHandler, 857  
ImageEventHandler, 884  
ImagePtr, 890  
ImageStatistics, 896  
InferenceBoundingBoxResult, 917  
int64\_vector\_t, 922  
IntegerNode, 928  
InterfaceArrivalEventHandler, 936  
InterfaceEventHandler, 938  
InterfaceList, 946  
InterfaceRemovalEventHandler, 951  
ISystem, 958  
ISystemEventHandler, 963  
LoggingEventHandler, 977  
Spinnaker::GenApi, 374  
StringNode, 1039  
SystemEventHandler, 1055  
operator==  
    BasePtr< T, B >, 405, 406  
    CFeatureBag, 627  
    CPointer< T, B >, 683  
    Exception, 745  
    gcstring, 769  
    Node, 991  
    Spinnaker, 328  
    Spinnaker::GenApi, 375  
operator[]  
    CameraList, 580  
    double\_vector\_t, 714  
    ICameraList, 795  
    IInterfaceList, 853  
    int64\_vector\_t, 922  
    InterfaceList, 946  
ostream\_type  
    ODevFileStreamBase< CharType, Traits >, 1004  
overflow  
    ODevFileStreamBuf< CharType, Traits >, 1007  
PACKBITS  
    TIFFOption, 1062  
PacketDelayToSet  
    GigEVisionPerformance.cpp, 1355  
PacketResendRequestCount  
    Camera, 534  
PacketSizeToSet  
    GigEVisionPerformance.cpp, 1355  
ParseArguments  
    GigEVisionPerformance.cpp, 1353  
PAYLOAD\_TYPE\_CHUNK\_DATA  
    Spinnaker, 289  
PAYLOAD\_TYPE\_CHUNK\_ONLY  
    Spinnaker, 289  
PAYLOAD\_TYPE\_CUSTOM\_ID  
    Spinnaker, 289  
PAYLOAD\_TYPE\_DEVICE\_SPECIFIC  
    Spinnaker, 289  
PAYLOAD\_TYPE\_EXTENDED\_CHUNK  
    Spinnaker, 289  
PAYLOAD\_TYPE\_FILE  
    Spinnaker, 289  
PAYLOAD\_TYPE\_H264  
    Spinnaker, 289  
PAYLOAD\_TYPE\_IMAGE  
    Spinnaker, 289  
PAYLOAD\_TYPE\_JPEG  
    Spinnaker, 289  
PAYLOAD\_TYPE\_JPEG2000  
    Spinnaker, 289  
PAYLOAD\_TYPE\_MULTI\_PART  
    Spinnaker, 289  
PAYLOAD\_TYPE\_RAW\_DATA  
    Spinnaker, 289  
PAYLOAD\_TYPE\_UNKNOWN  
    Spinnaker, 289  
PayloadSize  
    Camera, 534  
PayloadTypeIDs  
    Spinnaker, 288  
pbackfail  
    IDevFileStreamBuf< CharType, Traits >, 816  
PCFreq  
    PerformanceCounter, 189  
PerformanceCounter, 188  
    CounterStart, 189  
    GetPerformanceCounter, 189  
    PCFreq, 189  
    StartPerformanceCounter, 189  
Persistence Class, 157  
PersistFeature  
    CFeatureBag, 627  
    Spinnaker::GenApi, 375  
PGM  
    Spinnaker, 282  
PGMOption, 1007  
    binaryFile, 1008  
    PGMOption, 1008

reserved, 1008  
PixelColorFilter  
    Camera, 534  
PixelColorFilter\_BayerBG  
    Spinnaker, 289  
PixelColorFilter\_BayerGB  
    Spinnaker, 289  
PixelColorFilter\_BayerGR  
    Spinnaker, 289  
PixelColorFilter\_BayerRG  
    Spinnaker, 289  
PixelColorFilter\_None  
    Spinnaker, 289  
PixelColorFilterEnums  
    Spinnaker, 289  
PixelDynamicRangeMax  
    Camera, 535  
PixelDynamicRangeMin  
    Camera, 535  
PixelFormat  
    Camera, 535  
PixelFormat\_B10  
    Spinnaker, 292  
PixelFormat\_B12  
    Spinnaker, 292  
PixelFormat\_B12\_Jpeg  
    Spinnaker, 295  
PixelFormat\_B16  
    Spinnaker, 292  
PixelFormat\_B8  
    Spinnaker, 292  
PixelFormat\_BayerBG10  
    Spinnaker, 290  
PixelFormat\_BayerBG10p  
    Spinnaker, 290  
PixelFormat\_BayerBG10Packed  
    Spinnaker, 290  
PixelFormat\_BayerBG12  
    Spinnaker, 290  
PixelFormat\_BayerBG12p  
    Spinnaker, 290  
PixelFormat\_BayerBG12Packed  
    Spinnaker, 290  
PixelFormat\_BayerBG16  
    Spinnaker, 290  
PixelFormat\_BayerBG8  
    Spinnaker, 290  
PixelFormat\_BayerGB10  
    Spinnaker, 290  
PixelFormat\_BayerGB10p  
    Spinnaker, 290  
PixelFormat\_BayerGB10Packed  
    Spinnaker, 290  
PixelFormat\_BayerGB12  
    Spinnaker, 290  
PixelFormat\_BayerGB12p  
    Spinnaker, 290  
PixelFormat\_BayerGB12Packed  
    Spinnaker, 290  
reserved, 1008  
Spinnaker, 290  
PixelFormat\_BayerGB16  
    Spinnaker, 290  
PixelFormat\_BayerGB8  
    Spinnaker, 290  
PixelFormat\_BayerGR10  
    Spinnaker, 291  
PixelFormat\_BayerGR10p  
    Spinnaker, 290  
PixelFormat\_BayerGR10Packed  
    Spinnaker, 290  
PixelFormat\_BayerGR12  
    Spinnaker, 291  
PixelFormat\_BayerGR12p  
    Spinnaker, 290  
PixelFormat\_BayerGR12Packed  
    Spinnaker, 290  
PixelFormat\_BayerGR16  
    Spinnaker, 290  
PixelFormat\_BayerGR8  
    Spinnaker, 289  
PixelFormat\_BayerRG10  
    Spinnaker, 291  
PixelFormat\_BayerRG10p  
    Spinnaker, 290  
PixelFormat\_BayerRG10Packed  
    Spinnaker, 290  
PixelFormat\_BayerRG12  
    Spinnaker, 291  
PixelFormat\_BayerRG12p  
    Spinnaker, 290  
PixelFormat\_BayerRG12Packed  
    Spinnaker, 290  
PixelFormat\_BayerRG16  
    Spinnaker, 290  
PixelFormat\_BayerRG8  
    Spinnaker, 289  
PixelFormat\_BayerRGPolarized10p  
    Spinnaker, 294  
PixelFormat\_BayerRGPolarized12p  
    Spinnaker, 295  
PixelFormat\_BayerRGPolarized16  
    Spinnaker, 295  
PixelFormat\_BayerRGPolarized8  
    Spinnaker, 294  
PixelFormat\_BGR10  
    Spinnaker, 291  
PixelFormat\_BGR10p  
    Spinnaker, 291  
PixelFormat\_BGR12  
    Spinnaker, 291  
PixelFormat\_BGR12p  
    Spinnaker, 291  
PixelFormat\_BGR14  
    Spinnaker, 291  
PixelFormat\_BGR16  
    Spinnaker, 291  
PixelFormat\_BGR565p

Spinnaker, 291  
PixelFormat\_BGR8  
    Spinnaker, 290  
PixelFormat\_BGRa10  
    Spinnaker, 291  
PixelFormat\_BGRa10p  
    Spinnaker, 291  
PixelFormat\_BGRa12  
    Spinnaker, 291  
PixelFormat\_BGRa12p  
    Spinnaker, 291  
PixelFormat\_BGRa14  
    Spinnaker, 291  
PixelFormat\_BGRa16  
    Spinnaker, 291  
PixelFormat\_BGRa8  
    Spinnaker, 290  
PixelFormat\_BiColorBGRG10  
    Spinnaker, 293  
PixelFormat\_BiColorBGRG10p  
    Spinnaker, 293  
PixelFormat\_BiColorBGRG12  
    Spinnaker, 293  
PixelFormat\_BiColorBGRG12p  
    Spinnaker, 293  
PixelFormat\_BiColorBGRG8  
    Spinnaker, 292  
PixelFormat\_BiColorRGBG10  
    Spinnaker, 293  
PixelFormat\_BiColorRGBG10p  
    Spinnaker, 293  
PixelFormat\_BiColorRGBG12  
    Spinnaker, 293  
PixelFormat\_BiColorRGBG12p  
    Spinnaker, 293  
PixelFormat\_BiColorRGBG8  
    Spinnaker, 293  
PixelFormat\_Confidence1  
    Spinnaker, 292  
PixelFormat\_Confidence16  
    Spinnaker, 292  
PixelFormat\_Confidence1p  
    Spinnaker, 292  
PixelFormat\_Confidence32f  
    Spinnaker, 292  
PixelFormat\_Confidence8  
    Spinnaker, 292  
PixelFormat\_Coord3D\_A10p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_A12p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_A16  
    Spinnaker, 292  
PixelFormat\_Coord3D\_A32f  
    Spinnaker, 292  
PixelFormat\_Coord3D\_A8  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC10p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC10p\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC12p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC12p\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC16  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC16\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC32f  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC32f\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC8  
    Spinnaker, 292  
PixelFormat\_Coord3D\_ABC8\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC10p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC10p\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC12p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC12p\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC16  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC16\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC32f  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC32f\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC8  
    Spinnaker, 292  
PixelFormat\_Coord3D\_AC8\_Planar  
    Spinnaker, 292  
PixelFormat\_Coord3D\_B10p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_B12p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_B16  
    Spinnaker, 292  
PixelFormat\_Coord3D\_B32f  
    Spinnaker, 292  
PixelFormat\_Coord3D\_B8  
    Spinnaker, 292  
PixelFormat\_Coord3D\_C10p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_C12p  
    Spinnaker, 292  
PixelFormat\_Coord3D\_C16  
    Spinnaker, 292  
PixelFormat\_Coord3D\_C32f  
    Spinnaker, 292  
PixelFormat\_Coord3D\_C8

Spinnaker, 292  
PixelFormat\_G10  
    Spinnaker, 291  
PixelFormat\_G12  
    Spinnaker, 292  
PixelFormat\_G16  
    Spinnaker, 292  
PixelFormat\_G8  
    Spinnaker, 291  
PixelFormat\_GB12\_Jpeg  
    Spinnaker, 295  
PixelFormat\_GR12\_Jpeg  
    Spinnaker, 295  
PixelFormat\_JPEGColor8  
    Spinnaker, 295  
PixelFormat\_JPEGMono8  
    Spinnaker, 295  
PixelFormat\_LLCBayerRG8  
    Spinnaker, 295  
PixelFormat\_LLCMono8  
    Spinnaker, 295  
PixelFormat\_Mono10  
    Spinnaker, 290  
PixelFormat\_Mono10p  
    Spinnaker, 290  
PixelFormat\_Mono10Packed  
    Spinnaker, 290  
PixelFormat\_Mono12  
    Spinnaker, 290  
PixelFormat\_Mono12p  
    Spinnaker, 290  
PixelFormat\_Mono12Packed  
    Spinnaker, 290  
PixelFormat\_Mono14  
    Spinnaker, 290  
PixelFormat\_Mono16  
    Spinnaker, 289  
PixelFormat\_Mono16s  
    Spinnaker, 290  
PixelFormat\_Mono1p  
    Spinnaker, 290  
PixelFormat\_Mono2p  
    Spinnaker, 290  
PixelFormat\_Mono32f  
    Spinnaker, 290  
PixelFormat\_Mono4p  
    Spinnaker, 290  
PixelFormat\_Mono8  
    Spinnaker, 289  
PixelFormat\_Mono8s  
    Spinnaker, 290  
PixelFormat\_Polarized10p  
    Spinnaker, 294  
PixelFormat\_Polarized12p  
    Spinnaker, 294  
PixelFormat\_Polarized16  
    Spinnaker, 294  
PixelFormat\_Polarized8  
    Spinnaker, 294  
PixelFormat\_R10  
    Spinnaker, 291  
PixelFormat\_R12  
    Spinnaker, 291  
PixelFormat\_R12\_Jpeg  
    Spinnaker, 295  
PixelFormat\_R16  
    Spinnaker, 291  
PixelFormat\_R8  
    Spinnaker, 291  
PixelFormat\_Raw16  
    Spinnaker, 295  
PixelFormat\_Raw8  
    Spinnaker, 295  
PixelFormat\_RGB10  
    Spinnaker, 291  
PixelFormat\_RGB10\_Planar  
    Spinnaker, 291  
PixelFormat\_RGB10p  
    Spinnaker, 291  
PixelFormat\_RGB10p32  
    Spinnaker, 291  
PixelFormat\_RGB12  
    Spinnaker, 291  
PixelFormat\_RGB12\_Planar  
    Spinnaker, 291  
PixelFormat\_RGB12p  
    Spinnaker, 291  
PixelFormat\_RGB14  
    Spinnaker, 291  
PixelFormat\_RGB16  
    Spinnaker, 291  
PixelFormat\_RGB16\_Planar  
    Spinnaker, 291  
PixelFormat\_RGB16s  
    Spinnaker, 291  
PixelFormat\_RGB32f  
    Spinnaker, 291  
PixelFormat\_RGB565p  
    Spinnaker, 291  
PixelFormat\_RGB8  
    Spinnaker, 291  
PixelFormat\_RGB8\_Planar  
    Spinnaker, 291  
PixelFormat\_RGB8Packed  
    Spinnaker, 289  
PixelFormat\_RGBa10  
    Spinnaker, 291  
PixelFormat\_RGBa10p  
    Spinnaker, 291  
PixelFormat\_RGBa12  
    Spinnaker, 291  
PixelFormat\_RGBa12p  
    Spinnaker, 291  
PixelFormat\_RGBa14  
    Spinnaker, 291  
PixelFormat\_RGBa16

- Spinnaker, 291
- PixelFormat\_RGBa32f
  - Spinnaker, 291
- PixelFormat\_RGBa8
  - Spinnaker, 291
- PixelFormat\_SCF1WBWG10
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG10p
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG12
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG12p
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG14
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG16
  - Spinnaker, 293
- PixelFormat\_SCF1WBWG8
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB10
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB10p
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB12
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB12p
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB14
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB16
  - Spinnaker, 293
- PixelFormat\_SCF1WGWB8
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR10
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR10p
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR12
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR12p
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR14
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR16
  - Spinnaker, 293
- PixelFormat\_SCF1WGWR8
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG10
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG10p
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG12
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG12p
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG14
  - Spinnaker, 293
- PixelFormat\_SCF1WRWG16
  - Spinnaker, 293
- Spinnaker, 293
- PixelFormat\_SCF1WRWG8
  - Spinnaker, 293
- PixelFormat\_YCbCr10\_CbYCr
  - Spinnaker, 293
- PixelFormat\_YCbCr10p\_CbYCr
  - Spinnaker, 293
- PixelFormat\_YCbCr12\_CbYCr
  - Spinnaker, 293
- PixelFormat\_YCbCr12p\_CbYCr
  - Spinnaker, 293
- PixelFormat\_YCbCr411\_8
  - Spinnaker, 290
- PixelFormat\_YCbCr411\_8\_CbYYCrYY
  - Spinnaker, 293
- PixelFormat\_YCbCr422\_10
  - Spinnaker, 293
- PixelFormat\_YCbCr422\_10\_CbYCrY
  - Spinnaker, 293
- PixelFormat\_YCbCr422\_10p
  - Spinnaker, 293
- PixelFormat\_YCbCr422\_10p\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr422\_12
  - Spinnaker, 294
- PixelFormat\_YCbCr422\_12\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr422\_12p
  - Spinnaker, 294
- PixelFormat\_YCbCr422\_12p\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr422\_8
  - Spinnaker, 290
- PixelFormat\_YCbCr422\_8\_CbYCrY
  - Spinnaker, 293
- PixelFormat\_YCbCr601\_10\_CbYCr
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_10p\_CbYCr
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_12\_CbYCr
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_12p\_CbYCr
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_422\_10\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_422\_10p\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_422\_10p\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_422\_12\_CbYCrY
  - Spinnaker, 294
- PixelFormat\_YCbCr601\_422\_12p\_CbYCrY
  - Spinnaker, 294

Spinnaker, 294  
PixelFormat\_YCbCr601\_422\_12p\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr601\_422\_8  
    Spinnaker, 294  
PixelFormat\_YCbCr601\_422\_8\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr601\_8\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_10\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_10p\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_12\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_12p\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_10  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_10\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_10p  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_10p\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_12  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_12\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_12p  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_12p\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_8  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_422\_8\_CbYCrY  
    Spinnaker, 294  
PixelFormat\_YCbCr709\_8\_CbYCr  
    Spinnaker, 294  
PixelFormat\_YCbCr8  
    Spinnaker, 290  
PixelFormat\_YCbCr8\_CbYCr  
    Spinnaker, 293  
PixelFormat\_YUV411\_8\_UYYVYY  
    Spinnaker, 294  
PixelFormat\_YUV411Packed  
    Spinnaker, 290  
PixelFormat\_YUV422\_8  
    Spinnaker, 294  
PixelFormat\_YUV422\_8\_UYVY  
    Spinnaker, 294  
PixelFormat\_YUV422Packed  
    Spinnaker, 290  
PixelFormat\_YUV444Packed  
    Spinnaker, 290  
PixelFormat\_YUV8\_UYV  
    Spinnaker, 294  
Spinnaker, 294  
PixelFormatEnums  
    Spinnaker, 289  
PixelFormatInfoID  
    Camera, 535  
PixelFormatInfoSelector  
    Camera, 535  
PixelFormatInfoSelector\_B10  
    Spinnaker, 297  
PixelFormatInfoSelector\_B12  
    Spinnaker, 297  
PixelFormatInfoSelector\_B16  
    Spinnaker, 297  
PixelFormatInfoSelector\_B8  
    Spinnaker, 297  
PixelFormatInfoSelector\_BayerBG10  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerBG10p  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerBG12  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerBG12p  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerBG16  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerBG8  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerGB10  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerGB10p  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerGB12  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGB12p  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGB16  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGB8  
    Spinnaker, 295  
PixelFormatInfoSelector\_BayerGR10  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGR10p  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGR12  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGR12p  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGR16  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerGR8  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerRG10  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerRG10p  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerRG12  
    Spinnaker, 296  
PixelFormatInfoSelector\_BayerRG12p  
    Spinnaker, 296

- Spinnaker, 296
- PixelFormatInfoSelector\_BayerRG16  
    Spinnaker, 296
- PixelFormatInfoSelector\_BayerRG8  
    Spinnaker, 296
- PixelFormatInfoSelector\_BayerRGPolarized10p  
    Spinnaker, 300
- PixelFormatInfoSelector\_BayerRGPolarized12p  
    Spinnaker, 300
- PixelFormatInfoSelector\_BayerRGPolarized16  
    Spinnaker, 300
- PixelFormatInfoSelector\_BayerRGPolarized8  
    Spinnaker, 300
- PixelFormatInfoSelector\_BGR10  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR12  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR12p  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR14  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR16  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR565p  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGR8  
    Spinnaker, 297
- PixelFormatInfoSelector\_BGRa10  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa10p  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa12  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa12p  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa14  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa16  
    Spinnaker, 296
- PixelFormatInfoSelector\_BGRa8  
    Spinnaker, 296
- PixelFormatInfoSelector\_BiColorBGRG10  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorBGRG10p  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorBGRG12  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorBGRG12p  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorBGRG8  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorRGBG10  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorRGBG10p  
    Spinnaker, 298
- PixelFormatInfoSelector\_BiColorRGBG12  
    Spinnaker, 298
- Spinnaker, 298
- PixelFormatInfoSelector\_BiColorRGBG12p  
    Spinnaker, 298
- Spinnaker, 298
- PixelFormatInfoSelector\_BiColorRGBG8  
    Spinnaker, 298
- Spinnaker, 298
- PixelFormatInfoSelector\_Confidence1  
    Spinnaker, 298
- PixelFormatInfoSelector\_Confidence16  
    Spinnaker, 298
- PixelFormatInfoSelector\_Confidence1p  
    Spinnaker, 298
- PixelFormatInfoSelector\_Confidence32f  
    Spinnaker, 298
- PixelFormatInfoSelector\_Confidence8  
    Spinnaker, 298
- PixelFormatInfoSelector\_Coord3D\_A10p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_A12p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_A16  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_A32f  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_A8  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC10p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC12p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC12p\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC16  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC32f  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_ABC8  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC10p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC12p  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC16  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC16\_Planar  
    Spinnaker, 297
- PixelFormatInfoSelector\_Coord3D\_AC32f

Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_AC8  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_AC8\_Planar  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_B10p  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_B12p  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_B16  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_B32f  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_B8  
    Spinnaker, 297  
PixelFormatInfoSelector\_Coord3D\_C10p  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_C12p  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_C16  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_C32f  
    Spinnaker, 298  
PixelFormatInfoSelector\_Coord3D\_C8  
    Spinnaker, 298  
PixelFormatInfoSelector\_G10  
    Spinnaker, 297  
PixelFormatInfoSelector\_G12  
    Spinnaker, 297  
PixelFormatInfoSelector\_G16  
    Spinnaker, 297  
PixelFormatInfoSelector\_G8  
    Spinnaker, 297  
PixelFormatInfoSelector\_JPEGColor8  
    Spinnaker, 300  
PixelFormatInfoSelector\_JPEGMono8  
    Spinnaker, 300  
PixelFormatInfoSelector\_LLCBayerRG8  
    Spinnaker, 300  
PixelFormatInfoSelector\_LLCMono8  
    Spinnaker, 300  
PixelFormatInfoSelector\_Mono10  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono10p  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono12  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono12p  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono14  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono16  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono16s  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono1p

Spinnaker, 295  
PixelFormatInfoSelector\_Mono2p  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono32f  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono4p  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono8  
    Spinnaker, 295  
PixelFormatInfoSelector\_Mono8s  
    Spinnaker, 295  
PixelFormatInfoSelector\_Polarized10p  
    Spinnaker, 300  
PixelFormatInfoSelector\_Polarized12p  
    Spinnaker, 300  
PixelFormatInfoSelector\_Polarized16  
    Spinnaker, 300  
PixelFormatInfoSelector\_Polarized8  
    Spinnaker, 300  
PixelFormatInfoSelector\_R10  
    Spinnaker, 297  
PixelFormatInfoSelector\_R12  
    Spinnaker, 297  
PixelFormatInfoSelector\_R16  
    Spinnaker, 297  
PixelFormatInfoSelector\_R8  
    Spinnaker, 297  
PixelFormatInfoSelector\_RGB10  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB10\_Planar  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB10p  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB10p32  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB12  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB12\_Planar  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB12p  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB14  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB16  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB16\_Planar  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB16s  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB32f  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB565p  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB8  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGB8\_Planar  
    Spinnaker, 296  
PixelFormatInfoSelector\_RGBa10

- Spinnaker, 296
- PixelFormatInfoSelector\_RGBa10p
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa12
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa12p
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa14
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa16
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa32f
  - Spinnaker, 296
- PixelFormatInfoSelector\_RGBa8
  - Spinnaker, 296
- PixelFormatInfoSelector\_SCF1WBWG10
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WBWG10p
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WBWG12
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WBWG12p
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WBWG14
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WBWG16
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB10
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB10p
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB12
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB12p
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB14
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB16
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWB8
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WGWR10
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR10p
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR12
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR12p
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR14
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR16
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WGWR8
  - Spinnaker, 298
- PixelFormatInfoSelector\_SCF1WRWG10
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG10p
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG12
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG12p
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG14
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG16
  - Spinnaker, 299
- PixelFormatInfoSelector\_SCF1WRWG8
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr10\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr10p\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr12\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr12p\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr411\_8
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_10
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_10p
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_12
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_12p
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_8
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrYY
  - Spinnaker, 299
- PixelFormatInfoSelector\_YCbCr601\_422\_10
  - Spinnaker, 300
- PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY
  - Spinnaker, 300

Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_10p  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12p  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_8  
     Spinnaker, 299  
 PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr  
     Spinnaker, 299  
 PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_10  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_10p  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_12  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_12p  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_8  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YCbCr8  
     Spinnaker, 299  
 PixelFormatInfoSelector\_YCbCr8\_CbYCr  
     Spinnaker, 299  
 PixelFormatInfoSelector\_YUV411\_8\_UYYYVYY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YUV422\_8  
     Spinnaker, 300  
 Spinnaker, 300  
 PixelFormatInfoSelector\_YUV422\_8\_UYYVYY  
     Spinnaker, 300  
 PixelFormatInfoSelector\_YUV8\_UYV  
     Spinnaker, 300  
 PixelFormatInfoSelectorEnums  
     Spinnaker, 295  
 PixelFormatIntType  
     Spinnaker, 301  
 PixelFormatNamespaceID  
     Spinnaker, 301  
 PixelFormatToSet  
     GigEVisionPerformance.cpp, 1356  
 PixelSize  
     Camera, 536  
 PixelSize\_Bpp1  
     Spinnaker, 302  
 PixelSize\_Bpp10  
     Spinnaker, 302  
 PixelSize\_Bpp12  
     Spinnaker, 302  
 PixelSize\_Bpp14  
     Spinnaker, 302  
 PixelSize\_Bpp16  
     Spinnaker, 302  
 PixelSize\_Bpp20  
     Spinnaker, 302  
 PixelSize\_Bpp24  
     Spinnaker, 302  
 PixelSize\_Bpp30  
     Spinnaker, 302  
 PixelSize\_Bpp32  
     Spinnaker, 302  
 PixelSize\_Bpp36  
     Spinnaker, 302  
 PixelSize\_Bpp4  
     Spinnaker, 302  
 PixelSize\_Bpp48  
     Spinnaker, 302  
 PixelSize\_Bpp64  
     Spinnaker, 302  
 PixelSize\_Bpp8  
     Spinnaker, 302  
 PixelSize\_Bpp96  
     Spinnaker, 302  
 PixelSizeEnums  
     Spinnaker, 302  
 PMEMBERFUNC  
     Member\_NodeCallback< Client, Member >, 979  
 PNG  
     Spinnaker, 283  
 PNGOption, 1008  
     compressionLevel, 1009  
     interlaced, 1009  
     PNGOption, 1009  
     reserved, 1009

POEStatus  
    TransportLayerInterface, 1084  
POEStatus\_NotSupported  
    Spinnaker, 302  
POEStatus\_PowerOff  
    Spinnaker, 302  
POEStatus\_PowerOn  
    Spinnaker, 302  
POEStatusEnum  
    Spinnaker, 302  
Pointer Class, 158  
    CBasePtr, 159  
    CBooleanPtr, 159  
    CCategoryPtr, 160  
    CChunkPortPtr, 160  
    CCommandPtr, 160  
    CDeviceInfoPtr, 160  
    CEnumEntryPtr, 160  
    CEnumerationPtr, 160  
    CIIntegerPtr, 161  
    CNodeMapDynPtr, 161  
    CNodeMapPtr, 161  
    CNodePtr, 161  
    CPortConstructPtr, 161  
    CPortPtr, 161  
    CPortRecorderPtr, 162  
    CPortReplayPtr, 162  
    CPortWriteListPtr, 162  
    CRegisterPtr, 162  
    CSelectorPtr, 162  
    CStringPtr, 162  
    CValuePtr, 163  
    GetInterfaceName, 163  
    IsAvailable, 163  
    IsImplemented, 163  
    IsReadable, 163  
    IsWritable, 163  
Polarization.cpp  
    AcquireImages, 1381  
    ConfigureStream, 1381  
    CreateAndSaveAolpDolpImages, 1382  
    CreateAndSaveGlareReducedImage, 1382  
    CreateAndSaveStokesImages, 1382  
    CreateHeatmapImages, 1382  
    CreateNormalizedImage, 1382  
    ExtractAndSavePolarQuadImages, 1382  
    GetQuadFileNameAppendage, 1383  
    isPixelFormatColor, 1383  
    main, 1383  
    PrintDeviceInfo, 1383  
    RunSingleCamera, 1383  
    SaveImage, 1383  
PolarizationQuadrant  
    ImageUtilityPolarization, 907  
Poll  
    NodeMap, 1001  
    Spinnaker::GenApi, 375  
PopulateAdapterInfo  
    AdapterConfig, 186  
PortImpl Class, 164  
PortNode, 1010  
    ~PortNode, 1012  
    CacheChunkData, 1012  
    GetChunkID, 1012  
    GetPortHandle, 1013  
    GetSwapEndianess, 1013  
    PortNode, 1012  
    Read, 1013  
    Replay, 1013  
    SetPortImpl, 1013  
    SetReference, 1014  
    StartRecording, 1015  
    StopRecording, 1015  
    Write, 1015  
PortNode Class, 165  
PortRecorder, 1016  
    ~PortRecorder, 1017  
    GetAccessMode, 1017  
    PortRecorder, 1017  
    Read, 1018  
    Replay, 1018  
    SetReference, 1018  
    StartRecording, 1018  
    StopRecording, 1019  
    Write, 1019  
PortRecorder Class, 166  
PortReplay, 1020  
    ~PortReplay, 1021  
    GetAccessMode, 1021  
    GetPortReplayHandle, 1022  
    PortReplay, 1021  
    Read, 1022  
    Replay, 1022  
    SetReference, 1022  
    Write, 1023  
PortReplay Class, 167  
PortWriteList Class, 168  
PowerSupplyCurrent  
    Camera, 536  
PowerSupplyVoltage  
    Camera, 536  
PPM  
    Spinnaker, 282  
PPMOption, 1023  
    binaryFile, 1024  
    PPMOption, 1024  
    reserved, 1024  
Prefix  
    U3V\_COMMAND\_HEADER, 1100  
Preprocess  
    CNodeMapFactory, 664  
PreprocessXMLFromFile  
    Spinnaker::GenApi, 375  
PreprocessXMLFromZIPFile  
    Spinnaker::GenApi, 376  
PrintAllNodes

GigEVisionPerformance.cpp, 1353  
**PrintApplicationLayerDeviceInfo**  
 GenTLInfo\_QuickSpin.cpp, 1348  
**PrintBooleanNode**  
 NodeMapInfo.cpp, 1378  
**PrintBuildInfo**  
 HighDynamicRange.cpp, 1358  
**PrintCategoryNodeAndAllFeatures**  
 NodeMapInfo.cpp, 1378  
**PrintCommandNode**  
 NodeMapInfo.cpp, 1379  
**PrintCPUUsage**  
 GigEVisionPerformance.cpp, 1353  
**PrintDataStreamInfo**  
 GigEVisionPerformance.cpp, 1353  
**PrintDebugMessage**  
 FileAccess\_QuickSpin.cpp, 1347  
**PrintDeviceInfo**  
 Acquisition.cpp, 1282  
 AcquisitionMultipleThread.cpp, 1327  
 ActionCommand.cpp, 1329  
 BufferHandling.cpp, 1331  
 ChunkData.cpp, 1333  
 CounterAndTimer.cpp, 1335  
 DeviceEvents.cpp, 1337  
 Exposure.cpp, 1343  
 Exposure\_QuickSpin.cpp, 1344  
 FileAccess\_QuickSpin.cpp, 1347  
 GigEVisionPerformance.cpp, 1353  
 HighDynamicRange.cpp, 1358  
 ImageEvents.cpp, 1361  
 ImageFormatControl.cpp, 1362  
 ImageFormatControl\_QuickSpin.cpp, 1364  
 Inference.cpp, 1369  
 LogicBlock.cpp, 1373  
 LookupTable.cpp, 1374  
 NodeMapCallback.cpp, 1376  
 Polarization.cpp, 1383  
 SaveToAvi.cpp, 1385  
 Sequencer.cpp, 1387  
 SerialRxTx.cpp, 1390  
 Trigger.cpp, 1393  
 Trigger\_QuickSpin.cpp, 1396  
**PrintEnumerationNodeAndCurrentEntry**  
 NodeMapInfo.cpp, 1379  
**PrintEnumerationSelector**  
 NodeMapInfo.cpp, 1379  
**PrintExampleStatistics**  
 AcquisitionMultipleCameraRecovery.cpp, 1325  
**PrintFloatNode**  
 NodeMapInfo.cpp, 1379  
**PrintGenericHandlerMessage**  
 InterfaceEventHandlerImpl, 943  
**PrintIntegerNode**  
 NodeMapInfo.cpp, 1379  
**PrintNode**  
 NodeMapInfo.cpp, 1379  
**PrintResultMessage**  
 FileAccess\_QuickSpin.cpp, 1347  
**PrintRetrieveNodeFailure**  
 LookupTable.cpp, 1375  
 Sequencer.cpp, 1387  
**PrintStringNode**  
 NodeMapInfo.cpp, 1380  
**PrintTransportLayerDeviceInfo**  
 GenTLInfo\_QuickSpin.cpp, 1349  
**PrintTransportLayerInterfaceInfo**  
 GenTLInfo\_QuickSpin.cpp, 1349  
**PrintTransportLayerStreamInfo**  
 GenTLInfo\_QuickSpin.cpp, 1349  
**PrintUsage**  
 FileAccess\_QuickSpin.cpp, 1347  
 GigEVisionPerformance.cpp, 1353  
**PrintValueNode**  
 NodeMapInfo.cpp, 1380  
**progressive**  
 JPEGOption, 964  
**PureNumber**  
 Spinnaker::GenApi, 351  
**QUADRANT\_I0**  
 ImageUtilityPolarization, 907  
**QUADRANT\_I135**  
 ImageUtilityPolarization, 907  
**QUADRANT\_I45**  
 ImageUtilityPolarization, 907  
**QUADRANT\_I90**  
 ImageUtilityPolarization, 907  
**quality**  
 JPEGOption, 964  
 JPG2Option, 965  
 MJPGOption, 981  
**QueryInterface**  
 Enumeration.cpp, 1338  
 Enumeration\_QuickSpin.cpp, 1339  
**radius**  
 InferenceBoxCircle, 918  
**RAW**  
 Spinnaker, 283  
**rdbuf**  
 IDevFileStreamBase< CharType, Traits >, 814  
 ODevFileStreamBase< CharType, Traits >, 1005  
**Read**  
 CChunkPort, 603  
 CEventPort, 624  
 CPortImpl, 686  
 CRegisterPortImpl, 693  
 CTestPortStruct< CDataStruct >, 700  
 PortNode, 1013  
 PortRecorder, 1018  
 PortReplay, 1022  
**read**  
 FileProtocolAdapter, 749  
**ReadPort**  
 CameraBase, 572  
 ICameraBase, 790

ReadRegister  
    CRegisterPortImpl, 693

readType  
    NodeMapInfo.cpp, 1378

receiveBuffers  
    AdapterInfo, 397

receiveBuffersMax  
    AdapterInfo, 397

receiveBuffersMin  
    AdapterInfo, 398

receiveBuffersRegKey  
    AdapterInfo, 398

receiveBuffersStep  
    AdapterInfo, 398

rect  
    InferenceBoundingBox, 915

RED  
    Spinnaker, 313

Reference Interfaces, 169

RefreshCameraList  
    AcquisitionMultipleCameraRecovery.cpp, 1325

RegionDestination  
    Camera, 536

RegionDestination\_Stream0  
    Spinnaker, 303

RegionDestination\_Stream1  
    Spinnaker, 303

RegionDestination\_Stream2  
    Spinnaker, 303

RegionDestinationEnums  
    Spinnaker, 303

RegionMode  
    Camera, 536

RegionMode\_Off  
    Spinnaker, 303

RegionMode\_On  
    Spinnaker, 303

RegionModeEnums  
    Spinnaker, 303

RegionSelector  
    Camera, 537

RegionSelector\_All  
    Spinnaker, 303

RegionSelector\_Region0  
    Spinnaker, 303

RegionSelector\_Region1  
    Spinnaker, 303

RegionSelector\_Region2  
    Spinnaker, 303

RegionSelectorEnums  
    Spinnaker, 303

Register  
    Spinnaker::GenApi, 376

RegisterAllInterfaceEvents  
    SystemEventHandlerImpl, 1058

RegisterCallback  
    Node, 992  
    Spinnaker::GenApi, 376

RegisterEventHandler  
    CameraBase, 572, 573  
    ICameraBase, 790  
    IInterface, 844  
    Interface, 932  
    ISystem, 958  
    System, 1048

RegisterImageEventHandler  
    IDataStream, 811

RegisterInterfaceEventHandler  
    ISystem, 958  
    System, 1048

RegisterInterfaceEventToSystem  
    SystemEventHandlerImpl, 1058

RegisterLoggingEventHandler  
    ISystem, 958  
    System, 1048

RegisterNode, 1024  
    ~RegisterNode, 1026  
    Get, 1026  
    GetAddress, 1028  
    GetLength, 1028  
    RegisterNode, 1026  
    Set, 1028  
    SetReference, 1028

RegisterNode Class, 170

RegisterPortImpl Class, 171

Release  
    IImage, 832  
    Image, 876

ReleaseCameraDescriptionFileData  
    CNodeMapFactory, 664

ReleaseImage  
    IDataStream, 811

ReleaseInstance  
    ISystem, 958  
    System, 1049

RemoveByDeviceID  
    CameraList, 580  
    ICameraList, 796

RemoveByIndex  
    CameraList, 581  
    ICameraList, 796

RemoveBySerial  
    CameraList, 581  
    ICameraList, 796

ReplaceEnvironmentVariables  
    Spinnaker::GenICam, 390

Replay  
    CPortImpl, 686  
    CPortWriteList, 689  
    PortNode, 1013  
    PortRecorder, 1018  
    PortReplay, 1022  
    Spinnaker::GenApi, 377

ReqId  
    GVCP\_REQUEST\_HEADER, 782  
    U3V\_COMMAND\_HEADER, 1100

Reserved  
 U3V\_EVENT\_DATA, 1101

reserved  
 AVIOption, 402  
 BMPOption, 407  
 H264Option, 783  
 JPEGOption, 964  
 JPG2Option, 965  
 MJPGOption, 981  
 PGMOption, 1008  
 PNGOption, 1009  
 PPMOption, 1024  
 TIFFOption, 1062

ReservedOrEventSize  
 GVCP\_EVENT\_ITEM, 773  
 GVCP\_EVENT\_ITEM\_BASIC, 775  
 GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776

ResetCallbacks  
 NodeMapCallback.cpp, 1376

ResetCameraUserSetToDefault  
 AcquisitionMultipleCameraRecovery.cpp, 1325

ResetDeviceEvents  
 DeviceEvents.cpp, 1337

ResetExposure  
 Exposure.cpp, 1343  
 Exposure\_QuickSpin.cpp, 1344  
 LogicBlock.cpp, 1373

ResetImage  
 IImage, 832  
 Image, 877

ResetImageEvents  
 ImageEvents.cpp, 1361

ResetLookupTables  
 LookupTable.cpp, 1375

ResetSequencer  
 Sequencer.cpp, 1387

ResetStatistics  
 CTestPortStruct< CDataStruct >, 700

ResetTrigger  
 BufferHandling.cpp, 1331  
 CounterAndTimer.cpp, 1335  
 LogicBlock.cpp, 1373  
 Trigger.cpp, 1393  
 Trigger\_QuickSpin.cpp, 1396

resize  
 gcstring, 769

Restore  
 CSelectorSet, 696  
 Spinnaker::GenApi, 377

RetrieveAllAdapters  
 AdapterConfig, 187

ReverseX  
 Camera, 537

ReverseY  
 Camera, 537

RevokelImages  
 IDataStream, 811

RgbTransformLightSource  
 Camera, 537  
 Spinnaker, 304  
 RgbTransformLightSource\_Cloudy6500K  
 Spinnaker, 304  
 RgbTransformLightSource\_CoolFluorescent4000K  
 Spinnaker, 304  
 RgbTransformLightSource\_Custom  
 Spinnaker, 304  
 RgbTransformLightSource\_Daylight5000K  
 Spinnaker, 304  
 RgbTransformLightSource\_General  
 Spinnaker, 304  
 RgbTransformLightSource\_Shade8000K  
 Spinnaker, 304  
 RgbTransformLightSource\_Tungsten2800K  
 Spinnaker, 304  
 RgbTransformLightSource\_WarmFluorescent3000K  
 Spinnaker, 304  
 RgbTransformLightSourceEnums  
 Spinnaker, 303  
 RIGOROUS  
 Spinnaker, 246  
 RO  
 Spinnaker::GenApi, 347  
 rotatedRect  
 InferenceBoundingBox, 915  
 rotationAngle  
 InferenceBoxRotatedRect, 920  
 RunMultipleCameras  
 AcquisitionMultipleThread.cpp, 1327  
 ActionCommand.cpp, 1329  
 RunSingleCamera  
 Acquisition.cpp, 1282  
 BufferHandling.cpp, 1331  
 ChunkData.cpp, 1334  
 CounterAndTimer.cpp, 1335  
 DeviceEvents.cpp, 1337  
 Exposure.cpp, 1343  
 Exposure\_QuickSpin.cpp, 1344  
 GigEVisionPerformance.cpp, 1353  
 HighDynamicRange.cpp, 1358  
 ImageEvents.cpp, 1361  
 ImageFormatControl.cpp, 1363  
 ImageFormatControl\_QuickSpin.cpp, 1364  
 Inference.cpp, 1369  
 LogicBlock.cpp, 1373  
 LookupTable.cpp, 1375  
 NodeMapCallback.cpp, 1377  
 NodeMapInfo.cpp, 1380  
 Polarization.cpp, 1383  
 SaveToAvi.cpp, 1385  
 Sequencer.cpp, 1387  
 SerialRxTx.cpp, 1390  
 Trigger.cpp, 1394  
 Trigger\_QuickSpin.cpp, 1396  
 RW  
 Spinnaker::GenApi, 347  
 SATURATION  
 Spinnaker, 313

Saturation  
    Camera, 537

SaturationEnable  
    Camera, 538

Save  
    IImage, 832–834  
    Image, 878–880

SavelImage  
    Polarization.cpp, 1383

SaveToAvi.cpp  
    AcquireImages, 1385  
    chosenVideoType, 1385  
    H264, 1384  
    main, 1385  
    MJPG, 1384  
    PrintDeviceInfo, 1385  
    RunSingleCamera, 1385  
    SaveVectorToVideo, 1385  
    UNCOMPRESSED, 1384  
    videoType, 1384

SaveVectorToVideo  
    SaveToAvi.cpp, 1385

Scan3dAxisMax  
    Camera, 538

Scan3dAxisMin  
    Camera, 538

Scan3dCoordinateOffset  
    Camera, 538

Scan3dCoordinateReferenceSelector  
    Camera, 538

Scan3dCoordinateReferenceSelector\_RotationX  
    Spinnaker, 304

Scan3dCoordinateReferenceSelector\_RotationY  
    Spinnaker, 304

Scan3dCoordinateReferenceSelector\_RotationZ  
    Spinnaker, 304

Scan3dCoordinateReferenceSelector\_TranslationX  
    Spinnaker, 304

Scan3dCoordinateReferenceSelector\_TranslationY  
    Spinnaker, 304

Scan3dCoordinateReferenceSelector\_TranslationZ  
    Spinnaker, 304

Scan3dCoordinateReferenceSelectorEnums  
    Spinnaker, 304

Scan3dCoordinateReferenceValue  
    Camera, 538

Scan3dCoordinateScale  
    Camera, 539

Scan3dCoordinateSelector  
    Camera, 539

Scan3dCoordinateSelector\_CoordinateA  
    Spinnaker, 305

Scan3dCoordinateSelector\_CoordinateB  
    Spinnaker, 305

Scan3dCoordinateSelector\_CoordinateC  
    Spinnaker, 305

Scan3dCoordinateSelectorEnums  
    Spinnaker, 304

Scan3dCoordinateSystem  
    Camera, 539

Scan3dCoordinateSystem\_Cartesian  
    Spinnaker, 305

Scan3dCoordinateSystem\_Cylindrical  
    Spinnaker, 305

Scan3dCoordinateSystem\_Spherical  
    Spinnaker, 305

Scan3dCoordinateSystemEnums  
    Spinnaker, 305

Scan3dCoordinateSystemReference  
    Camera, 539

Scan3dCoordinateSystemReference\_Anchor  
    Spinnaker, 305

Scan3dCoordinateSystemReference\_Transformed  
    Spinnaker, 305

Scan3dCoordinateSystemReferenceEnums  
    Spinnaker, 305

Scan3dCoordinateTransformSelector  
    Camera, 539

Scan3dCoordinateTransformSelector\_RotationX  
    Spinnaker, 306

Scan3dCoordinateTransformSelector\_RotationY  
    Spinnaker, 306

Scan3dCoordinateTransformSelector\_RotationZ  
    Spinnaker, 306

Scan3dCoordinateTransformSelector\_TranslationX  
    Spinnaker, 306

Scan3dCoordinateTransformSelector\_TranslationY  
    Spinnaker, 306

Scan3dCoordinateTransformSelector\_TranslationZ  
    Spinnaker, 306

Scan3dCoordinateTransformSelectorEnums  
    Spinnaker, 305

Scan3dDistanceUnit  
    Camera, 539

Scan3dDistanceUnit\_Inch  
    Spinnaker, 306

Scan3dDistanceUnit\_Millimeter  
    Spinnaker, 306

Scan3dDistanceUnitEnums  
    Spinnaker, 306

Scan3dInvalidDataFlag  
    Camera, 540

Scan3dInvalidDataValue  
    Camera, 540

Scan3dOutputMode  
    Camera, 540

Scan3dOutputMode\_CalibratedABC\_Grid  
    Spinnaker, 306

Scan3dOutputMode\_CalibratedABC\_PointCloud  
    Spinnaker, 306

Scan3dOutputMode\_CalibratedAC  
    Spinnaker, 307

Scan3dOutputMode\_CalibratedAC\_Linescan  
    Spinnaker, 307

Scan3dOutputMode\_CalibratedC  
    Spinnaker, 307

Scan3dOutputMode\_CalibratedC\_Linescan  
    Spinnaker, 307

Scan3dOutputMode\_DisparityC  
    Spinnaker, 307

Scan3dOutputMode\_DisparityC\_Linescan  
    Spinnaker, 307

Scan3dOutputMode\_RectifiedC  
    Spinnaker, 307

Scan3dOutputMode\_RectifiedC\_Linescan  
    Spinnaker, 307

Scan3dOutputMode\_UncalibratedC  
    Spinnaker, 306

Scan3dOutputModeEnums  
    Spinnaker, 306

Scan3dTransformValue  
    Camera, 540

SecondsCounter, 189  
    endTime, 190  
    GetSecondsCounter, 190  
    StartSecondsCounter, 190  
    startTime, 190  
    timeDiff, 190

SelectorSet Class, 172

SendActionCommand  
    IInterface, 844  
    Interface, 932  
    ISystem, 959  
    System, 1049

SensorDescription  
    Camera, 540

SensorDigitizationTaps  
    Camera, 540

SensorDigitizationTaps\_Eight  
    Spinnaker, 307

SensorDigitizationTaps\_Four  
    Spinnaker, 307

SensorDigitizationTaps\_One  
    Spinnaker, 307

SensorDigitizationTaps\_Ten  
    Spinnaker, 307

SensorDigitizationTaps\_Three  
    Spinnaker, 307

SensorDigitizationTaps\_Two  
    Spinnaker, 307

SensorDigitizationTapsEnums  
    Spinnaker, 307

SensorHeight  
    Camera, 541

SensorShutterMode  
    Camera, 541

SensorShutterMode\_Global  
    Spinnaker, 308

SensorShutterMode\_GlobalReset  
    Spinnaker, 308

SensorShutterMode\_Rolling  
    Spinnaker, 308

SensorShutterModeEnums  
    Spinnaker, 307

SensorTaps  
    Camera, 541

SensorTaps\_Eight  
    Spinnaker, 308

SensorTaps\_Four  
    Spinnaker, 308

SensorTaps\_One  
    Spinnaker, 308

SensorTaps\_Ten  
    Spinnaker, 308

SensorTaps\_Three  
    Spinnaker, 308

SensorTaps\_Two  
    Spinnaker, 308

SensorTapsEnums  
    Spinnaker, 308

SensorWidth  
    Camera, 541

Sequencer.cpp  
    AcquireImages, 1386  
    ConfigureSequencerPartOne, 1386  
    ConfigureSequencerPartTwo, 1386  
    main, 1387  
    PrintDeviceInfo, 1387  
    PrintRetrieveNodeFailure, 1387  
    ResetSequencer, 1387  
    RunSingleCamera, 1387  
    SetSingleState, 1387

SequencerConfigurationMode  
    Camera, 541

SequencerConfigurationMode\_Off  
    Spinnaker, 308

SequencerConfigurationMode\_On  
    Spinnaker, 308

SequencerConfigurationModeEnums  
    Spinnaker, 308

SequencerConfigurationValid  
    Camera, 541

SequencerConfigurationValid\_No  
    Spinnaker, 309

SequencerConfigurationValid\_Yes  
    Spinnaker, 309

SequencerConfigurationValidEnums  
    Spinnaker, 309

SequencerFeatureEnable  
    Camera, 542

SequencerMode  
    Camera, 542

SequencerMode\_Off  
    Spinnaker, 309

SequencerMode\_On  
    Spinnaker, 309

SequencerModeEnums  
    Spinnaker, 309

SequencerPathSelector  
    Camera, 542

SequencerSetActive  
    Camera, 542

SequencerSetLoad  
    Camera, 543  
SequencerSetNext  
    Camera, 543  
SequencerSetSave  
    Camera, 543  
SequencerSetSelector  
    Camera, 543  
SequencerSetStart  
    Camera, 543  
SequencerSetValid  
    Camera, 544  
SequencerSetValid\_No  
    Spinnaker, 309  
SequencerSetValid\_Yes  
    Spinnaker, 309  
SequencerSetValidEnums  
    Spinnaker, 309  
SequencerTriggerActivation  
    Camera, 544  
SequencerTriggerActivation\_AnyEdge  
    Spinnaker, 310  
SequencerTriggerActivation\_FallingEdge  
    Spinnaker, 310  
SequencerTriggerActivation\_LevelHigh  
    Spinnaker, 310  
SequencerTriggerActivation\_LevelLow  
    Spinnaker, 310  
SequencerTriggerActivation\_RisingEdge  
    Spinnaker, 310  
SequencerTriggerActivationEnums  
    Spinnaker, 309  
SequencerTriggerSource  
    Camera, 544  
SequencerTriggerSource\_FrameStart  
    Spinnaker, 310  
SequencerTriggerSource\_Off  
    Spinnaker, 310  
SequencerTriggerSourceEnums  
    Spinnaker, 310  
SIGNAL\_PORT\_BAUD\_RATE  
    SerialRxTx.cpp, 1389  
SIGNAL\_PORT\_COMMUNICATION\_TIMEOUT\_MILLISECONDS  
    SerialRxTx.cpp, 1389  
SIGNAL\_PORT\_DELAY  
    SerialRxTx.cpp, 1389  
SIGNAL\_PORT\_PARITY\_BITS  
    SerialRxTx.cpp, 1389  
SIGNAL\_PORT\_STOP\_BITS  
    SerialRxTx.cpp, 1389  
SerialPortBaudRate  
    Camera, 544  
SerialPortBaudRate\_Baud115200  
    Spinnaker, 310  
SerialPortBaudRate\_Baud1200  
    Spinnaker, 310  
SerialPortBaudRate\_Baud14400  
    Spinnaker, 310  
SerialPortBaudRate\_Baud19200  
    Spinnaker, 310  
SerialPortBaudRate\_Baud230400  
    Spinnaker, 310  
SerialPortBaudRate\_Baud2400  
    Spinnaker, 310  
SerialPortBaudRate\_Baud300  
    Spinnaker, 310  
SerialPortBaudRate\_Baud38400  
    Spinnaker, 310  
SerialPortBaudRate\_Baud460800  
    Spinnaker, 310  
SerialPortBaudRate\_Baud4800  
    Spinnaker, 310  
SerialPortBaudRate\_Baud57600  
    Spinnaker, 310  
SerialPortBaudRate\_Baud600  
    Spinnaker, 310  
SerialPortBaudRate\_Baud921600  
    Spinnaker, 310  
SerialPortBaudRate\_Baud9600  
    Spinnaker, 310  
SerialPortBaudRateEnums  
    Spinnaker, 310  
SerialPortDataBits  
    Camera, 545  
SerialPortParity  
    Camera, 545  
SerialPortParity\_Even  
    Spinnaker, 311  
SerialPortParity\_Mark  
    Spinnaker, 311  
SerialPortParity\_None  
    Spinnaker, 311  
SerialPortParity\_Odd  
    Spinnaker, 311  
SerialPortParity\_Space  
    Spinnaker, 311  
SerialPortParityEnums  
    Spinnaker, 311  
SerialPortSelector  
    Camera, 545  
SerialPortSelector\_SerialPort0  
    Spinnaker, 311  
SerialPortSelectorEnums  
    Spinnaker, 311  
SerialPortSource  
    Camera, 545  
SerialPortSource\_Line0  
    Spinnaker, 311  
SerialPortSource\_Line1  
    Spinnaker, 311  
SerialPortSource\_Line2  
    Spinnaker, 311  
SerialPortSource\_Line3  
    Spinnaker, 311  
SerialPortSource\_Off  
    Spinnaker, 311

SerialPortSourceEnums  
     Spinnaker, 311

SerialPortStopBits  
     Camera, 545

SerialPortStopBits\_Bits1  
     Spinnaker, 312

SerialPortStopBits\_Bits1AndAHalf  
     Spinnaker, 312

SerialPortStopBits\_Bits2  
     Spinnaker, 312

SerialPortStopBitsEnums  
     Spinnaker, 312

SerialReceiveFramingErrorCount  
     Camera, 545

SerialReceiveParityErrorCount  
     Camera, 546

SerialReceiveQueueClear  
     Camera, 546

SerialReceiveQueueCurrentCharacterCount  
     Camera, 546

SerialReceiveQueueMaxCharacterCount  
     Camera, 546

SerialRx  
     SerialRxTx.cpp, 1390

SerialRxTx.cpp  
     CleanUp, 1390  
     COM\_PORT\_COUNT\_MAX, 1388  
     ConfigureDevice, 1390  
     DATA\_BITS, 1389  
     main, 1390  
     MILLISECOND, 1389  
     PrintDeviceInfo, 1390  
     RunSingleCamera, 1390  
     SERIAL\_PORT baud RATE, 1389  
     SERIAL\_PORT COMMUNICATION\_TIMEOUT\_MILLISECOND, 1389  
     SERIAL\_PORT\_DELAY, 1389  
     SERIAL\_PORT\_PARITY\_BITS, 1389  
     SERIAL\_PORT\_STOP\_BITS, 1389  
     SerialRx, 1390  
     SerialTx, 1391  
     TWO\_SECOND\_DELAY, 1389

SerialTransmitQueueCurrentCharacterCount  
     Camera, 546

SerialTransmitQueueMaxCharacterCount  
     Camera, 546

SerialTx  
     SerialRxTx.cpp, 1391

Set  
     RegisterNode, 1028

SET\_GUID  
     Spinnaker::GenApi, 377

SetBufferOwnership  
     CameraBase, 573  
     ICameraBase, 791

SetChannelStatus  
     IImageStatistics, 840  
     ImageStatistics, 896

SetChunkEnable  
     Inference.cpp, 1369

SetChunks  
     ChunkData, 647  
     IChunkData, 805

SetCookie  
     CPortWriteList, 689  
     Spinnaker::GenApi, 377

SetDefaultColorProcessing  
     Image, 880

SetEnumReference  
     CEnumerationTRef< EnumT >, 608

SetEventPayload  
     EventHandler, 738

SetEventType  
     EventHandler, 738

SetFirst  
     CSelectorSet, 696

SetFrameRate  
     GigEVisionPerformance.cpp, 1354

SetGenICamCacheFolder  
     Spinnaker::GenICam, 390

SetGenICamCLProtocolFolder  
     Spinnaker::GenICam, 390

SetGenICamLogConfig  
     Spinnaker::GenICam, 390

SetHeatmapColorGradient  
     ImageUtilityHeatmap, 905

SetHeatmapRange  
     ImageUtilityHeatmap, 905

SetInfo  
     CFeatureBag, 627

SetIntValue  
     EnumNode, 730

SetLoggingEventPriorityLevel  
     ISystem, 959  
     System, 1050

SetMaximumFileSize  
     SpinVideo, 1035

SetMessageCallback  
     SpinUpdate.h, 1271

SetNext  
     CSelectorSet, 696  
     Spinnaker::GenApi, 378

SetNodeHandle  
     Node, 992

SetNodeMap  
     Node, 992

SetNumEnums  
     CEnumerationTRef< EnumT >, 608  
     Spinnaker::GenApi, 378

SetPortImpl  
     CChunkPort, 603  
     CEventPort, 624  
     CPortImpl, 686  
     CRegisterPortImpl, 693  
     PortNode, 1013

SetProgressCallback  
    SpinUpdate.h, 1271

SetReference  
    BooleanNode, 410  
    CategoryNode, 586  
    CEnumerationTRef< EnumT >, 608  
    CommandNode, 677  
    EnumEntryNode, 725  
    EnumNode, 730  
    FloatNode, 756  
    FloatRegNode, 759  
    IntegerNode, 928  
    IntRegNode, 953  
    Node, 992  
    PortNode, 1014  
    PortRecorder, 1018  
    PortReplay, 1022  
    RegisterNode, 1028  
    StringNode, 1039  
    StringRegNode, 1042  
    ValueNode, 1105

SetSingleState  
    Sequencer.cpp, 1387

SetupCounterAndTimer  
    CounterAndTimer.cpp, 1336

SetUserBuffers  
    CameraBase, 573, 574  
    ICameraBase, 791

SetValue  
    BooleanNode, 410  
    CEnumerationTRef< EnumT >, 609  
    FloatNode, 756  
    IntegerNode, 928  
    StringNode, 1039

Sharpening  
    Camera, 547

SharpeningAuto  
    Camera, 547

SharpeningEnable  
    Camera, 547

SharpeningThreshold  
    Camera, 547

Signed  
    Spinnaker::GenApi, 351

SingleChunkData\_t, 1029  
    ChunkID, 1029  
    ChunkLength, 1029  
    ChunkOffset, 1029

SingleChunkDataStr\_t, 1030  
    ChunkID, 1030  
    ChunkLength, 1030  
    ChunkOffset, 1030

size  
    double\_vector\_t, 714  
    gcstring, 769  
    int64\_vector\_t, 922

SleepyWrapper  
    AcquisitionMultipleCameraRecovery.cpp, 1325

ActionCommand.cpp, 1329  
BufferHandling.cpp, 1331  
ImageEvents.cpp, 1361

SOFTWARE  
    Trigger.cpp, 1393  
    Trigger\_QuickSpin.cpp, 1395

SoftwareSignalPulse  
    Camera, 548

SoftwareSignalSelector  
    Camera, 548

SoftwareSignalSelector\_SoftwareSignal0  
    Spinnaker, 312

SoftwareSignalSelector\_SoftwareSignal1  
    Spinnaker, 312

SoftwareSignalSelector\_SoftwareSignal2  
    Spinnaker, 312

SoftwareSignalSelectorEnums  
    Spinnaker, 312

SourceCount  
    Camera, 548

SourceDataRange  
    ImageUtility, 898

SourceSelector  
    Camera, 548

SourceSelector\_All  
    Spinnaker, 312

SourceSelector\_Source0  
    Spinnaker, 312

SourceSelector\_Source1  
    Spinnaker, 312

SourceSelector\_Source2  
    Spinnaker, 312

SourceSelectorEnums  
    Spinnaker, 312

SPECIFIC  
    DeviceEvents.cpp, 1337

Spinnaker, 191  
    AcquisitionMode\_Continuous, 230  
    AcquisitionMode\_MultiFrame, 230  
    AcquisitionMode\_SingleFrame, 230  
    AcquisitionModeEnums, 230  
    AcquisitionStatusSelector\_AcquisitionActive, 230  
    AcquisitionStatusSelector\_AcquisitionTransfer, 231  
    AcquisitionStatusSelector\_AcquisitionTriggerWait,  
        230  
    AcquisitionStatusSelector\_ExposureActive, 231  
    AcquisitionStatusSelector\_FrameActive, 231  
    AcquisitionStatusSelector\_FrameTriggerWait, 231  
    AcquisitionStatusSelectorEnums, 230  
    ACTION\_COMMAND\_STATUS\_ACTION\_LATE,  
        231  
    ACTION\_COMMAND\_STATUS\_ERROR, 231  
    ACTION\_COMMAND\_STATUS\_NO\_REF\_TIME,  
        231  
    ACTION\_COMMAND\_STATUS\_OK, 231  
    ACTION\_COMMAND\_STATUS\_OVERFLOW, 231  
    ActionCommandStatus, 231  
    ActionUnconditionalMode\_Off, 231

ActionUnconditionalMode\_On, 231  
ActionUnconditionalModeEnums, 231  
AdcBitDepth\_Bit10, 232  
AdcBitDepth\_Bit12, 232  
AdcBitDepth\_Bit14, 232  
AdcBitDepth\_Bit8, 232  
AdcBitDepthEnums, 231  
AutoAlgorithmSelector\_Ae, 232  
AutoAlgorithmSelector\_Awb, 232  
AutoAlgorithmSelectorEnums, 232  
AutoExposureControlPriority\_ExposureTime, 232  
AutoExposureControlPriority\_Gain, 232  
AutoExposureControlPriorityEnums, 232  
AutoExposureLightingMode\_AutoDetect, 233  
AutoExposureLightingMode\_Backlight, 233  
AutoExposureLightingMode\_Frontlight, 233  
AutoExposureLightingMode\_Normal, 233  
AutoExposureLightingModeEnums, 232  
AutoExposureMeteringMode\_Average, 233  
AutoExposureMeteringMode\_CenterWeighted, 233  
AutoExposureMeteringMode\_HistogramPeak, 233  
AutoExposureMeteringMode\_Partial, 233  
AutoExposureMeteringMode\_Spot, 233  
AutoExposureMeteringModeEnums, 233  
AutoExposureTargetGreyValueAuto\_Continuous, 233  
AutoExposureTargetGreyValueAuto\_Off, 233  
AutoExposureTargetGreyValueAutoEnums, 233  
BalanceRatioSelector\_Blue, 234  
BalanceRatioSelector\_Red, 234  
BalanceRatioSelectorEnums, 234  
BalanceWhiteAuto\_Continuous, 234  
BalanceWhiteAuto\_Off, 234  
BalanceWhiteAuto\_Once, 234  
BalanceWhiteAutoEnums, 234  
BalanceWhiteAutoProfile\_Indoor, 234  
BalanceWhiteAutoProfile\_Outdoor, 234  
BalanceWhiteAutoProfileEnums, 234  
BILINEAR, 246  
BinningHorizontalMode\_Average, 235  
BinningHorizontalMode\_Sum, 235  
BinningHorizontalModeEnums, 235  
BinningSelector\_All, 235  
BinningSelector\_ISP, 235  
BinningSelector\_Sensor, 235  
BinningSelectorEnums, 235  
BinningVerticalMode\_Average, 235  
BinningVerticalMode\_Sum, 235  
BinningVerticalModeEnums, 235  
BlackLevelAuto\_Continuous, 236  
BlackLevelAuto\_Off, 236  
BlackLevelAuto\_Once, 236  
BlackLevelAutoBalance\_Continuous, 236  
BlackLevelAutoBalance\_Off, 236  
BlackLevelAutoBalance\_Once, 236  
BlackLevelAutoBalanceEnums, 236  
BlackLevelAutoEnums, 236  
BlackLevelSelector\_All, 236  
BlackLevelSelector\_Analog, 236  
BlackLevelSelector\_Digital, 236  
BlackLevelSelectorEnums, 236  
BLUE, 313  
BMP, 282  
BUFFER\_OWNERSHIP\_SYSTEM, 237  
BUFFER\_OWNERSHIP\_USER, 237  
BufferOwnership, 237  
ChunkBlackLevelSelector\_All, 237  
ChunkBlackLevelSelectorEnums, 237  
ChunkCounterSelector\_Counter0, 237  
ChunkCounterSelector\_Counter1, 237  
ChunkCounterSelector\_Counter2, 237  
ChunkCounterSelectorEnums, 237  
ChunkEncoderSelector\_Encoder0, 238  
ChunkEncoderSelector\_Encoder1, 238  
ChunkEncoderSelector\_Encoder2, 238  
ChunkEncoderSelectorEnums, 237  
ChunkEncoderStatus\_EncoderDown, 238  
ChunkEncoderStatus\_EncoderIdle, 238  
ChunkEncoderStatus\_EncoderStatic, 238  
ChunkEncoderStatus\_EncoderUp, 238  
ChunkEncoderStatusEnums, 238  
ChunkExposureTimeSelector\_Blue, 238  
ChunkExposureTimeSelector\_Common, 238  
ChunkExposureTimeSelector\_Cyan, 238  
ChunkExposureTimeSelector\_Green, 238  
ChunkExposureTimeSelector\_Infrared, 238  
ChunkExposureTimeSelector\_Magenta, 238  
ChunkExposureTimeSelector\_Red, 238  
ChunkExposureTimeSelector\_Stage1, 238  
ChunkExposureTimeSelector\_Stage2, 238  
ChunkExposureTimeSelector\_Ultraviolet, 238  
ChunkExposureTimeSelector\_Yellow, 238  
ChunkExposureTimeSelectorEnums, 238  
ChunkGainSelector\_All, 239  
ChunkGainSelector\_Blue, 239  
ChunkGainSelector\_Green, 239  
ChunkGainSelector\_Red, 239  
ChunkGainSelectorEnums, 238  
ChunkImageComponent\_Color, 239  
ChunkImageComponent\_Confidence, 239  
ChunkImageComponent\_Disparity, 239  
ChunkImageComponent\_Infrared, 239  
ChunkImageComponent\_Intensity, 239  
ChunkImageComponent\_Range, 239  
ChunkImageComponent\_Scatter, 239  
ChunkImageComponent\_Ultraviolet, 239  
ChunkImageComponentEnums, 239  
ChunkPixelFormat\_BayerBG8, 240  
ChunkPixelFormat\_BayerGB8, 240  
ChunkPixelFormat\_BayerGR8, 240  
ChunkPixelFormat\_BayerRG8, 240  
ChunkPixelFormat\_Mono12Packed, 239  
ChunkPixelFormat\_Mono16, 239  
ChunkPixelFormat\_Mono8, 239  
ChunkPixelFormat\_RGB8Packed, 239

ChunkPixelFormat\_YCbCr601\_422\_8\_CbYCrY, 240  
ChunkPixelFormat\_YUV422Packed, 240  
ChunkPixelFormatEnums, 239  
ChunkRegionID\_Region0, 240  
ChunkRegionID\_Region1, 240  
ChunkRegionID\_Region2, 240  
ChunkRegionIDEnums, 240  
ChunkScan3dCoordinateReferenceSelector\_RotationX, 240  
ChunkScan3dCoordinateReferenceSelector\_RotationY, 240  
ChunkScan3dCoordinateReferenceSelector\_RotationZ, 240  
ChunkScan3dCoordinateReferenceSelector\_TranslationX, 240  
ChunkScan3dCoordinateReferenceSelector\_TranslationY, 240  
ChunkScan3dCoordinateReferenceSelector\_TranslationZ, 240  
ChunkScan3dCoordinateReferenceSelectorEnums, 240  
ChunkScan3dCoordinateSelector\_CoordinateA, 241  
ChunkScan3dCoordinateSelector\_CoordinateB, 241  
ChunkScan3dCoordinateSelector\_CoordinateC, 241  
ChunkScan3dCoordinateSelectorEnums, 240  
ChunkScan3dCoordinateSystem\_Cartesian, 241  
ChunkScan3dCoordinateSystem\_Cylindrical, 241  
ChunkScan3dCoordinateSystem\_Spherical, 241  
ChunkScan3dCoordinateSystemEnums, 241  
ChunkScan3dCoordinateSystemReference\_Anchor, 241  
ChunkScan3dCoordinateSystemReference\_Transformed, 241  
ChunkScan3dCoordinateSystemReferenceEnums, 241  
ChunkScan3dCoordinateTransformSelector\_RotationX, 242  
ChunkScan3dCoordinateTransformSelector\_RotationY, 242  
ChunkScan3dCoordinateTransformSelector\_RotationZ, 242  
ChunkScan3dCoordinateTransformSelector\_TranslationX, 242  
ChunkScan3dCoordinateTransformSelector\_TranslationY, 242  
ChunkScan3dCoordinateTransformSelector\_TranslationZ, 242  
ChunkScan3dCoordinateTransformSelectorEnums, 241  
ChunkScan3dDistanceUnit\_Inch, 242  
ChunkScan3dDistanceUnit\_Millimeter, 242  
ChunkScan3dDistanceUnitEnums, 242  
ChunkScan3dOutputMode\_CalibratedABC\_Grid, 242  
ChunkScan3dOutputMode\_CalibratedABC\_PointCloud, 242  
ChunkScan3dOutputMode\_CalibratedAC, 243  
ChunkScan3dOutputMode\_CalibratedAC\_Linescan, 243  
ChunkScan3dOutputMode\_CalibratedC, 243  
ChunkScan3dOutputMode\_CalibratedC\_Linescan, 243  
ChunkScan3dOutputMode\_DisparityC, 243  
ChunkScan3dOutputMode\_DisparityC\_Linescan, 243  
ChunkScan3dOutputMode\_RectifiedC, 243  
ChunkScan3dOutputMode\_RectifiedC\_Linescan, 243  
ChunkScan3dOutputMode\_UncalibratedC, 242  
ChunkScan3dOutputModeEnums, 242  
ChunkSelector\_BlackLevel, 244  
ChunkSelector\_CRC, 243  
ChunkSelector\_ExposureEndLineStatusAll, 244  
ChunkSelector\_ExposureTime, 243  
ChunkSelector\_FrameID, 243  
ChunkSelector\_Gain, 244  
ChunkSelector\_Height, 243  
ChunkSelector\_Image, 243  
ChunkSelector\_OffsetX, 243  
ChunkSelector\_OffsetY, 243  
ChunkSelector\_PixelFormat, 244  
ChunkSelector\_SequencerSetActive, 244  
ChunkSelector\_SerialData, 244  
ChunkSelector\_Timestamp, 244  
ChunkSelector\_Width, 243  
ChunkSelectorEnums, 243  
ChunkSourceID\_Source0, 244  
ChunkSourceID\_Source1, 244  
ChunkSourceID\_Source2, 244  
ChunkSourceIDEnums, 244  
ChunkTimerSelector\_Timer0, 244  
ChunkTimerSelector\_Timer1, 244  
ChunkTimerSelector\_Timer2, 244  
ChunkTimerSelectorEnums, 244  
ChunkTransferStreamID\_Stream0, 245  
ChunkTransferStreamID\_Stream1, 245  
ChunkTransferStreamID\_Stream2, 245  
ChunkTransferStreamID\_Stream3, 245  
ChunkTransferStreamIDEnums, 244  
CIConfiguration\_Base, 245  
CIConfiguration\_DualBase, 245  
CIConfiguration\_EightyBit, 245  
CIConfiguration\_Full, 245  
CIConfiguration\_Medium, 245  
CIConfigurationEnums, 245  
CITimeSlotsCount\_One, 245  
CITimeSlotsCount\_Three, 245  
CITimeSlotsCount\_Two, 245  
CITimeSlotsCountEnums, 245  
ColorProcessingAlgorithm, 245  
ColorTransformationSelector\_RGBtoRGB, 246  
ColorTransformationSelector\_RGBtoYUV, 246

ColorTransformationSelectorEnums, 246  
ColorTransformationValueSelector\_Gain00, 247  
ColorTransformationValueSelector\_Gain01, 247  
ColorTransformationValueSelector\_Gain02, 247  
ColorTransformationValueSelector\_Gain10, 247  
ColorTransformationValueSelector\_Gain11, 247  
ColorTransformationValueSelector\_Gain12, 247  
ColorTransformationValueSelector\_Gain20, 247  
ColorTransformationValueSelector\_Gain21, 247  
ColorTransformationValueSelector\_Gain22, 247  
ColorTransformationValueSelector\_Offset0, 247  
ColorTransformationValueSelector\_Offset1, 247  
ColorTransformationValueSelector\_Offset2, 247  
ColorTransformationValueSelectorEnums, 246  
CounterEventActivation\_AnyEdge, 247  
CounterEventActivation\_FallingEdge, 247  
CounterEventActivation\_LevelHigh, 247  
CounterEventActivation\_LevelLow, 247  
CounterEventActivation\_RisingEdge, 247  
CounterEventActivationEnums, 247  
CounterEventSource\_Counter0End, 248  
CounterEventSource\_Counter0Start, 248  
CounterEventSource\_Counter1End, 248  
CounterEventSource\_Counter1Start, 248  
CounterEventSource\_ExposureEnd, 248  
CounterEventSource\_ExposureStart, 248  
CounterEventSource\_FrameTriggerWait, 248  
CounterEventSource\_Line0, 247  
CounterEventSource\_Line1, 247  
CounterEventSource\_Line2, 247  
CounterEventSource\_Line3, 247  
CounterEventSource\_LogicBlock0, 248  
CounterEventSource\_LogicBlock1, 248  
CounterEventSource\_MHzTick, 247  
CounterEventSource\_Off, 247  
CounterEventSource\_UserOutput0, 248  
CounterEventSource\_UserOutput1, 248  
CounterEventSource\_UserOutput2, 248  
CounterEventSource\_UserOutput3, 248  
CounterEventSourceEnums, 247  
CounterResetActivation\_AnyEdge, 248  
CounterResetActivation\_FallingEdge, 248  
CounterResetActivation\_LevelHigh, 248  
CounterResetActivation\_LevelLow, 248  
CounterResetActivation\_RisingEdge, 248  
CounterResetActivationEnums, 248  
CounterResetSource\_Counter0End, 249  
CounterResetSource\_Counter0Start, 249  
CounterResetSource\_Counter1End, 249  
CounterResetSource\_Counter1Start, 249  
CounterResetSource\_ExposureEnd, 249  
CounterResetSource\_ExposureStart, 249  
CounterResetSource\_FrameTriggerWait, 249  
CounterResetSource\_Line0, 248  
CounterResetSource\_Line1, 248  
CounterResetSource\_Line2, 248  
CounterResetSource\_Line3, 249  
CounterResetSource\_LogicBlock0, 249  
CounterResetSource\_LogicBlock1, 249  
CounterResetSource\_Off, 248  
CounterResetSource\_UserOutput0, 249  
CounterResetSource\_UserOutput1, 249  
CounterResetSource\_UserOutput2, 249  
CounterResetSource\_UserOutput3, 249  
CounterResetSourceEnums, 248  
CounterSelector\_Counter0, 249  
CounterSelector\_Counter1, 249  
CounterSelectorEnums, 249  
CounterStatus\_CounterActive, 249  
CounterStatus\_CounterCompleted, 249  
CounterStatus\_CounterIdle, 249  
CounterStatus\_CounterOverflow, 249  
CounterStatus\_CounterTriggerWait, 249  
CounterStatusEnums, 249  
CounterTriggerActivation\_AnyEdge, 250  
CounterTriggerActivation\_FallingEdge, 250  
CounterTriggerActivation\_LevelHigh, 250  
CounterTriggerActivation\_LevelLow, 250  
CounterTriggerActivation\_RisingEdge, 250  
CounterTriggerActivationEnums, 250  
CounterTriggerSource\_Counter0End, 250  
CounterTriggerSource\_Counter0Start, 250  
CounterTriggerSource\_Counter1End, 250  
CounterTriggerSource\_Counter1Start, 250  
CounterTriggerSource\_ExposureEnd, 250  
CounterTriggerSource\_ExposureStart, 250  
CounterTriggerSource\_FrameTriggerWait, 250  
CounterTriggerSource\_Line0, 250  
CounterTriggerSource\_Line1, 250  
CounterTriggerSource\_Line2, 250  
CounterTriggerSource\_Line3, 250  
CounterTriggerSource\_LogicBlock0, 250  
CounterTriggerSource\_LogicBlock1, 250  
CounterTriggerSource\_Off, 250  
CounterTriggerSource\_UserOutput0, 250  
CounterTriggerSource\_UserOutput1, 250  
CounterTriggerSource\_UserOutput2, 250  
CounterTriggerSource\_UserOutput3, 250  
CounterTriggerSourceEnums, 250  
CxpConnectionTestMode\_Mode1, 251  
CxpConnectionTestMode\_Off, 251  
CxpConnectionTestModeEnums, 251  
CxpLinkConfiguration\_Auto, 251  
CxpLinkConfiguration\_CXP1\_X1, 251  
CxpLinkConfiguration\_CXP1\_X2, 251  
CxpLinkConfiguration\_CXP1\_X3, 251  
CxpLinkConfiguration\_CXP1\_X4, 251  
CxpLinkConfiguration\_CXP1\_X5, 252  
CxpLinkConfiguration\_CXP1\_X6, 252  
CxpLinkConfiguration\_CXP2\_X1, 251  
CxpLinkConfiguration\_CXP2\_X2, 251  
CxpLinkConfiguration\_CXP2\_X3, 251  
CxpLinkConfiguration\_CXP2\_X4, 251  
CxpLinkConfiguration\_CXP2\_X5, 252  
CxpLinkConfiguration\_CXP2\_X6, 252  
CxpLinkConfiguration\_CXP3\_X1, 251

CxpLinkConfiguration\_CXP3\_X2, 251  
CxpLinkConfiguration\_CXP3\_X3, 251  
CxpLinkConfiguration\_CXP3\_X4, 251  
CxpLinkConfiguration\_CXP3\_X5, 252  
CxpLinkConfiguration\_CXP3\_X6, 252  
CxpLinkConfiguration\_CXP5\_X1, 251  
CxpLinkConfiguration\_CXP5\_X2, 251  
CxpLinkConfiguration\_CXP5\_X3, 251  
CxpLinkConfiguration\_CXP5\_X4, 251  
CxpLinkConfiguration\_CXP5\_X5, 252  
CxpLinkConfiguration\_CXP5\_X6, 252  
CxpLinkConfiguration\_CXP6\_X1, 251  
CxpLinkConfiguration\_CXP6\_X2, 251  
CxpLinkConfiguration\_CXP6\_X3, 251  
CxpLinkConfiguration\_CXP6\_X4, 251  
CxpLinkConfiguration\_CXP6\_X5, 252  
CxpLinkConfiguration\_CXP6\_X6, 252  
CxpLinkConfigurationEnums, 251  
CxpLinkConfigurationPreferred\_CXP1\_X1, 252  
CxpLinkConfigurationPreferred\_CXP1\_X2, 252  
CxpLinkConfigurationPreferred\_CXP1\_X3, 252  
CxpLinkConfigurationPreferred\_CXP1\_X4, 252  
CxpLinkConfigurationPreferred\_CXP1\_X5, 252  
CxpLinkConfigurationPreferred\_CXP1\_X6, 253  
CxpLinkConfigurationPreferred\_CXP2\_X1, 252  
CxpLinkConfigurationPreferred\_CXP2\_X2, 252  
CxpLinkConfigurationPreferred\_CXP2\_X3, 252  
CxpLinkConfigurationPreferred\_CXP2\_X4, 252  
CxpLinkConfigurationPreferred\_CXP2\_X5, 252  
CxpLinkConfigurationPreferred\_CXP2\_X6, 253  
CxpLinkConfigurationPreferred\_CXP3\_X1, 252  
CxpLinkConfigurationPreferred\_CXP3\_X2, 252  
CxpLinkConfigurationPreferred\_CXP3\_X3, 252  
CxpLinkConfigurationPreferred\_CXP3\_X4, 252  
CxpLinkConfigurationPreferred\_CXP3\_X5, 252  
CxpLinkConfigurationPreferred\_CXP3\_X6, 253  
CxpLinkConfigurationPreferred\_CXP5\_X1, 252  
CxpLinkConfigurationPreferred\_CXP5\_X2, 252  
CxpLinkConfigurationPreferred\_CXP5\_X3, 252  
CxpLinkConfigurationPreferred\_CXP5\_X4, 252  
CxpLinkConfigurationPreferred\_CXP5\_X5, 252  
CxpLinkConfigurationPreferred\_CXP5\_X6, 253  
CxpLinkConfigurationPreferred\_CXP6\_X1, 252  
CxpLinkConfigurationPreferred\_CXP6\_X2, 252  
CxpLinkConfigurationPreferred\_CXP6\_X3, 252  
CxpLinkConfigurationPreferred\_CXP6\_X4, 252  
CxpLinkConfigurationPreferred\_CXP6\_X5, 253  
CxpLinkConfigurationPreferred\_CXP6\_X6, 253  
CxpLinkConfigurationPreferredEnums, 252  
CxpLinkConfigurationStatus\_CXP1\_X1, 253  
CxpLinkConfigurationStatus\_CXP1\_X2, 253  
CxpLinkConfigurationStatus\_CXP1\_X3, 253  
CxpLinkConfigurationStatus\_CXP1\_X4, 253  
CxpLinkConfigurationStatus\_CXP1\_X5, 253  
CxpLinkConfigurationStatus\_CXP1\_X6, 254  
CxpLinkConfigurationStatus\_CXP2\_X1, 253  
CxpLinkConfigurationStatus\_CXP2\_X2, 253  
CxpLinkConfigurationStatus\_CXP2\_X3, 253  
CxpLinkConfigurationStatus\_CXP2\_X4, 253  
CxpLinkConfigurationStatus\_CXP2\_X5, 253  
CxpLinkConfigurationStatus\_CXP2\_X6, 254  
CxpLinkConfigurationStatus\_CXP5\_X1, 253  
CxpLinkConfigurationStatus\_CXP5\_X2, 253  
CxpLinkConfigurationStatus\_CXP5\_X3, 253  
CxpLinkConfigurationStatus\_CXP5\_X4, 253  
CxpLinkConfigurationStatus\_CXP5\_X5, 253  
CxpLinkConfigurationStatus\_CXP5\_X6, 254  
CxpLinkConfigurationStatus\_CXP6\_X1, 253  
CxpLinkConfigurationStatus\_CXP6\_X2, 253  
CxpLinkConfigurationStatus\_CXP6\_X3, 253  
CxpLinkConfigurationStatus\_CXP6\_X4, 253  
CxpLinkConfigurationStatus\_CXP6\_X5, 254  
CxpLinkConfigurationStatus\_CXP6\_X6, 254  
CxpLinkConfigurationStatus\_None, 253  
CxpLinkConfigurationStatus\_Pending, 253  
CxpLinkConfigurationStatusEnums, 253  
CxpPoCxpStatus\_Auto, 254  
CxpPoCxpStatus\_Off, 254  
CxpPoCxpStatus\_Tripped, 254  
CxpPoCxpStatusEnums, 254  
DecimationHorizontalMode\_Discard, 254  
DecimationHorizontalModeEnums, 254  
DecimationSelector\_All, 255  
DecimationSelector\_Sensor, 255  
DecimationSelectorEnums, 254  
DecimationVerticalMode\_Discard, 255  
DecimationVerticalModeEnums, 255  
DEFAULT, 246  
DefectCorrectionMode\_Average, 255  
DefectCorrectionMode\_Highlight, 255  
DefectCorrectionMode\_Zero, 255  
DefectCorrectionModeEnums, 255  
Deinterlacing\_LineDuplication, 256  
Deinterlacing\_Off, 256  
Deinterlacing\_Weave, 256  
DeinterlacingEnums, 255  
DEPRECATED\_CLASS, 327  
DeviceAccessStatus\_Busy, 256  
DeviceAccessStatus\_NoAccess, 256  
DeviceAccessStatus\_OpenReadOnly, 256  
DeviceAccessStatus\_OpenReadWrite, 256  
DeviceAccessStatus\_ReadOnly, 256  
DeviceAccessStatus\_ReadWrite, 256  
DeviceAccessStatus\_Unknown, 256  
DeviceAccessStatusEnum, 256  
DeviceCharacterSet\_ASCII, 256  
DeviceCharacterSet\_UTF8, 256  
DeviceCharacterSetEnums, 256  
DeviceClockSelector\_CameraLink, 257  
DeviceClockSelector\_Sensor, 257

DeviceClockSelector\_SensorDigitization, 257  
DeviceClockSelectorEnums, 256  
DeviceConnectionStatus\_Active, 257  
DeviceConnectionStatus\_Inactive, 257  
DeviceConnectionStatusEnums, 257  
DeviceCurrentSpeed\_FullSpeed, 257  
DeviceCurrentSpeed\_HighSpeed, 257  
DeviceCurrentSpeed\_LowSpeed, 257  
DeviceCurrentSpeed\_SuperSpeed, 257  
DeviceCurrentSpeed\_UnknownSpeed, 257  
DeviceCurrentSpeedEnum, 257  
DeviceEndianessMechanism\_Legacy, 258  
DeviceEndianessMechanism\_Standard, 258  
DeviceEndianessMechanismEnum, 257  
DeviceIndicatorMode\_Active, 258  
DeviceIndicatorMode\_ErrorStatus, 258  
DeviceIndicatorMode\_Inactive, 258  
DeviceIndicatorModeEnums, 258  
DeviceLinkHeartbeatMode\_Off, 258  
DeviceLinkHeartbeatMode\_On, 258  
DeviceLinkHeartbeatModeEnums, 258  
DeviceLinkThroughputLimitMode\_Off, 259  
DeviceLinkThroughputLimitMode\_On, 259  
DeviceLinkThroughputLimitModeEnums, 258  
DevicePowerSupplySelector\_External, 259  
DevicePowerSupplySelectorEnums, 259  
DeviceRegistersEndianness\_Big, 259  
DeviceRegistersEndianness\_Little, 259  
DeviceRegistersEndiannessEnums, 259  
DeviceScanType\_Areascan, 259  
DeviceScanTypeEnums, 259  
DeviceSerialPortBaudRate\_Baud115200, 260  
DeviceSerialPortBaudRate\_Baud19200, 260  
DeviceSerialPortBaudRate\_Baud230400, 260  
DeviceSerialPortBaudRate\_Baud38400, 260  
DeviceSerialPortBaudRate\_Baud460800, 260  
DeviceSerialPortBaudRate\_Baud57600, 260  
DeviceSerialPortBaudRate\_Baud921600, 260  
DeviceSerialPortBaudRate\_Baud9600, 260  
DeviceSerialPortBaudRateEnums, 260  
DeviceSerialPortSelector\_CameraLink, 260  
DeviceSerialPortSelectorEnums, 260  
DeviceStreamChannelEndianness\_Big, 260  
DeviceStreamChannelEndianness\_Little, 260  
DeviceStreamChannelEndiannessEnums, 260  
DeviceStreamChannelType\_Receiver, 261  
DeviceStreamChannelType\_Transmitter, 261  
DeviceStreamChannelTypeEnums, 261  
DeviceTapGeometry\_Geometry\_10X, 262  
DeviceTapGeometry\_Geometry\_10X\_1Y, 262  
DeviceTapGeometry\_Geometry\_1X, 261  
DeviceTapGeometry\_Geometry\_1X10, 262  
DeviceTapGeometry\_Geometry\_1X10\_1Y, 262  
DeviceTapGeometry\_Geometry\_1X2, 261  
DeviceTapGeometry\_Geometry\_1X2\_1Y, 261  
DeviceTapGeometry\_Geometry\_1X2\_1Y2, 261  
DeviceTapGeometry\_Geometry\_1X2\_2YE, 262  
DeviceTapGeometry\_Geometry\_1X3, 261  
DeviceTapGeometry\_Geometry\_1X3\_1Y, 261  
DeviceTapGeometry\_Geometry\_1X4, 262  
DeviceTapGeometry\_Geometry\_1X4\_1Y, 261  
DeviceTapGeometry\_Geometry\_1X8, 262  
DeviceTapGeometry\_Geometry\_1X8\_1Y, 262  
DeviceTapGeometry\_Geometry\_1X\_1Y, 261  
DeviceTapGeometry\_Geometry\_1X\_1Y2, 261  
DeviceTapGeometry\_Geometry\_1X\_2YE, 261  
DeviceTapGeometry\_Geometry\_2X, 261  
DeviceTapGeometry\_Geometry\_2X2, 262  
DeviceTapGeometry\_Geometry\_2X2\_1Y, 261  
DeviceTapGeometry\_Geometry\_2X2E, 262  
DeviceTapGeometry\_Geometry\_2X2E\_1YGeometry\_2X2M\_1Y, 261  
DeviceTapGeometry\_Geometry\_2X2E\_2YE, 262  
DeviceTapGeometry\_Geometry\_2X2M, 262  
DeviceTapGeometry\_Geometry\_2X\_1Y, 261  
DeviceTapGeometry\_Geometry\_2X\_1Y2Geometry\_2XE\_1Y, 261  
DeviceTapGeometry\_Geometry\_2X\_2YE, 262  
DeviceTapGeometry\_Geometry\_2XE, 261  
DeviceTapGeometry\_Geometry\_2XE\_1Y2, 261  
DeviceTapGeometry\_Geometry\_2XE\_2YE, 262  
DeviceTapGeometry\_Geometry\_2XM, 261  
DeviceTapGeometry\_Geometry\_2XM\_1Y, 261  
DeviceTapGeometry\_Geometry\_2XM\_1Y2, 261  
DeviceTapGeometry\_Geometry\_2XM\_2YE, 262  
DeviceTapGeometry\_Geometry\_3X, 261  
DeviceTapGeometry\_Geometry\_3X\_1Y, 261  
DeviceTapGeometry\_Geometry\_4X, 262  
DeviceTapGeometry\_Geometry\_4X2, 262  
DeviceTapGeometry\_Geometry\_4X2\_1Y, 262  
DeviceTapGeometry\_Geometry\_4X2E, 262  
DeviceTapGeometry\_Geometry\_4X2E\_1Y, 262  
DeviceTapGeometry\_Geometry\_4X\_1Y, 261  
DeviceTapGeometry\_Geometry\_8X, 262  
DeviceTapGeometry\_Geometry\_8X\_1Y, 262  
DeviceTapGeometryEnums, 261  
DeviceTemperatureSelector\_Sensor, 262  
DeviceTemperatureSelectorEnums, 262  
DeviceTLType\_CameraLink, 263  
DeviceTLType\_CameraLinkHS, 263  
DeviceTLType\_CoaXPress, 263  
DeviceTLType\_Custom, 263  
DeviceTLType\_GigEVision, 263  
DeviceTLType\_USB3Vision, 263  
DeviceTLTypeEnums, 262  
DeviceType\_CameraLink, 263  
DeviceType\_CameraLinkHS, 263  
DeviceType\_CoaXPress, 263  
DeviceType\_Custom, 263  
DeviceType\_GigEVision, 263  
DeviceType\_Peripheral, 263  
DeviceType\_Receiver, 263  
DeviceType\_Transceiver, 263  
DeviceType\_Transmitter, 263  
DeviceType\_USB3Vision, 263  
DeviceTypeEnum, 263

DeviceTypeEnums, 263  
DIRECTIONAL\_FILTER, 246  
EDGE\_SENSING, 246  
EncoderMode\_FourPhase, 264  
EncoderMode\_HighResolution, 264  
EncoderModeEnums, 263  
EncoderOutputMode\_DirectionDown, 264  
EncoderOutputMode\_DirectionUp, 264  
EncoderOutputMode\_Motion, 264  
EncoderOutputMode\_Off, 264  
EncoderOutputMode\_PositionDown, 264  
EncoderOutputMode\_PositionUp, 264  
EncoderOutputModeEnums, 264  
EncoderResetActivation\_AnyEdge, 265  
EncoderResetActivation\_FallingEdge, 265  
EncoderResetActivation\_LevelHigh, 265  
EncoderResetActivation\_LevelLow, 265  
EncoderResetActivation\_RisingEdge, 265  
EncoderResetActivationEnums, 264  
EncoderResetSource\_AcquisitionEnd, 265  
EncoderResetSource\_AcquisitionStart, 265  
EncoderResetSource\_AcquisitionTrigger, 265  
EncoderResetSource\_Action0, 266  
EncoderResetSource\_Action1, 266  
EncoderResetSource\_Action2, 266  
EncoderResetSource\_Counter0End, 265  
EncoderResetSource\_Counter0Start, 265  
EncoderResetSource\_Counter1End, 265  
EncoderResetSource\_Counter1Start, 265  
EncoderResetSource\_Counter2End, 265  
EncoderResetSource\_Counter2Start, 265  
EncoderResetSource\_ExposureEnd, 265  
EncoderResetSource\_ExposureStart, 265  
EncoderResetSource\_FrameEnd, 265  
EncoderResetSource\_FrameStart, 265  
EncoderResetSource\_FrameTrigger, 265  
EncoderResetSource\_Line0, 265  
EncoderResetSource\_Line1, 265  
EncoderResetSource\_Line2, 265  
EncoderResetSource\_LinkTrigger0, 266  
EncoderResetSource\_LinkTrigger1, 266  
EncoderResetSource\_LinkTrigger2, 266  
EncoderResetSource\_Off, 265  
EncoderResetSource\_SoftwareSignal0, 265  
EncoderResetSource\_SoftwareSignal1, 265  
EncoderResetSource\_SoftwareSignal2, 266  
EncoderResetSource\_Timer0End, 265  
EncoderResetSource\_Timer0Start, 265  
EncoderResetSource\_Timer1End, 265  
EncoderResetSource\_Timer1Start, 265  
EncoderResetSource\_Timer2End, 265  
EncoderResetSource\_Timer2Start, 265  
EncoderResetSource\_UserOutput0, 265  
EncoderResetSource\_UserOutput1, 265  
EncoderResetSource\_UserOutput2, 265  
EncoderResetSourceEnums, 265  
EncoderSelector\_Encoder0, 266  
EncoderSelector\_Encoder1, 266  
EncoderSelector\_Encoder2, 266  
EncoderSelectorEnums, 266  
EncoderSourceA\_Line0, 266  
EncoderSourceA\_Line1, 266  
EncoderSourceA\_Line2, 266  
EncoderSourceA\_Off, 266  
EncoderSourceAEnums, 266  
EncoderSourceB\_Line0, 267  
EncoderSourceB\_Line1, 267  
EncoderSourceB\_Line2, 267  
EncoderSourceB\_Off, 267  
EncoderSourceBEnums, 267  
EncoderStatus\_EncoderDown, 267  
EncoderStatus\_EncoderIdle, 267  
EncoderStatus\_EncoderStatic, 267  
EncoderStatus\_EncoderUp, 267  
EncoderStatusEnums, 267  
Error, 267  
EVENT\_TIMEOUT\_INFINITE, 329  
EVENT\_TIMEOUT\_NONE, 329  
EventNotification\_Off, 270  
EventNotification\_On, 270  
EventNotificationEnums, 268  
EventSelector\_Error, 270  
EventSelector\_ExposureEnd, 270  
EventSelector\_SerialPortReceive, 270  
EventSelectorEnums, 270  
EventType, 270  
ExposureActiveMode\_AllPixels, 271  
ExposureActiveMode\_AnyPixels, 271  
ExposureActiveMode\_Line1, 271  
ExposureActiveModeEnums, 270  
ExposureAuto\_Continuous, 271  
ExposureAuto\_Off, 271  
ExposureAuto\_Once, 271  
ExposureAutoEnums, 271  
ExposureMode\_Timed, 271  
ExposureMode\_TriggerWidth, 271  
ExposureModeEnums, 271  
ExposureTimeMode\_Common, 273  
ExposureTimeMode\_Individual, 273  
ExposureTimeModeEnums, 271  
ExposureTimeSelector\_Blue, 273  
ExposureTimeSelector\_Common, 273  
ExposureTimeSelector\_Cyan, 273  
ExposureTimeSelector\_Green, 273  
ExposureTimeSelector\_Infrared, 273  
ExposureTimeSelector\_Magenta, 273  
ExposureTimeSelector\_Red, 273  
ExposureTimeSelector\_Stage1, 273  
ExposureTimeSelector\_Stage2, 273  
ExposureTimeSelector\_Ultraviolet, 273  
ExposureTimeSelector\_Yellow, 273  
ExposureTimeSelectorEnums, 273  
FileOpenMode\_Read, 273  
FileOpenMode\_ReadWrite, 273  
FileOpenMode\_Write, 273  
FileOpenModeEnums, 273

FileOperationSelector\_Close, 274  
 FileOperationSelector\_Delete, 274  
 FileOperationSelector\_Open, 274  
 FileOperationSelector\_Read, 274  
 FileOperationSelector\_Write, 274  
 FileOperationSelectorEnums, 274  
 FileOperationStatus\_Failure, 274  
 FileOperationStatus\_Overflow, 274  
 FileOperationStatus\_Success, 274  
 FileOperationStatusEnums, 274  
 FileSelector\_SerialPort0, 274  
 FileSelector\_UserFile1, 274  
 FileSelector\_UserSet0, 274  
 FileSelector\_UserSet1, 274  
 FileSelector\_UserSetDefault, 274  
 FileSelectorEnums, 274  
 FilterDriverStatus\_Disabled, 275  
 FilterDriverStatus\_Enabled, 275  
 FilterDriverStatus\_NotSupported, 275  
 FilterDriverStatusEnum, 275  
 FROM\_FILE\_EXT, 282  
 GainAuto\_Continuous, 275  
 GainAuto\_Off, 275  
 GainAuto\_Once, 275  
 GainAutoBalance\_Continuous, 275  
 GainAutoBalance\_Off, 275  
 GainAutoBalance\_Once, 275  
 GainAutoBalanceEnums, 275  
 GainAutoEnums, 275  
 GainSelector\_All, 276  
 GainSelectorEnums, 276  
 GENICAM\_ERR\_ACCESS, 268  
 GENICAM\_ERR\_BAD\_ALLOCATION, 268  
 GENICAM\_ERR\_DYNAMIC\_CAST, 268  
 GENICAM\_ERR\_GENERIC, 268  
 GENICAM\_ERR\_INVALID\_ARGUMENT, 268  
 GENICAM\_ERR\_LOGICAL, 268  
 GENICAM\_ERR\_OUT\_OF\_RANGE, 268  
 GENICAM\_ERR\_PROPERTY, 268  
 GENICAM\_ERR\_RUN\_TIME, 268  
 GENICAM\_ERR\_TIMEOUT, 268  
 GenICamXMLLocation\_Device, 276  
 GenICamXMLLocation\_Host, 276  
 GenICamXMLLocationEnum, 276  
 GevCCP\_ControlAccess, 277  
 GevCCP\_EnumEntry\_GevCCP\_ControlAccess, 276  
 GevCCP\_EnumEntry\_GevCCP\_ExclusiveAccess, 276  
 GevCCP\_EnumEntry\_GevCCP\_OpenAccess, 276  
 GevCCP\_ExclusiveAccess, 277  
 GevCCP\_OpenAccess, 277  
 GevCCPEnum, 276  
 GevCCPEnums, 276  
 GevCurrentPhysicalLinkConfiguration\_DynamicLAG, 277  
 GevCurrentPhysicalLinkConfiguration\_MultiLink, 277  
 GevCurrentPhysicalLinkConfiguration\_SingleLink, 277  
 GevCurrentPhysicalLinkConfiguration\_StaticLAG, 277  
 GevCurrentPhysicalLinkConfigurationEnums, 277  
 GevGVCPExtendedStatusCodesSelector\_Version1\_1, 277  
 GevGVCPExtendedStatusCodesSelector\_Version2\_0, 277  
 GevGVCPExtendedStatusCodesSelectorEnums, 277  
 GevGVSPExtendedIDMode\_Off, 278  
 GevGVSPExtendedIDMode\_On, 278  
 GevGVSPExtendedIDModeEnums, 277  
 GevIEEE1588ClockAccuracy\_Unknown, 278  
 GevIEEE1588ClockAccuracyEnums, 278  
 GevIEEE1588Mode\_Auto, 278  
 GevIEEE1588Mode\_SlaveOnly, 278  
 GevIEEE1588ModeEnums, 278  
 GevIEEE1588Status\_Disabled, 278  
 GevIEEE1588Status\_Faulty, 278  
 GevIEEE1588Status\_Initializing, 278  
 GevIEEE1588Status\_Listening, 279  
 GevIEEE1588Status\_Master, 279  
 GevIEEE1588Status\_Passive, 279  
 GevIEEE1588Status\_PreMaster, 279  
 GevIEEE1588Status\_Slave, 279  
 GevIEEE1588Status\_Uncalibrated, 279  
 GevIEEE1588StatusEnums, 278  
 GevIPConfigurationStatus\_DHCP, 279  
 GevIPConfigurationStatus\_ForceIP, 279  
 GevIPConfigurationStatus\_LLA, 279  
 GevIPConfigurationStatus\_None, 279  
 GevIPConfigurationStatus\_PersistentIP, 279  
 GevIPConfigurationStatusEnums, 279  
 GevPhysicalLinkConfiguration\_DynamicLAG, 279  
 GevPhysicalLinkConfiguration\_MultiLink, 279  
 GevPhysicalLinkConfiguration\_SingleLink, 279  
 GevPhysicalLinkConfiguration\_StaticLAG, 279  
 GevPhysicalLinkConfigurationEnums, 279  
 GevSupportedOptionSelector\_Action, 280  
 GevSupportedOptionSelector\_CCPApplicationSocket, 280  
 GevSupportedOptionSelector\_CommandsConcatenation, 280  
 GevSupportedOptionSelector\_DiscoveryAckDelay, 280  
 GevSupportedOptionSelector\_DiscoveryAckDelayWritable, 280  
 GevSupportedOptionSelector\_Event, 280  
 GevSupportedOptionSelector\_EventData, 280  
 GevSupportedOptionSelector\_ExtendedStatusCodes, 280  
 GevSupportedOptionSelector\_HeartbeatDisable, 280  
 GevSupportedOptionSelector\_IPConfigurationDHCP, 280  
 GevSupportedOptionSelector\_IPConfigurationLLA,

280  
GevSupportedOptionSelector\_IPConfigurationPersistentIP, 281  
280  
GevSupportedOptionSelector\_LinkSpeed, 280  
GevSupportedOptionSelector\_ManifestTable, 280  
GevSupportedOptionSelector\_MessageChannelSourceSocket, 280  
280  
GevSupportedOptionSelector\_PacketResend, 280  
GevSupportedOptionSelector\_PendingAck, 280  
GevSupportedOptionSelector\_SerialNumber, 280  
GevSupportedOptionSelector\_StreamChannelSourceSocket, 280  
280  
GevSupportedOptionSelector\_TestData, 280  
GevSupportedOptionSelector\_UserDefinedName, 280  
GevSupportedOptionSelector\_WriteMem, 280  
GevSupportedOptionSelectorEnums, 279  
GREEN, 313  
GREY, 313  
GUIXMLLocation\_Device, 280  
GUIXMLLocation\_Host, 280  
GUIXMLLocationEnum, 280  
HQ\_LINEAR, 246  
HUE, 313  
IMAGE\_CHUNK\_DATA\_INVALID, 283  
IMAGE\_CRC\_CHECK\_FAILED, 283  
IMAGE\_DATA\_INCOMPLETE, 283  
IMAGE\_DATA\_OVERFLOW, 283  
IMAGE\_FILE\_FORMAT\_FORCE\_32BITS, 283  
IMAGE\_INFO\_INCONSISTENT, 283  
IMAGE\_LEADER\_BUFFER\_SIZE\_INCONSISTENT, 283  
IMAGE\_MISSING\_LEADER, 283  
IMAGE\_MISSING\_PACKETS, 283  
IMAGE\_MISSING\_TRAILER, 283  
IMAGE\_NO\_ERROR, 283  
IMAGE\_NO\_SYSTEM\_RESOURCES, 283  
IMAGE\_PACKETID\_INCONSISTENT, 283  
IMAGE\_TRAILER\_BUFFER\_SIZE\_INCONSISTENT, 283  
IMAGE\_UNKNOWN\_ERROR, 283  
ImageComponentSelector\_Color, 281  
ImageComponentSelector\_Confidence, 281  
ImageComponentSelector\_Disparity, 281  
ImageComponentSelector\_Infrared, 281  
ImageComponentSelector\_Intensity, 281  
ImageComponentSelector\_Range, 281  
ImageComponentSelector\_Scatter, 281  
ImageComponentSelector\_Ultraviolet, 281  
ImageComponentSelectorEnums, 280  
ImageCompressionJPEGFormatOption\_BaselineOptimized, 281  
ImageCompressionJPEGFormatOption\_BaselineStandard, 281  
ImageCompressionJPEGFormatOption\_Lossless, 281  
ImageCompressionJPEGFormatOption\_Progressive, 281  
ImageCompressionJPEGFormatOptionEnums, 281  
ImageCompressionMode\_Lossless, 282  
ImageCompressionMode\_Off, 282  
ImageCompressionModeEnums, 282  
ImageCompressionRateOption\_FixBitrate, 282  
ImageCompressionRateOption\_FixQuality, 282  
ImageCompressionRateOptionEnums, 282  
ImageFileFormat, 282  
ImageStatus, 283  
INFERENCE\_BOX\_TYPE\_CIRCLE, 284  
INFERENCE\_BOX\_TYPE\_RECTANGLE, 284  
INFERENCE\_BOX\_TYPE\_ROTATED\_RECTANGLE, 284  
InferenceBoxType, 283  
InterfaceType\_CameraLink, 284  
InterfaceType\_CameraLinkHS, 284  
InterfaceType\_CoaXPress, 284  
InterfaceType\_Custom, 284  
InterfaceType\_GigEVision, 284  
InterfaceType\_USB3Vision, 284  
InterfaceTypeEnum, 284  
IntType\_FLOAT32, 301  
IntType\_INT16, 301  
IntType\_INT8, 301  
IntType\_UINT10, 301  
IntType\_UINT10P, 301  
IntType\_UINT10p, 301  
IntType\_UINT12, 301  
IntType\_UINT12P, 301  
IntType\_UINT12p, 301  
IntType\_UINT14, 301  
IntType\_UINT16, 301  
IntType\_UINT8, 301  
IntType\_UNKNOWN, 301  
IPP, 246  
JPEG, 282  
JPEG12\_C, 283  
JPEG2000, 282  
LIGHTNESS, 313  
LineFormat\_LVDS, 284  
LineFormat\_NoConnect, 284  
LineFormat\_OpenDrain, 284  
LineFormat\_OptoCoupled, 284  
LineFormat\_RS422, 284  
LineFormat\_TriState, 284  
LineFormat\_TTL, 284  
LineFormatEnums, 284  
LineInputFilterSelector\_Debounce, 285  
LineInputFilterSelector\_Deglitch, 285  
LineInputFilterSelectorEnums, 285  
LineMode\_Input, 285  
LineMode\_Output, 285  
LineModeEnums, 285  
LineSelector\_Line0, 285  
LineSelector\_Line1, 285  
LineSelector\_Line2, 285  
LineSelector\_Line3, 285

LineSelectorEnums, 285  
 LineSource\_AllPixel, 286  
 LineSource\_AnyPixel, 286  
 LineSource\_Counter0Active, 286  
 LineSource\_Counter1Active, 286  
 LineSource\_ExposureActive, 286  
 LineSource\_FrameTriggerWait, 286  
 LineSource\_Line0, 286  
 LineSource\_Line1, 286  
 LineSource\_Line2, 286  
 LineSource\_Line3, 286  
 LineSource\_LogicBlock0, 286  
 LineSource\_LogicBlock1, 286  
 LineSource\_Off, 286  
 LineSource\_PPSSignal, 286  
 LineSource\_SerialPort0, 286  
 LineSource\_UserOutput0, 286  
 LineSource\_UserOutput1, 286  
 LineSource\_UserOutput2, 286  
 LineSource\_UserOutput3, 286  
 LineSourceEnums, 285  
 LOG\_LEVEL\_ALERT, 313  
 LOG\_LEVEL\_CRIT, 313  
 LOG\_LEVEL\_DEBUG, 313  
 LOG\_LEVEL\_ERROR, 313  
 LOG\_LEVEL\_FATAL, 313  
 LOG\_LEVEL\_INFO, 313  
 LOG\_LEVEL\_NOTICE, 313  
 LOG\_LEVEL\_NOTSET, 313  
 LOG\_LEVEL\_OFF, 313  
 LOG\_LEVEL\_WARN, 313  
 LogicBlockLUTInputActivation\_AnyEdge, 286  
 LogicBlockLUTInputActivation\_FallingEdge, 286  
 LogicBlockLUTInputActivation\_LevelHigh, 286  
 LogicBlockLUTInputActivation\_LevelLow, 286  
 LogicBlockLUTInputActivation\_RisingEdge, 286  
 LogicBlockLUTInputActivationEnums, 286  
 LogicBlockLUTInputSelector\_Input0, 287  
 LogicBlockLUTInputSelector\_Input1, 287  
 LogicBlockLUTInputSelector\_Input2, 287  
 LogicBlockLUTInputSelector\_Input3, 287  
 LogicBlockLUTInputSelectorEnums, 286  
 LogicBlockLUTInputSource\_AcquisitionActive, 287  
 LogicBlockLUTInputSource\_Counter0End, 287  
 LogicBlockLUTInputSource\_Counter0Start, 287  
 LogicBlockLUTInputSource\_Counter1End, 287  
 LogicBlockLUTInputSource\_Counter1Start, 287  
 LogicBlockLUTInputSource\_ExposureEnd, 287  
 LogicBlockLUTInputSource\_ExposureStart, 287  
 LogicBlockLUTInputSource\_FrameTriggerWait, 287  
 LogicBlockLUTInputSource\_Line0, 287  
 LogicBlockLUTInputSource\_Line1, 287  
 LogicBlockLUTInputSource\_Line2, 287  
 LogicBlockLUTInputSource\_Line3, 287  
 LogicBlockLUTInputSource\_LogicBlock0, 287  
 LogicBlockLUTInputSource\_LogicBlock1, 287  
 LogicBlockLUTInputSource\_UserOutput0, 287  
 LogicBlockLUTInputSource\_UserOutput1, 287  
 LogicBlockLUTInputSource\_UserOutput2, 287  
 LogicBlockLUTInputSource\_UserOutput3, 287  
 LogicBlockLUTInputSource\_Zero, 287  
 LogicBlockLUTInputSourceEnums, 287  
 LogicBlockLUTSelector\_Enable, 288  
 LogicBlockLUTSelector\_Value, 288  
 LogicBlockLUTSelectorEnums, 287  
 LogicBlockSelector\_LightLevel, 288  
 LogicBlockSelector\_LogicBlock0, 288  
 LogicBlockSelector\_LogicBlock1, 288  
 LogicBlockSelectorEnums, 288  
 LUTSelector\_LUT1, 288  
 LUTSelectorEnums, 288  
 NEAREST\_NEIGHBOR, 246  
 NEAREST\_NEIGHBOR\_AVG, 246  
 NO\_COLOR\_PROCESSING, 246  
 NUM\_ACQUISITIONMODE, 230  
 NUM\_ACQUISITIONSTATUSSELECTOR, 231  
 NUM\_ACTIONUNCONDITIONALMODE, 231  
 NUM\_ADCBITDEPTH, 232  
 NUM\_AUTOALGORITHMSELECTOR, 232  
 NUM\_AUTOEXPOSURECONTROLPRIORITY, 232  
 NUM\_AUTOEXPOSURELIGHTINGMODE, 233  
 NUM\_AUTOEXPOSUREMETERINGMODE, 233  
 NUM\_AUTOEXPOSURETARGETGREYVALUEAUTO, 233  
 NUM\_BALANCERATIOSELECTOR, 234  
 NUM\_BALANCEWHITEAUTO, 234  
 NUM\_BALANCEWHITEAUTOPROFILE, 234  
 NUM\_BINNINGHORIZONTALMODE, 235  
 NUM\_BINNINGSELECTOR, 235  
 NUM\_BINNINGVERTICALMODE, 235  
 NUM\_BLACKLEVELAUTO, 236  
 NUM\_BLACKLEVELAUTOBALANCE, 236  
 NUM\_BLACKLEVELSELECTOR, 236  
 NUM\_CHUNKBLACKLEVELSELECTOR, 237  
 NUM\_CHUNKCOUNTERSELECTOR, 237  
 NUM\_CHUNKENCODERSELECTOR, 238  
 NUM\_CHUNKENCODERSTATUS, 238  
 NUM\_CHUNKEXPOSURETIMESELECTOR, 238  
 NUM\_CHUNKGAINSELECTOR, 239  
 NUM\_CHUNKIMAGECOMPONENT, 239  
 NUM\_CHUNKPIXELFORMAT, 240  
 NUM\_CHUNKREGIONID, 240  
 NUM\_CHUNKSCAN3DCOORDINATEREferenceSELECTOR, 240  
 NUM\_CHUNKSCAN3DCOORDINATESELECTOR, 241  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEM, 241  
 NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE, 241  
 NUM\_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR, 242  
 NUM\_CHUNKSCAN3DDISTANCEUNIT, 242  
 NUM\_CHUNKSCAN3DOUTPUTMODE, 243  
 NUM\_CHUNKSELECTOR, 244

NUM\_CHUNKSOURCEID, 244  
NUM\_CHUNKTIMERSELECTOR, 244  
NUM\_CHUNKTRANSFERSTREAMID, 245  
NUM\_CLCONFIGURATION, 245  
NUM\_CLTIMESLOTSCOUNT, 245  
NUM\_COLORTRANSFORMATIONSELECTOR, 246  
NUM\_COLORTRANSFORMATIONVALUESELECTOR, 247  
NUM\_COUNTEREVENTACTIVATION, 247  
NUM\_COUNTEREVENTSOURCE, 248  
NUM\_COUNTERRESETACTIVATION, 248  
NUM\_COUNTERRESETSOURCE, 249  
NUM\_COUNTERSELECTOR, 249  
NUM\_COUNTERSTATUS, 249  
NUM\_COUNTERTRIGGERACTIVATION, 250  
NUM\_COUNTERTRIGGERSOURCE, 250  
NUM\_CXP CONNECTIONTESTMODE, 251  
NUM\_CXLINKCONFIGURATION, 252  
NUM\_CXLINKCONFIGURATIONPREFERRED, 253  
NUM\_CXLINKCONFIGURATIONSTATUS, 254  
NUM\_CXPOCXPSTATUS, 254  
NUM\_DECIMATIONHORIZONTALMODE, 254  
NUM\_DECIMATIONSELECTOR, 255  
NUM\_DECIMATIONVERTICALMODE, 255  
NUM\_DEFECTCORRECTIONMODE, 255  
NUM\_DEINTERLACING, 256  
NUM\_DEVICECHARACTERSET, 256  
NUM\_DEVICECLOCKSELECTOR, 257  
NUM\_DEVICECONNECTIONSTATUS, 257  
NUM\_DEVICEINDICATORMODE, 258  
NUM\_DEVICELINKHEARTBEATMODE, 258  
NUM\_DEVICELINKTHROUGHPUTLIMITMODE, 259  
NUM\_DEVICEPOWERSUPPLYSELECTOR, 259  
NUM\_DEVICEREGISTERSENDIАНNESS, 259  
NUM\_DEVICESCANTYPE, 259  
NUM\_DEVICESERIALPORTBAUDRATE, 260  
NUM\_DEVICESERIALPORTSELECTOR, 260  
NUM\_DEVICESTREAMCHANNELENDIАНNESS, 260  
NUM\_DEVICESTREAMCHANNELTYPE, 261  
NUM\_DEVICETAPGEOMETRY, 262  
NUM\_DEVICETEMPERATURESELECTOR, 262  
NUM\_DEVICETLTTYPE, 263  
NUM\_DEVICETYPE, 263  
NUM\_ENCODERMODE, 264  
NUM\_ENCODEROUTPUTMODE, 264  
NUM\_ENCODERRESETACTIVATION, 265  
NUM\_ENCODERRESETSOURCE, 266  
NUM\_ENCODERSELECTOR, 266  
NUM\_ENCODERSOURCEA, 266  
NUM\_ENCODERSOURCEB, 267  
NUM\_ENCODERSTATUS, 267  
NUM\_EVENTNOTIFICATION, 270  
NUM\_EVENTSELECTOR, 270  
NUM\_EXPOSUREACTIVE MODE, 271  
NUM\_EXPOSUREAUTO, 271  
NUM\_EXPOSUREMODE, 271  
NUM\_EXPOSURETIME MODE, 273  
NUM\_EXPOSURETIMESELECTOR, 273  
NUM\_FILEOPENMODE, 273  
NUM\_FILEOPERATIONSELECTOR, 274  
NUM\_FILEOPERATIONSTATUS, 274  
NUM\_FILESELECTOR, 274  
NUM\_GAINAUTO, 275  
NUM\_GAINAUTOBALANCE, 275  
NUM\_GAINSELECTOR, 276  
NUM\_GEV CCP, 277  
NUM\_GEV CURRENTPHYSICALLINKCONFIGURATION, 277  
NUM\_GEV GVCPEXTENDEDSTATUSCODESELECTOR, 277  
NUM\_GEV GVSPEXTENDEDIDMODE, 278  
NUM\_GEV IEEE1588CLOCKACCURACY, 278  
NUM\_GEV IEEE1588MODE, 278  
NUM\_GEV IEEE1588STATUS, 279  
NUM\_GEV IP CONFIGURATIONSTATUS, 279  
NUM\_GEV PHYSICALLINKCONFIGURATION, 279  
NUM\_GEV SUPPORTEDOPTIONSELECTOR, 280  
NUM\_IMAGECOMPONENTSELECTOR, 281  
NUM\_IMAGECOMPRESSIONJPEGFORMATOPTION, 281  
NUM\_IMAGECOMPRESSIONMODE, 282  
NUM\_IMAGECOMPRESSIONRATEOPTION, 282  
NUM\_LINEFORMAT, 284  
NUM\_LINEINPUTFILTERSELECTOR, 285  
NUM\_LINEMODE, 285  
NUM\_LINESELECTOR, 285  
NUM\_LINESOURCE, 286  
NUM\_LOGICBLOCKLUTINPUTACTIVATION, 286  
NUM\_LOGICBLOCKLUTINPUTSELECTOR, 287  
NUM\_LOGICBLOCKLUTINPUTSOURCE, 287  
NUM\_LOGICBLOCKLUTSELECTOR, 288  
NUM\_LOGICBLOCKSELECTOR, 288  
NUM\_LUTSELECTOR, 288  
NUM\_PIXELCOLORFILTER, 289  
NUM\_PIXELFORMAT, 295  
NUM\_PIXELFORMATINFOSELECTOR, 301  
NUM\_PIXELSIZE, 302  
NUM\_REGIONDESTINATION, 303  
NUM\_REGIONMODE, 303  
NUM\_REGIONSELECTOR, 303  
NUM\_RGBTRANSFORMLIGHTSOURCE, 304  
NUM\_SCAN3DCOORDINATEREFERENCESELECTOR, 304  
NUM\_SCAN3DCOORDINATESELECTOR, 305  
NUM\_SCAN3DCOORDINATESYSTEM, 305  
NUM\_SCAN3DCOORDINATESYSTEMREFERENCE, 305  
NUM\_SCAN3DCOORDINATETRANSFORMSELECTOR, 306  
NUM\_SCAN3DDISTANCEUNIT, 306  
NUM\_SCAN3DOUTPUTMODE, 307

NUM\_SENSORDIGITIZATIONTAPS, 307  
 NUM\_SENSORSHUTTERMODE, 308  
 NUM\_SENSORTAPS, 308  
 NUM\_SEQUENCERCONFIGURATIONMODE,  
     308  
 NUM\_SEQUENCERCONFIGURATIONVALID, 309  
 NUM\_SEQUENCERMODE, 309  
 NUM\_SEQUENCERSETVALID, 309  
 NUM\_SEQUENCERTRIGGERACTIVATION, 310  
 NUM\_SEQUENCERTRIGGERSOURCE, 310  
 NUM\_SERIALPORTBAUDRATE, 310  
 NUM\_SERIALPORTPARITY, 311  
 NUM\_SERIALPORTSELECTOR, 311  
 NUM\_SERIALPORTSOURCE, 311  
 NUM\_SERIALPORTSTOPBITS, 312  
 NUM\_SOFTWARESIGNALSELECTOR, 312  
 NUM\_SOURCESELECTOR, 312  
 NUM\_STATISTICS\_CHANNELS, 313  
 NUM\_TESTPATTERN, 315  
 NUM\_TESTPATTERNGENERATORSELECTOR,  
     315  
 NUM\_TIMERSELECTOR, 317  
 NUM\_TIMERSTATUS, 317  
 NUM\_TIMERTRIGGERACTIVATION, 317  
 NUM\_TIMERTRIGGERSOURCE, 319  
 NUM\_TRANSFERCOMPONENTSELECTOR, 319  
 NUM\_TRANSFERCONTROLMODE, 320  
 NUM\_TRANSFEROPERATIONMODE, 320  
 NUM\_TRANSFERQUEUEMODE, 320  
 NUM\_TRANSFERSELECTOR, 321  
 NUM\_TRANSFERSTATUSSELECTOR, 321  
 NUM\_TRANSFERTRIGGERACTIVATION, 322  
 NUM\_TRANSFERTRIGGERMODE, 322  
 NUM\_TRANSFERTRIGGERSELECTOR, 322  
 NUM\_TRANSFERTRIGGERSOURCE, 323  
 NUM\_TRIGGERACTIVATION, 324  
 NUM\_TRIGGERMODE, 324  
 NUM\_TRIGGEROVERLAP, 324  
 NUM\_TRIGGERSELECTOR, 325  
 NUM\_TRIGGERSOURCE, 325  
 NUM\_USEROUTPUTSELECTOR, 326  
 NUM\_USERSETDEFAULT, 326  
 NUM\_USERSETSELECTOR, 326  
 NUM\_WHITECLIPSELECTOR, 327  
 NUMDEVICEACCESSSTATUS, 256  
 NUMDEVICECURRENTSPEED, 257  
 NUMDEVICEENDIANESSMECHANISM, 258  
 NUMDEVICETYPE, 263  
 NUMFILTERDRIVERSTATUS, 275  
 NUMGENICAMXMLLOCATION, 276  
 NUMGEVCCP, 276  
 NUMGUIMXMLLOCATION, 280  
 NUMINTERFACETYPE, 284  
 NUMPOESTATUS, 302  
 NUMSTREAMBUFFERCOUNTMODE, 314  
 NUMSTREAMBUFFERHANDLINGMODE, 314  
 NUMSTREAMTYPE, 315  
 NUMTLTYPE, 319  
 operator==, 328  
 PAYLOAD\_TYPE\_CHUNK\_DATA, 289  
 PAYLOAD\_TYPE\_CHUNK\_ONLY, 289  
 PAYLOAD\_TYPE\_CUSTOM\_ID, 289  
 PAYLOAD\_TYPE\_DEVICE\_SPECIFIC, 289  
 PAYLOAD\_TYPE\_EXTENDED\_CHUNK, 289  
 PAYLOAD\_TYPE\_FILE, 289  
 PAYLOAD\_TYPE\_H264, 289  
 PAYLOAD\_TYPE\_IMAGE, 289  
 PAYLOAD\_TYPE\_JPEG, 289  
 PAYLOAD\_TYPE\_JPEG2000, 289  
 PAYLOAD\_TYPE\_MULTI\_PART, 289  
 PAYLOAD\_TYPE\_RAW\_DATA, 289  
 PAYLOAD\_TYPE\_UNKNOWN, 289  
 PayloadTypeInfoIDs, 288  
 PGM, 282  
 PixelColorFilter\_BayerBG, 289  
 PixelColorFilter\_BayerGB, 289  
 PixelColorFilter\_BayerGR, 289  
 PixelColorFilter\_BayerRG, 289  
 PixelColorFilter\_None, 289  
 PixelColorFilterEnums, 289  
 PixelFormat\_B10, 292  
 PixelFormat\_B12, 292  
 PixelFormat\_B12\_Jpeg, 295  
 PixelFormat\_B16, 292  
 PixelFormat\_B8, 292  
 PixelFormat\_BayerBG10, 290  
 PixelFormat\_BayerBG10p, 290  
 PixelFormat\_BayerBG10Packed, 290  
 PixelFormat\_BayerBG12, 290  
 PixelFormat\_BayerBG12p, 290  
 PixelFormat\_BayerBG12Packed, 290  
 PixelFormat\_BayerBG16, 290  
 PixelFormat\_BayerBG8, 290  
 PixelFormat\_BayerGB10, 290  
 PixelFormat\_BayerGB10p, 290  
 PixelFormat\_BayerGB10Packed, 290  
 PixelFormat\_BayerGB12, 290  
 PixelFormat\_BayerGB12p, 290  
 PixelFormat\_BayerGB12Packed, 290  
 PixelFormat\_BayerGB16, 290  
 PixelFormat\_BayerGB8, 290  
 PixelFormat\_BayerGR10, 291  
 PixelFormat\_BayerGR10p, 290  
 PixelFormat\_BayerGR10Packed, 290  
 PixelFormat\_BayerGR12, 291  
 PixelFormat\_BayerGR12p, 290  
 PixelFormat\_BayerGR12Packed, 290  
 PixelFormat\_BayerGR16, 290  
 PixelFormat\_BayerGR8, 289  
 PixelFormat\_BayerRG10, 291  
 PixelFormat\_BayerRG10p, 290  
 PixelFormat\_BayerRG10Packed, 290  
 PixelFormat\_BayerRG12, 291  
 PixelFormat\_BayerRG12p, 290  
 PixelFormat\_BayerRG12Packed, 290  
 PixelFormat\_BayerRG16, 290

PixelFormat\_BayerRG8, 289  
PixelFormat\_BayerRGPolarized10p, 294  
PixelFormat\_BayerRGPolarized12p, 295  
PixelFormat\_BayerRGPolarized16, 295  
PixelFormat\_BayerRGPolarized8, 294  
PixelFormat\_BGR10, 291  
PixelFormat\_BGR10p, 291  
PixelFormat\_BGR12, 291  
PixelFormat\_BGR12p, 291  
PixelFormat\_BGR14, 291  
PixelFormat\_BGR16, 291  
PixelFormat\_BGR565p, 291  
PixelFormat\_BGR8, 290  
PixelFormat\_BGRa10, 291  
PixelFormat\_BGRa10p, 291  
PixelFormat\_BGRa12, 291  
PixelFormat\_BGRa12p, 291  
PixelFormat\_BGRa14, 291  
PixelFormat\_BGRa16, 291  
PixelFormat\_BGRa8, 290  
PixelFormat\_BiColorBGRG10, 293  
PixelFormat\_BiColorBGRG10p, 293  
PixelFormat\_BiColorBGRG12, 293  
PixelFormat\_BiColorBGRG12p, 293  
PixelFormat\_BiColorBGRG8, 292  
PixelFormat\_BiColorRGBG10, 293  
PixelFormat\_BiColorRGBG10p, 293  
PixelFormat\_BiColorRGBG12, 293  
PixelFormat\_BiColorRGBG12p, 293  
PixelFormat\_BiColorRGBG8, 293  
PixelFormat\_Confidence1, 292  
PixelFormat\_Confidence16, 292  
PixelFormat\_Confidence1p, 292  
PixelFormat\_Confidence32f, 292  
PixelFormat\_Confidence8, 292  
PixelFormat\_Coord3D\_A10p, 292  
PixelFormat\_Coord3D\_A12p, 292  
PixelFormat\_Coord3D\_A16, 292  
PixelFormat\_Coord3D\_A32f, 292  
PixelFormat\_Coord3D\_A8, 292  
PixelFormat\_Coord3D\_ABC10p, 292  
PixelFormat\_Coord3D\_ABC10p\_Planar, 292  
PixelFormat\_Coord3D\_ABC12p, 292  
PixelFormat\_Coord3D\_ABC12p\_Planar, 292  
PixelFormat\_Coord3D\_ABC16, 292  
PixelFormat\_Coord3D\_ABC16\_Planar, 292  
PixelFormat\_Coord3D\_ABC32f, 292  
PixelFormat\_Coord3D\_ABC32f\_Planar, 292  
PixelFormat\_Coord3D\_ABC8, 292  
PixelFormat\_Coord3D\_ABC8\_Planar, 292  
PixelFormat\_Coord3D\_AC10p, 292  
PixelFormat\_Coord3D\_AC10p\_Planar, 292  
PixelFormat\_Coord3D\_AC12p, 292  
PixelFormat\_Coord3D\_AC12p\_Planar, 292  
PixelFormat\_Coord3D\_AC16, 292  
PixelFormat\_Coord3D\_AC16\_Planar, 292  
PixelFormat\_Coord3D\_AC32f, 292  
PixelFormat\_Coord3D\_AC32f\_Planar, 292  
PixelFormat\_Coord3D\_AC8, 292  
PixelFormat\_Coord3D\_AC8\_Planar, 292  
PixelFormat\_Coord3D\_B10p, 292  
PixelFormat\_Coord3D\_B12p, 292  
PixelFormat\_Coord3D\_B16, 292  
PixelFormat\_Coord3D\_B32f, 292  
PixelFormat\_Coord3D\_B8, 292  
PixelFormat\_Coord3D\_C10p, 292  
PixelFormat\_Coord3D\_C12p, 292  
PixelFormat\_Coord3D\_C16, 292  
PixelFormat\_Coord3D\_C32f, 292  
PixelFormat\_Coord3D\_C8, 292  
PixelFormat\_G10, 291  
PixelFormat\_G12, 292  
PixelFormat\_G16, 292  
PixelFormat\_G8, 291  
PixelFormat\_GB12\_Jpeg, 295  
PixelFormat\_GR12\_Jpeg, 295  
PixelFormat\_JPEGColor8, 295  
PixelFormat\_JPEGMono8, 295  
PixelFormat\_LLCBayerRG8, 295  
PixelFormat\_LLCMono8, 295  
PixelFormat\_Mono10, 290  
PixelFormat\_Mono10p, 290  
PixelFormat\_Mono10Packed, 290  
PixelFormat\_Mono12, 290  
PixelFormat\_Mono12p, 290  
PixelFormat\_Mono12Packed, 290  
PixelFormat\_Mono14, 290  
PixelFormat\_Mono16, 289  
PixelFormat\_Mono16s, 290  
PixelFormat\_Mono1p, 290  
PixelFormat\_Mono2p, 290  
PixelFormat\_Mono32f, 290  
PixelFormat\_Mono4p, 290  
PixelFormat\_Mono8, 289  
PixelFormat\_Mono8s, 290  
PixelFormat\_Polarized10p, 294  
PixelFormat\_Polarized12p, 294  
PixelFormat\_Polarized16, 294  
PixelFormat\_Polarized8, 294  
PixelFormat\_R10, 291  
PixelFormat\_R12, 291  
PixelFormat\_R12\_Jpeg, 295  
PixelFormat\_R16, 291  
PixelFormat\_R8, 291  
PixelFormat\_Raw16, 295  
PixelFormat\_Raw8, 295  
PixelFormat\_RGB10, 291  
PixelFormat\_RGB10\_Planar, 291  
PixelFormat\_RGB10p, 291  
PixelFormat\_RGB10p32, 291  
PixelFormat\_RGB12, 291  
PixelFormat\_RGB12\_Planar, 291  
PixelFormat\_RGB12p, 291  
PixelFormat\_RGB14, 291  
PixelFormat\_RGB16, 291  
PixelFormat\_RGB16\_Planar, 291

- PixelFormat\_RGB16s, 291  
 PixelFormat\_RGB32f, 291  
 PixelFormat\_RGB565p, 291  
 PixelFormat\_RGB8, 291  
 PixelFormat\_RGB8\_Planar, 291  
 PixelFormat\_RGB8Packed, 289  
 PixelFormat\_RGBa10, 291  
 PixelFormat\_RGBa10p, 291  
 PixelFormat\_RGBa12, 291  
 PixelFormat\_RGBa12p, 291  
 PixelFormat\_RGBa14, 291  
 PixelFormat\_RGBa16, 291  
 PixelFormat\_RGBa32f, 291  
 PixelFormat\_RGBa8, 291  
 PixelFormat\_SCF1WBWG10, 293  
 PixelFormat\_SCF1WBWG10p, 293  
 PixelFormat\_SCF1WBWG12, 293  
 PixelFormat\_SCF1WBWG12p, 293  
 PixelFormat\_SCF1WBWG14, 293  
 PixelFormat\_SCF1WBWG16, 293  
 PixelFormat\_SCF1WBWG8, 293  
 PixelFormat\_SCF1WGWB10, 293  
 PixelFormat\_SCF1WGWB10p, 293  
 PixelFormat\_SCF1WGWB12, 293  
 PixelFormat\_SCF1WGWB12p, 293  
 PixelFormat\_SCF1WGWB14, 293  
 PixelFormat\_SCF1WGWB16, 293  
 PixelFormat\_SCF1WGWB8, 293  
 PixelFormat\_SCF1WGWR10, 293  
 PixelFormat\_SCF1WGWR10p, 293  
 PixelFormat\_SCF1WGWR12, 293  
 PixelFormat\_SCF1WGWR12p, 293  
 PixelFormat\_SCF1WGWR14, 293  
 PixelFormat\_SCF1WGWR16, 293  
 PixelFormat\_SCF1WGWR8, 293  
 PixelFormat\_SCF1WRWG10, 293  
 PixelFormat\_SCF1WRWG10p, 293  
 PixelFormat\_SCF1WRWG12, 293  
 PixelFormat\_SCF1WRWG12p, 293  
 PixelFormat\_SCF1WRWG14, 293  
 PixelFormat\_SCF1WRWG16, 293  
 PixelFormat\_SCF1WRWG8, 293  
 PixelFormat\_YCbCr10\_CbYCr, 293  
 PixelFormat\_YCbCr10p\_CbYCr, 293  
 PixelFormat\_YCbCr12\_CbYCr, 293  
 PixelFormat\_YCbCr12p\_CbYCr, 293  
 PixelFormat\_YCbCr411\_8, 290  
 PixelFormat\_YCbCr411\_8\_CbYYCrYY, 293  
 PixelFormat\_YCbCr422\_10, 293  
 PixelFormat\_YCbCr422\_10\_CbYCrY, 293  
 PixelFormat\_YCbCr422\_10p, 293  
 PixelFormat\_YCbCr422\_10p\_CbYCrY, 294  
 PixelFormat\_YCbCr422\_12, 294  
 PixelFormat\_YCbCr422\_12\_CbYCrY, 294  
 PixelFormat\_YCbCr422\_12p, 294  
 PixelFormat\_YCbCr422\_12p\_CbYCrY, 294  
 PixelFormat\_YCbCr422\_8, 290  
 PixelFormat\_YCbCr422\_8\_CbYCrY, 293  
 PixelFormat\_YCbCr601\_10\_CbYCr, 294  
 PixelFormat\_YCbCr601\_10p\_CbYCr, 294  
 PixelFormat\_YCbCr601\_12\_CbYCr, 294  
 PixelFormat\_YCbCr601\_12p\_CbYCr, 294  
 PixelFormat\_YCbCr601\_411\_8\_CbYYCrYY, 294  
 PixelFormat\_YCbCr601\_422\_10, 294  
 PixelFormat\_YCbCr601\_422\_10\_CbYCrY, 294  
 PixelFormat\_YCbCr601\_422\_10p, 294  
 PixelFormat\_YCbCr601\_422\_10p\_CbYCrY, 294  
 PixelFormat\_YCbCr601\_422\_12, 294  
 PixelFormat\_YCbCr601\_422\_12\_CbYCrY, 294  
 PixelFormat\_YCbCr601\_422\_12p, 294  
 PixelFormat\_YCbCr601\_422\_12p\_CbYCrY, 294  
 PixelFormat\_YCbCr601\_422\_8, 294  
 PixelFormat\_YCbCr601\_422\_8\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_8\_CbYCr, 294  
 PixelFormat\_YCbCr709\_10\_CbYCr, 294  
 PixelFormat\_YCbCr709\_10p\_CbYCr, 294  
 PixelFormat\_YCbCr709\_12\_CbYCr, 294  
 PixelFormat\_YCbCr709\_12p\_CbYCr, 294  
 PixelFormat\_YCbCr709\_411\_8\_CbYYCrYY, 294  
 PixelFormat\_YCbCr709\_422\_10, 294  
 PixelFormat\_YCbCr709\_422\_10\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_422\_10p, 294  
 PixelFormat\_YCbCr709\_422\_10p\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_422\_12, 294  
 PixelFormat\_YCbCr709\_422\_12\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_422\_12p, 294  
 PixelFormat\_YCbCr709\_422\_12p\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_422\_8, 294  
 PixelFormat\_YCbCr709\_422\_8\_CbYCrY, 294  
 PixelFormat\_YCbCr709\_8\_CbYCr, 294  
 PixelFormat\_YCbCr8, 290  
 PixelFormat\_YCbCr8\_CbYCr, 293  
 PixelFormat\_YUV411\_8\_UYYVYY, 294  
 PixelFormat\_YUV411Packed, 290  
 PixelFormat\_YUV422\_8, 294  
 PixelFormat\_YUV422\_8\_UYVY, 294  
 PixelFormat\_YUV422Packed, 290  
 PixelFormat\_YUV444Packed, 290  
 PixelFormat\_YUV8\_UYV, 294  
 PixelFormatEnums, 289  
 PixelFormatInfoSelector\_B10, 297  
 PixelFormatInfoSelector\_B12, 297  
 PixelFormatInfoSelector\_B16, 297  
 PixelFormatInfoSelector\_B8, 297  
 PixelFormatInfoSelector\_BayerBG10, 295  
 PixelFormatInfoSelector\_BayerBG10p, 295  
 PixelFormatInfoSelector\_BayerBG12, 295  
 PixelFormatInfoSelector\_BayerBG12p, 295  
 PixelFormatInfoSelector\_BayerBG16, 295  
 PixelFormatInfoSelector\_BayerBG8, 295  
 PixelFormatInfoSelector\_BayerGB10, 295  
 PixelFormatInfoSelector\_BayerGB10p, 295  
 PixelFormatInfoSelector\_BayerGB12, 296  
 PixelFormatInfoSelector\_BayerGB12p, 296  
 PixelFormatInfoSelector\_BayerGB16, 296  
 PixelFormatInfoSelector\_BayerGB8, 295

- PixelFormatInfoSelector\_BayerGR10, 296  
PixelFormatInfoSelector\_BayerGR10p, 296  
PixelFormatInfoSelector\_BayerGR12, 296  
PixelFormatInfoSelector\_BayerGR12p, 296  
PixelFormatInfoSelector\_BayerGR16, 296  
PixelFormatInfoSelector\_BayerGR8, 296  
PixelFormatInfoSelector\_BayerRG10, 296  
PixelFormatInfoSelector\_BayerRG10p, 296  
PixelFormatInfoSelector\_BayerRG12, 296  
PixelFormatInfoSelector\_BayerRG12p, 296  
PixelFormatInfoSelector\_BayerRG16, 296  
PixelFormatInfoSelector\_BayerRG8, 296  
PixelFormatInfoSelector\_BayerRGPolarized10p, 300  
PixelFormatInfoSelector\_BayerRGPolarized12p, 300  
PixelFormatInfoSelector\_BayerRGPolarized16, 300  
PixelFormatInfoSelector\_BayerRGPolarized8, 300  
PixelFormatInfoSelector\_BGR10, 297  
PixelFormatInfoSelector\_BGR10p, 297  
PixelFormatInfoSelector\_BGR12, 297  
PixelFormatInfoSelector\_BGR12p, 297  
PixelFormatInfoSelector\_BGR14, 297  
PixelFormatInfoSelector\_BGR16, 297  
PixelFormatInfoSelector\_BGR565p, 297  
PixelFormatInfoSelector\_BGR8, 297  
PixelFormatInfoSelector\_BGRa10, 296  
PixelFormatInfoSelector\_BGRa10p, 296  
PixelFormatInfoSelector\_BGRa12, 296  
PixelFormatInfoSelector\_BGRa12p, 296  
PixelFormatInfoSelector\_BGRa14, 296  
PixelFormatInfoSelector\_BGRa16, 296  
PixelFormatInfoSelector\_BGRa8, 296  
PixelFormatInfoSelector\_BiColorBGRG10, 298  
PixelFormatInfoSelector\_BiColorBGRG10p, 298  
PixelFormatInfoSelector\_BiColorBGRG12, 298  
PixelFormatInfoSelector\_BiColorBGRG12p, 298  
PixelFormatInfoSelector\_BiColorBGRG8, 298  
PixelFormatInfoSelector\_BiColorRGBG10, 298  
PixelFormatInfoSelector\_BiColorRGBG10p, 298  
PixelFormatInfoSelector\_BiColorRGBG12, 298  
PixelFormatInfoSelector\_BiColorRGBG12p, 298  
PixelFormatInfoSelector\_BiColorRGBG8, 298  
PixelFormatInfoSelector\_Confidence1, 298  
PixelFormatInfoSelector\_Confidence16, 298  
PixelFormatInfoSelector\_Confidence1p, 298  
PixelFormatInfoSelector\_Confidence32f, 298  
PixelFormatInfoSelector\_Confidence8, 298  
PixelFormatInfoSelector\_Coord3D\_A10p, 297  
PixelFormatInfoSelector\_Coord3D\_A12p, 297  
PixelFormatInfoSelector\_Coord3D\_A16, 297  
PixelFormatInfoSelector\_Coord3D\_A32f, 297  
PixelFormatInfoSelector\_Coord3D\_A8, 297  
PixelFormatInfoSelector\_Coord3D\_ABC10p, 297  
PixelFormatInfoSelector\_Coord3D\_ABC10p\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_ABC12p, 297  
PixelFormatInfoSelector\_Coord3D\_ABC16, 297  
PixelFormatInfoSelector\_Coord3D\_ABC16\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_ABC32f, 297  
PixelFormatInfoSelector\_Coord3D\_ABC32f\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_ABC8, 297  
PixelFormatInfoSelector\_Coord3D\_ABC8\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_AC10p, 297  
PixelFormatInfoSelector\_Coord3D\_AC10p\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_AC12p, 297  
PixelFormatInfoSelector\_Coord3D\_AC12p\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_AC16, 297  
PixelFormatInfoSelector\_Coord3D\_AC16\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_AC32f, 297  
PixelFormatInfoSelector\_Coord3D\_AC32f\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_AC8, 297  
PixelFormatInfoSelector\_Coord3D\_AC8\_Planar, 297  
PixelFormatInfoSelector\_Coord3D\_B10p, 297  
PixelFormatInfoSelector\_Coord3D\_B12p, 297  
PixelFormatInfoSelector\_Coord3D\_B16, 298  
PixelFormatInfoSelector\_Coord3D\_B32f, 298  
PixelFormatInfoSelector\_Coord3D\_B8, 297  
PixelFormatInfoSelector\_Coord3D\_C10p, 298  
PixelFormatInfoSelector\_Coord3D\_C12p, 298  
PixelFormatInfoSelector\_Coord3D\_C16, 298  
PixelFormatInfoSelector\_Coord3D\_C32f, 298  
PixelFormatInfoSelector\_Coord3D\_C8, 298  
PixelFormatInfoSelector\_G10, 297  
PixelFormatInfoSelector\_G12, 297  
PixelFormatInfoSelector\_G16, 297  
PixelFormatInfoSelector\_G8, 297  
PixelFormatInfoSelector\_JPEGColor8, 300  
PixelFormatInfoSelector\_JPEGMono8, 300  
PixelFormatInfoSelector\_LLCBayerRG8, 300  
PixelFormatInfoSelector\_LLCMono8, 300  
PixelFormatInfoSelector\_Mono10, 295  
PixelFormatInfoSelector\_Mono10p, 295  
PixelFormatInfoSelector\_Mono12, 295  
PixelFormatInfoSelector\_Mono12p, 295  
PixelFormatInfoSelector\_Mono14, 295  
PixelFormatInfoSelector\_Mono16, 295  
PixelFormatInfoSelector\_Mono16s, 295  
PixelFormatInfoSelector\_Mono1p, 295  
PixelFormatInfoSelector\_Mono2p, 295  
PixelFormatInfoSelector\_Mono32f, 295  
PixelFormatInfoSelector\_Mono4p, 295  
PixelFormatInfoSelector\_Mono8, 295  
PixelFormatInfoSelector\_Mono8s, 295  
PixelFormatInfoSelector\_Polarized10p, 300

- PixelFormatInfoSelector\_Polarized12p, 300  
 PixelFormatInfoSelector\_Polarized16, 300  
 PixelFormatInfoSelector\_Polarized8, 300  
 PixelFormatInfoSelector\_R10, 297  
 PixelFormatInfoSelector\_R12, 297  
 PixelFormatInfoSelector\_R16, 297  
 PixelFormatInfoSelector\_R8, 297  
 PixelFormatInfoSelector\_RGB10, 296  
 PixelFormatInfoSelector\_RGB10\_Planar, 296  
 PixelFormatInfoSelector\_RGB10p, 296  
 PixelFormatInfoSelector\_RGB10p32, 296  
 PixelFormatInfoSelector\_RGB12, 296  
 PixelFormatInfoSelector\_RGB12\_Planar, 296  
 PixelFormatInfoSelector\_RGB12p, 296  
 PixelFormatInfoSelector\_RGB14, 296  
 PixelFormatInfoSelector\_RGB16, 296  
 PixelFormatInfoSelector\_RGB16\_Planar, 296  
 PixelFormatInfoSelector\_RGB16s, 296  
 PixelFormatInfoSelector\_RGB32f, 296  
 PixelFormatInfoSelector\_RGB565p, 296  
 PixelFormatInfoSelector\_RGB8, 296  
 PixelFormatInfoSelector\_RGB8\_Planar, 296  
 PixelFormatInfoSelector\_RGba10, 296  
 PixelFormatInfoSelector\_RGba10p, 296  
 PixelFormatInfoSelector\_RGba12, 296  
 PixelFormatInfoSelector\_RGba12p, 296  
 PixelFormatInfoSelector\_RGba14, 296  
 PixelFormatInfoSelector\_RGba16, 296  
 PixelFormatInfoSelector\_RGba32f, 296  
 PixelFormatInfoSelector\_RGba8, 296  
 PixelFormatInfoSelector\_SCF1WBWG10, 298  
 PixelFormatInfoSelector\_SCF1WBWG10p, 298  
 PixelFormatInfoSelector\_SCF1WBWG12, 298  
 PixelFormatInfoSelector\_SCF1WBWG12p, 298  
 PixelFormatInfoSelector\_SCF1WBWG14, 298  
 PixelFormatInfoSelector\_SCF1WBWG16, 298  
 PixelFormatInfoSelector\_SCF1WBWG8, 298  
 PixelFormatInfoSelector\_SCF1WGWB10, 298  
 PixelFormatInfoSelector\_SCF1WGWB10p, 298  
 PixelFormatInfoSelector\_SCF1WGWB12, 298  
 PixelFormatInfoSelector\_SCF1WGWB12p, 298  
 PixelFormatInfoSelector\_SCF1WGWB14, 298  
 PixelFormatInfoSelector\_SCF1WGWB16, 298  
 PixelFormatInfoSelector\_SCF1WGWB8, 298  
 PixelFormatInfoSelector\_SCF1WGWR10, 299  
 PixelFormatInfoSelector\_SCF1WGWR10p, 299  
 PixelFormatInfoSelector\_SCF1WGWR12, 299  
 PixelFormatInfoSelector\_SCF1WGWR12p, 299  
 PixelFormatInfoSelector\_SCF1WGWR14, 299  
 PixelFormatInfoSelector\_SCF1WGWR16, 299  
 PixelFormatInfoSelector\_SCF1WGWR8, 298  
 PixelFormatInfoSelector\_SCF1WRWG10, 299  
 PixelFormatInfoSelector\_SCF1WRWG10p, 299  
 PixelFormatInfoSelector\_SCF1WRWG12, 299  
 PixelFormatInfoSelector\_SCF1WRWG12p, 299  
 PixelFormatInfoSelector\_SCF1WRWG14, 299  
 PixelFormatInfoSelector\_SCF1WRWG16, 299  
 PixelFormatInfoSelector\_SCF1WRWG8, 299  
 PixelFormatInfoSelector\_YCbCr10\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr10p\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr12\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr12p\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr411\_8, 299  
 PixelFormatInfoSelector\_YCbCr411\_8\_CbYYCrYY, 299  
 PixelFormatInfoSelector\_YCbCr422\_10, 299  
 PixelFormatInfoSelector\_YCbCr422\_10\_CbYCrY, 299  
 PixelFormatInfoSelector\_YCbCr422\_10p, 299  
 PixelFormatInfoSelector\_YCbCr422\_10p\_CbYCrY, 299  
 PixelFormatInfoSelector\_YCbCr422\_12, 299  
 PixelFormatInfoSelector\_YCbCr422\_12\_CbYCrY, 299  
 PixelFormatInfoSelector\_YCbCr422\_12p, 299  
 PixelFormatInfoSelector\_YCbCr422\_12p\_CbYCrY, 299  
 PixelFormatInfoSelector\_YCbCr422\_8, 299  
 PixelFormatInfoSelector\_YCbCr422\_8\_CbYCrY, 299  
 PixelFormatInfoSelector\_YCbCr601\_10\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr601\_10p\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr601\_12\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr601\_12p\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr601\_411\_8\_CbYYCrYY, 299  
 PixelFormatInfoSelector\_YCbCr601\_422\_10, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_10\_CbYCrY, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_10p, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_10p\_CbYCrY, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12\_CbYCrY, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12p, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_12p\_CbYCrY, 300  
 PixelFormatInfoSelector\_YCbCr601\_422\_8, 299  
 PixelFormatInfoSelector\_YCbCr601\_422\_8\_CbYCrY, 300  
 PixelFormatInfoSelector\_YCbCr601\_8\_CbYCr, 299  
 PixelFormatInfoSelector\_YCbCr709\_10\_CbYCr, 300  
 PixelFormatInfoSelector\_YCbCr709\_10p\_CbYCr, 300  
 PixelFormatInfoSelector\_YCbCr709\_12\_CbYCr, 300  
 PixelFormatInfoSelector\_YCbCr709\_12p\_CbYCr, 300  
 PixelFormatInfoSelector\_YCbCr709\_411\_8\_CbYYCrYY,

300  
PixelFormatInfoSelector\_YCbCr709\_422\_10, 300  
PixelFormatInfoSelector\_YCbCr709\_422\_10\_CbYCrY,  
300  
PixelFormatInfoSelector\_YCbCr709\_422\_10p, 300  
PixelFormatInfoSelector\_YCbCr709\_422\_10p\_CbYCrY,  
300  
PixelFormatInfoSelector\_YCbCr709\_422\_12, 300  
PixelFormatInfoSelector\_YCbCr709\_422\_12\_CbYCrY,  
300  
PixelFormatInfoSelector\_YCbCr709\_422\_12p, 300  
PixelFormatInfoSelector\_YCbCr709\_422\_12p\_CbYCrY,  
300  
PixelFormatInfoSelector\_YCbCr709\_422\_8, 300  
PixelFormatInfoSelector\_YCbCr709\_422\_8\_CbYCrY,  
300  
PixelFormatInfoSelector\_YCbCr709\_8\_CbYCr,  
300  
PixelFormatInfoSelector\_YCbCr8, 299  
PixelFormatInfoSelector\_YCbCr8\_CbYCr, 299  
PixelFormatInfoSelector\_YUV411\_8\_UYYVYY,  
300  
PixelFormatInfoSelector\_YUV422\_8, 300  
PixelFormatInfoSelector\_YUV422\_8\_UYVY, 300  
PixelFormatInfoSelector\_YUV8\_UYV, 300  
PixelFormatInfoSelectorEnums, 295  
PixelFormatIntType, 301  
PixelFormatNamespaceID, 301  
PixelSize\_Bpp1, 302  
PixelSize\_Bpp10, 302  
PixelSize\_Bpp12, 302  
PixelSize\_Bpp14, 302  
PixelSize\_Bpp16, 302  
PixelSize\_Bpp2, 302  
PixelSize\_Bpp20, 302  
PixelSize\_Bpp24, 302  
PixelSize\_Bpp30, 302  
PixelSize\_Bpp32, 302  
PixelSize\_Bpp36, 302  
PixelSize\_Bpp4, 302  
PixelSize\_Bpp48, 302  
PixelSize\_Bpp64, 302  
PixelSize\_Bpp8, 302  
PixelSize\_Bpp96, 302  
PixelSizeEnums, 302  
PNG, 283  
POEStatus\_NotSupported, 302  
POEStatus\_PowerOff, 302  
POEStatus\_PowerOn, 302  
POEStatusEnum, 302  
PPM, 282  
RAW, 283  
RED, 313  
RegionDestination\_Stream0, 303  
RegionDestination\_Stream1, 303  
RegionDestination\_Stream2, 303  
RegionDestinationEnums, 303  
RegionMode\_Off, 303  
RegionMode\_On, 303  
RegionModeEnums, 303  
RegionSelector\_All, 303  
RegionSelector\_Region0, 303  
RegionSelector\_Region1, 303  
RegionSelector\_Region2, 303  
RegionSelectorEnums, 303  
RgbTransformLightSource\_Cloudy6500K, 304  
RgbTransformLightSource\_CoolFluorescent4000K,  
304  
RgbTransformLightSource\_Custom, 304  
RgbTransformLightSource\_Daylight5000K, 304  
RgbTransformLightSource\_General, 304  
RgbTransformLightSource\_Shade8000K, 304  
RgbTransformLightSource\_Tungsten2800K, 304  
RgbTransformLightSource\_WarmFluorescent3000K,  
304  
RgbTransformLightSourceEnums, 303  
RIGOROUS, 246  
SATURATION, 313  
Scan3dCoordinateReferenceSelector\_RotationX,  
304  
Scan3dCoordinateReferenceSelector\_RotationY,  
304  
Scan3dCoordinateReferenceSelector\_RotationZ,  
304  
Scan3dCoordinateReferenceSelector\_TranslationX,  
304  
Scan3dCoordinateReferenceSelector\_TranslationY,  
304  
Scan3dCoordinateReferenceSelector\_TranslationZ,  
304  
Scan3dCoordinateReferenceSelectorEnums, 304  
Scan3dCoordinateSelector\_CoordinateA, 305  
Scan3dCoordinateSelector\_CoordinateB, 305  
Scan3dCoordinateSelector\_CoordinateC, 305  
Scan3dCoordinateSelectorEnums, 304  
Scan3dCoordinateSystem\_Cartesian, 305  
Scan3dCoordinateSystem\_Cylindrical, 305  
Scan3dCoordinateSystem\_Spherical, 305  
Scan3dCoordinateSystemEnums, 305  
Scan3dCoordinateSystemReference\_Anchor, 305  
Scan3dCoordinateSystemReference\_Transformed,  
305  
Scan3dCoordinateSystemReferenceEnums, 305  
Scan3dCoordinateTransformSelector\_RotationX,  
306  
Scan3dCoordinateTransformSelector\_RotationY,  
306  
Scan3dCoordinateTransformSelector\_RotationZ,  
306  
Scan3dCoordinateTransformSelector\_TranslationX,  
306  
Scan3dCoordinateTransformSelector\_TranslationY,  
306  
Scan3dCoordinateTransformSelector\_TranslationZ,  
306  
Scan3dCoordinateTransformSelectorEnums, 305

Scan3dDistanceUnit\_Inch, 306  
 Scan3dDistanceUnit\_Millimeter, 306  
 Scan3dDistanceUnitEnums, 306  
 Scan3dOutputMode\_CalibratedABC\_Grid, 306  
 Scan3dOutputMode\_CalibratedABC\_PointCloud,  
     306  
 Scan3dOutputMode\_CalibratedAC, 307  
 Scan3dOutputMode\_CalibratedAC\_Linescan, 307  
 Scan3dOutputMode\_CalibratedC, 307  
 Scan3dOutputMode\_CalibratedC\_Linescan, 307  
 Scan3dOutputMode\_DisparityC, 307  
 Scan3dOutputMode\_DisparityC\_Linescan, 307  
 Scan3dOutputMode\_RectifiedC, 307  
 Scan3dOutputMode\_RectifiedC\_Linescan, 307  
 Scan3dOutputMode\_UncalibratedC, 306  
 Scan3dOutputModeEnums, 306  
 SensorDigitizationTaps\_Eight, 307  
 SensorDigitizationTaps\_Four, 307  
 SensorDigitizationTaps\_One, 307  
 SensorDigitizationTaps\_Ten, 307  
 SensorDigitizationTaps\_Three, 307  
 SensorDigitizationTaps\_Two, 307  
 SensorDigitizationTapsEnums, 307  
 SensorShutterMode\_Global, 308  
 SensorShutterMode\_GlobalReset, 308  
 SensorShutterMode\_Rolling, 308  
 SensorShutterModeEnums, 307  
 SensorTaps\_Eight, 308  
 SensorTaps\_Four, 308  
 SensorTaps\_One, 308  
 SensorTaps\_Ten, 308  
 SensorTaps\_Three, 308  
 SensorTaps\_Two, 308  
 SensorTapsEnums, 308  
 SequencerConfigurationMode\_Off, 308  
 SequencerConfigurationMode\_On, 308  
 SequencerConfigurationModeEnums, 308  
 SequencerConfigurationValid\_No, 309  
 SequencerConfigurationValid\_Yes, 309  
 SequencerConfigurationValidEnums, 309  
 SequencerMode\_Off, 309  
 SequencerMode\_On, 309  
 SequencerModeEnums, 309  
 SequencerSetValid\_No, 309  
 SequencerSetValid\_Yes, 309  
 SequencerSetValidEnums, 309  
 SequencerTriggerActivation\_AnyEdge, 310  
 SequencerTriggerActivation\_FallingEdge, 310  
 SequencerTriggerActivation\_LevelHigh, 310  
 SequencerTriggerActivation\_LevelLow, 310  
 SequencerTriggerActivation\_RisingEdge, 310  
 SequencerTriggerActivationEnums, 309  
 SequencerTriggerSource\_FrameStart, 310  
 SequencerTriggerSource\_Off, 310  
 SequencerTriggerSourceEnums, 310  
 SerialPortBaudRate\_Baud115200, 310  
 SerialPortBaudRate\_Baud1200, 310  
 SerialPortBaudRate\_Baud14400, 310  
 SerialPortBaudRate\_Baud19200, 310  
 SerialPortBaudRate\_Baud230400, 310  
 SerialPortBaudRate\_Baud2400, 310  
 SerialPortBaudRate\_Baud300, 310  
 SerialPortBaudRate\_Baud38400, 310  
 SerialPortBaudRate\_Baud460800, 310  
 SerialPortBaudRate\_Baud4800, 310  
 SerialPortBaudRate\_Baud57600, 310  
 SerialPortBaudRate\_Baud600, 310  
 SerialPortBaudRate\_Baud921600, 310  
 SerialPortBaudRate\_Baud9600, 310  
 SerialPortBaudRateEnums, 310  
 SerialPortParity\_Even, 311  
 SerialPortParity\_Mark, 311  
 SerialPortParity\_None, 311  
 SerialPortParity\_Odd, 311  
 SerialPortParity\_Space, 311  
 SerialPortParityEnums, 311  
 SerialPortSelector\_SerialPort0, 311  
 SerialPortSelectorEnums, 311  
 SerialPortSource\_Line0, 311  
 SerialPortSource\_Line1, 311  
 SerialPortSource\_Line2, 311  
 SerialPortSource\_Line3, 311  
 SerialPortSource\_Off, 311  
 SerialPortSourceEnums, 311  
 SerialPortStopBits\_Bits1, 312  
 SerialPortStopBits\_Bits1AndAHalf, 312  
 SerialPortStopBits\_Bits2, 312  
 SerialPortStopBitsEnums, 312  
 SoftwareSignalSelector\_SoftwareSignal0, 312  
 SoftwareSignalSelector\_SoftwareSignal1, 312  
 SoftwareSignalSelector\_SoftwareSignal2, 312  
 SoftwareSignalSelectorEnums, 312  
 SourceSelector\_All, 312  
 SourceSelector\_Source0, 312  
 SourceSelector\_Source1, 312  
 SourceSelector\_Source2, 312  
 SourceSelectorEnums, 312  
 SPINNAKER\_ERR\_ABORT, 268  
 SPINNAKER\_ERR\_ACCESS\_DENIED, 268  
 SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL, 268  
 SPINNAKER\_ERR\_BUSY, 268  
 SPINNAKER\_ERR\_CUSTOM\_ID, 268  
 SPINNAKER\_ERR\_ERROR, 268  
 SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION,  
     268  
 SPINNAKER\_ERR\_IM\_CONVERT, 268  
 SPINNAKER\_ERR\_IM\_COPY, 268  
 SPINNAKER\_ERR\_IM\_DECOMPRESSION, 268  
 SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN, 268  
 SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE,  
     268  
 SPINNAKER\_ERR\_IM\_MALLOC, 268  
 SPINNAKER\_ERR\_IM\_MIN\_MAX, 268  
 SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED, 268  
 SPINNAKER\_ERR\_INVALID\_ADDRESS, 268  
 SPINNAKER\_ERR\_INVALID\_BUFFER, 268

SPINNAKER\_ERR\_INVALID\_HANDLE, 268  
SPINNAKER\_ERR\_INVALID\_ID, 268  
SPINNAKER\_ERR\_INVALID\_INDEX, 268  
SPINNAKER\_ERR\_INVALID\_PARAMETER, 268  
SPINNAKER\_ERR\_INVALID\_VALUE, 268  
SPINNAKER\_ERR\_IO, 268  
SPINNAKER\_ERR\_NO\_DATA, 268  
SPINNAKER\_ERR\_NOT\_AVAILABLE, 268  
SPINNAKER\_ERR\_NOT\_IMPLEMENTED, 268  
SPINNAKER\_ERR\_NOT\_INITIALIZED, 268  
SPINNAKER\_ERR\_OUT\_OF\_MEMORY, 268  
SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA,  
    268  
SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED,  
    268  
SPINNAKER\_ERR\_RESOURCE\_IN\_USE, 268  
SPINNAKER\_ERR\_SUCCESS, 267  
SPINNAKER\_ERR\_TIMEOUT, 268  
SPINNAKER\_EVENT\_ARRIVAL\_REMOVAL, 270  
SPINNAKER\_EVENT\_DEVICE, 270  
SPINNAKER\_EVENT\_DEVICE\_SPECIFIC, 270  
SPINNAKER\_EVENT\_INTERFACE\_ARRIVAL\_REMOVAL,  
    270  
SPINNAKER\_EVENT\_LOGGING\_EVENT, 270  
SPINNAKER\_EVENT\_NEW\_BUFFER, 270  
SPINNAKER\_EVENT\_UNKNOWN, 270  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID,  
    302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV,  
    301  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC,  
    302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_16BIT  
    302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_32BIT  
    302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN,  
    301  
SpinnakerLogLevel, 312  
StatisticsChannel, 313  
StreamBufferCountMode\_Auto, 314  
StreamBufferCountMode\_Manual, 313  
StreamBufferCountModeEnum, 313  
StreamBufferHandlingMode\_NewestFirst, 314  
StreamBufferHandlingMode\_NewestOnly, 314  
StreamBufferHandlingMode\_OldestFirst, 314  
StreamBufferHandlingMode\_OldestFirstOverwrite,  
    314  
StreamBufferHandlingModeEnum, 314  
StreamType\_CameraLink, 315  
StreamType\_CameraLinkHS, 315  
StreamType\_CoaXPress, 315  
StreamType\_Custom, 315  
StreamType\_GigEVision, 315  
StreamType\_USB3Vision, 315  
StreamTypeEnum, 314  
TestPattern\_Increment, 315  
TestPattern\_Off, 315  
TestPattern\_SensorTestPattern, 315  
TestPatternEnums, 315  
TestPatternGeneratorSelector\_PipelineStart, 315  
TestPatternGeneratorSelector\_Sensor, 315  
TestPatternGeneratorSelectorEnums, 315  
TIFF, 283  
TimerSelector\_Timer0, 317  
TimerSelector\_Timer1, 317  
TimerSelector\_Timer2, 317  
TimerSelectorEnums, 315  
TimerStatus\_TimerActive, 317  
TimerStatus\_TimerCompleted, 317  
TimerStatus\_TimerIdle, 317  
TimerStatus\_TimerTriggerWait, 317  
TimerStatusEnums, 317  
TimerTriggerActivation\_AnyEdge, 317  
TimerTriggerActivation\_FallingEdge, 317  
TimerTriggerActivation\_LevelHigh, 317  
TimerTriggerActivation\_LevelLow, 317  
TimerTriggerActivation\_RisingEdge, 317  
TimerTriggerActivationEnums, 317  
TimerTriggerSource\_AcquisitionEnd, 318  
TimerTriggerSource\_AcquisitionStart, 318  
TimerTriggerSource\_AcquisitionTrigger, 318  
TimerTriggerSource\_Action0, 318  
TimerTriggerSource\_Action1, 319  
TimerTriggerSource\_Action2, 319  
TimerTriggerSource\_Counter0End, 318  
TimerTriggerSource\_Counter0Start, 318  
TimerTriggerSource\_Counter1End, 318  
TimerTriggerSource\_Counter1Start, 318  
TimerTriggerSource\_Counter2End, 318  
TimerTriggerSource\_Counter2Start, 318  
TimerTriggerSource\_Encoder0, 318  
TimerTriggerSource\_Encoder1, 318  
TimerTriggerSource\_Encoder2, 318  
TimerTriggerSource\_ExposureEnd, 318  
TimerTriggerSource\_ExposureStart, 318  
TimerTriggerSource\_FrameBurstEnd, 318  
TimerTriggerSource\_FrameBurstStart, 318  
TimerTriggerSource\_FrameEnd, 318  
TimerTriggerSource\_FrameStart, 318  
TimerTriggerSource\_FrameTrigger, 318  
TimerTriggerSource\_Line0, 318  
TimerTriggerSource\_Line1, 318  
TimerTriggerSource\_Line2, 318  
TimerTriggerSource\_LineEnd, 318  
TimerTriggerSource\_LineStart, 318  
TimerTriggerSource\_LineTrigger, 318  
TimerTriggerSource\_LinkTrigger0, 319  
TimerTriggerSource\_LinkTrigger1, 319  
TimerTriggerSource\_LinkTrigger2, 319  
TimerTriggerSource\_Off, 318  
TimerTriggerSource\_SoftwareSignal0, 318  
TimerTriggerSource\_SoftwareSignal1, 318  
TimerTriggerSource\_SoftwareSignal2, 318  
TimerTriggerSource\_Timer0End, 318  
TimerTriggerSource\_Timer0Start, 318

TimerTriggerSource\_Timer1End, 318  
TimerTriggerSource\_Timer1Start, 318  
TimerTriggerSource\_Timer2End, 318  
TimerTriggerSource\_Timer2Start, 318  
TimerTriggerSource\_UserOutput0, 318  
TimerTriggerSource\_UserOutput1, 318  
TimerTriggerSource\_UserOutput2, 318  
TimerTriggerSourceEnums, 317  
TLType\_CameraLink, 319  
TLType\_CameraLinkHS, 319  
TLType\_CoaxPress, 319  
TLType\_Custom, 319  
TLType\_GigEVision, 319  
TLType\_Mixed, 319  
TLType\_USB3Vision, 319  
TLTypeEnum, 319  
TransferComponentSelector\_All, 319  
TransferComponentSelector\_Blue, 319  
TransferComponentSelector\_Green, 319  
TransferComponentSelector\_Red, 319  
TransferComponentSelectorEnums, 319  
TransferControlMode\_Automatic, 320  
TransferControlMode\_Basic, 320  
TransferControlMode\_UserControlled, 320  
TransferControlModeEnums, 320  
TransferOperationMode\_Continuous, 320  
TransferOperationMode\_MultiBlock, 320  
TransferOperationModeEnums, 320  
TransferQueueMode\_FirstInFirstOut, 320  
TransferQueueModeEnums, 320  
TransferSelector\_All, 321  
TransferSelector\_Stream0, 321  
TransferSelector\_Stream1, 321  
TransferSelector\_Stream2, 321  
TransferSelectorEnums, 320  
TransferStatusSelector\_Paused, 321  
TransferStatusSelector\_QueueOverflow, 321  
TransferStatusSelector\_Stopped, 321  
TransferStatusSelector\_Stopping, 321  
TransferStatusSelector\_Streaming, 321  
TransferStatusSelectorEnums, 321  
TransferTriggerActivation\_AnyEdge, 321  
TransferTriggerActivation\_FallingEdge, 321  
TransferTriggerActivation\_LevelHigh, 322  
TransferTriggerActivation\_LevelLow, 322  
TransferTriggerActivation\_RisingEdge, 321  
TransferTriggerActivationEnums, 321  
TransferTriggerMode\_Off, 322  
TransferTriggerMode\_On, 322  
TransferTriggerModeEnums, 322  
TransferTriggerSelector\_TransferAbort, 322  
TransferTriggerSelector\_TransferActive, 322  
TransferTriggerSelector\_TransferBurstStart, 322  
TransferTriggerSelector\_TransferBurstStop, 322  
TransferTriggerSelector\_TransferPause, 322  
TransferTriggerSelector\_TransferResume, 322  
TransferTriggerSelector\_TransferStart, 322  
TransferTriggerSelector\_TransferStop, 322  
TransferTriggerSourceEnums, 322  
TransferTriggerSource\_Action0, 323  
TransferTriggerSource\_Action1, 323  
TransferTriggerSource\_Action2, 323  
TransferTriggerSource\_Counter0End, 323  
TransferTriggerSource\_Counter0Start, 323  
TransferTriggerSource\_Counter1End, 323  
TransferTriggerSource\_Counter1Start, 323  
TransferTriggerSource\_Counter2End, 323  
TransferTriggerSource\_Counter2Start, 323  
TransferTriggerSource\_Line0, 323  
TransferTriggerSource\_Line1, 323  
TransferTriggerSource\_Line2, 323  
TransferTriggerSource\_SoftwareSignal0, 323  
TransferTriggerSource\_SoftwareSignal1, 323  
TransferTriggerSource\_SoftwareSignal2, 323  
TransferTriggerSource\_Timer0End, 323  
TransferTriggerSource\_Timer0Start, 323  
TransferTriggerSource\_Timer1End, 323  
TransferTriggerSource\_Timer1Start, 323  
TransferTriggerSource\_Timer2End, 323  
TransferTriggerSource\_Timer2Start, 323  
TransferTriggerSourceEnums, 322  
TriggerActivation\_AnyEdge, 324  
TriggerActivation\_FallingEdge, 324  
TriggerActivation\_LevelHigh, 324  
TriggerActivation\_LevelLow, 324  
TriggerActivation\_RisingEdge, 324  
TriggerActivationEnums, 324  
TriggerMode\_Off, 324  
TriggerMode\_On, 324  
TriggerModeEnums, 324  
TriggerOverlap\_Off, 324  
TriggerOverlap\_PreviousFrame, 324  
TriggerOverlap\_ReadOut, 324  
TriggerOverlapEnums, 324  
TriggerSelector\_AcquisitionStart, 325  
TriggerSelector\_FrameBurstStart, 325  
TriggerSelector\_FrameStart, 325  
TriggerSelectorEnums, 324  
TriggerSource\_Action0, 325  
TriggerSource\_Counter0End, 325  
TriggerSource\_Counter0Start, 325  
TriggerSource\_Counter1End, 325  
TriggerSource\_Counter1Start, 325  
TriggerSource\_Line0, 325  
TriggerSource\_Line1, 325  
TriggerSource\_Line2, 325  
TriggerSource\_Line3, 325  
TriggerSource\_LogicBlock0, 325  
TriggerSource\_LogicBlock1, 325  
TriggerSource\_Software, 325  
TriggerSource\_UserOutput0, 325  
TriggerSource\_UserOutput1, 325  
TriggerSource\_UserOutput2, 325  
TriggerSource\_UserOutput3, 325  
TriggerSourceEnums, 325  
UNKNOWN\_PIXELFORMAT, 295

UserOutputSelector\_UserOutput0, 326  
UserOutputSelector\_UserOutput1, 326  
UserOutputSelector\_UserOutput2, 326  
UserOutputSelector\_UserOutput3, 326  
UserOutputSelectorEnums, 325  
UserSetDefault\_Default, 326  
UserSetDefault\_UserSet0, 326  
UserSetDefault\_UserSet1, 326  
UserSetDefaultEnums, 326  
UserSetSelector\_Default, 326  
UserSetSelector\_UserSet0, 326  
UserSetSelector\_UserSet1, 326  
UserSetSelectorEnums, 326  
WEIGHTED\_DIRECTIONAL\_FILTER, 246  
WhiteClipSelector\_All, 327  
WhiteClipSelector\_Blue, 327  
WhiteClipSelector\_Green, 327  
WhiteClipSelector\_Red, 327  
WhiteClipSelector\_Tap1, 327  
WhiteClipSelector\_Tap2, 327  
WhiteClipSelector\_U, 327  
WhiteClipSelector\_V, 327  
WhiteClipSelector\_Y, 327  
WhiteClipSelectorEnums, 326  
Spinnaker Classes, 39  
Spinnaker Definitions, 73  
Spinnaker EventHandler Classes, 49  
Spinnaker GenApi Classes, 93  
Spinnaker GenApi Enums, 178  
Spinnaker GenApi Interfaces, 95  
Spinnaker GenApi Utilities, 123  
Spinnaker Headers, 71  
Spinnaker Platform, 74  
Spinnaker QuickSpin Classes, 80  
Spinnaker Video Class, 75  
Spinnaker Video Definitions, 76  
Spinnaker.h, 72  
Spinnaker::GenApi, 329  
  \_ClearXMLCache, 354  
  \_Connect, 354  
  \_CycleDetectAccesMode, 347  
  \_EAccessMode, 347  
  \_ECachingMode, 347  
  \_ECallbackType, 348  
  \_EDisplayNotation, 348  
  \_EEndianess, 348  
  \_EGenApiSchemaVersion, 349  
  \_EIncMode, 349  
  \_EInputDirection, 349  
  \_EInterfaceType, 350  
  \_ELinkType, 350  
  \_ENameSpace, 350  
  \_ERepresentation, 351  
  \_ESign, 351  
  \_ESlope, 351  
  \_EStandardNameSpace, 352  
  \_EVisibility, 352  
  \_EXMLValidation, 352  
  \_EYesNo, 353  
  \_GetDeviceName, 354  
  \_GetNode, 355  
  \_GetNodes, 355  
  \_GetSupportedSchemaVersions, 355  
  \_InvalidateNodes, 355  
  \_LoadXMLFromFile, 355  
  \_LoadXMLFromFileInject, 355  
  \_LoadXMLFromString, 355  
  \_LoadXMLFromStringInject, 356  
  \_LoadXMLFromZIPData, 356  
  \_LoadXMLFromZIPFile, 356  
  \_Poll, 356  
  \_Undefined, 349  
  \_UndefinedAccesMode, 347  
  \_UndefinedCachingMode, 348  
  \_UndefinedEDisplayNotation, 348  
  \_UndefinedESlope, 352  
  \_UndefinedEXMLValidation, 353  
  \_UndefinedEndian, 349  
  \_UndefinedNameSpace, 351  
  \_UndefinedRepresentation, 351  
  \_UndefinedSign, 351  
  \_UndefinedStandardNameSpace, 352  
  \_UndefinedVisibility, 352  
  \_UndefinedYesNo, 353  
Address, 379  
Automatic, 352  
Beginner, 352  
BigEndian, 349  
Boolean, 351  
CacheChunkData, 356  
CacheUsage\_Automatic, 353  
CacheUsage\_ForceRead, 353  
CacheUsage\_ForceWrite, 353  
CacheUsage\_Ignore, 353  
CallbackHandleType, 344  
CastToIDestroy, 356  
CBooleanRef, 344  
cbPostInsideLock, 348  
cbPostOutsideLock, 348  
CCategoryRef, 344  
CCommandRef, 345  
CEnumEntryRef, 345  
CEnumerationRef, 345  
CFloatRef, 345  
CIntegerRef, 345  
CL, 352  
CNodeMapRef, 345  
CNodeRef, 345  
COMMAND\_MAGIC, 379  
Connect, 357  
ContentType\_Xml, 354  
ContentType\_ZippedXml, 354  
CPortRecorderRef, 345  
CPortRef, 346  
CRegisterRef, 346  
CSelectorRef, 346

CStringRef, 346  
ctDependingNodes, 350  
ctInValidatingChildren, 350  
ctParentNodes, 350  
ctReadingChildren, 350  
ctTerminalNodes, 350  
ctWritingChildren, 350  
Custom, 351  
CValueRef, 346  
Decreasing, 352  
Deregister, 357  
DeregisterCallback, 357  
EatComments, 357  
ECacheUsage\_t, 353  
EContentType\_t, 354  
Expert, 352  
ExtractIndependentSubtree, 358  
fixedIncrement, 349  
fnAutomatic, 348  
fnFixed, 348  
fnScientific, 348  
FromString, 358  
GENCP\_COMMAND\_HEADER\_SIZE, 379  
GENCP\_EVENT\_BASIC\_SIZE, 379  
GENCP\_EVENT\_CMD\_ID, 379  
Get, 358  
GetAddress, 359  
GetAlias, 359  
GetCachingMode, 359  
GetCastAlias, 359  
GetChildren, 359  
GetCookie, 360  
GetCurrentEntry, 360  
GetDescription, 360  
GetDeviceName, 360  
GetDeviceVersion, 360  
GetDisplayName, 360  
GetDisplayNotation, 361  
GetDisplayPrecision, 361  
GetDocuURL, 361  
GetEntries, 361  
GetEntry, 361  
GetEntryByName, 362  
GetEventID, 362  
GetGenApiVersion, 362  
GetInc, 362  
GetIncMode, 362  
GetIntValue, 362  
GetLength, 363  
GetListOfValidValues, 363  
GetLock, 363  
GetMax, 363  
GetMaxLength, 363  
GetMin, 364  
GetNameSpace, 364  
GetNode, 364  
GetNodeMap, 364  
GetNumericValue, 364  
GetNumNodes, 364  
GetParents, 364  
GetPollingTime, 365  
GetPrincipalInterfaceType, 365  
GetProductGuid, 365  
GetProperty, 365  
GetPropertyNames, 365  
GetRepresentation, 365  
GetSchemaVersion, 366  
GetSelectedFeatures, 366  
GetSelectingFeatures, 366  
GetSelectorList, 366  
GetStandardNameSpace, 366  
GetSupportedSchemaVersions, 367  
GetSwapEndianess, 367  
GetSymbolic, 367  
GetToolTip, 367  
GetUnit, 367  
GetValue, 367  
GetVendorName, 368  
GetVersionGuid, 368  
GetVisibility, 368  
GEV, 352  
Guru, 352  
GVCP\_MESSAGE\_TAGS, 354  
HasInc, 368  
HexNumber, 351  
IBase, 380  
IBoolean, 380  
ICategory, 380  
IChunkPort, 380  
 ICommand, 380  
IDestroy, 380  
IDevFileStream, 346  
IDeviceInfo, 381  
idFrom, 349  
idNone, 349  
idTo, 349  
IEnumEntry, 381  
IEnumeration, 381  
IEnumerationT, 381  
IEnumReference, 381  
IFloat, 382  
IIDC, 352  
IInteger, 382  
ImposeAccessMode, 368  
ImposeMax, 368, 369  
ImposeMin, 369  
ImposeVisibility, 369  
Increasing, 352  
INode, 382  
INodeMap, 382  
INodeMapDyn, 382  
intfIBase, 350  
intfIBoolean, 350  
intfICategory, 350  
intfICommand, 350  
intfIEnumEntry, 350

intfIEnumeration, 350  
intfIFloat, 350  
intfIInteger, 350  
intfIPort, 350  
intfIRegister, 350  
intfIString, 350  
intfIValue, 350  
Invalidate, 383  
InvalidateNode, 369  
InvalidateNodes, 369  
Invisible, 352  
IPersistScript, 383  
IPort, 383  
IPortConstruct, 383  
IPortRecorder, 383  
IPortReplay, 383  
IPortWriteList, 384  
IPv4Address, 351  
IReference, 384  
IRegister, 384  
IsAccessModeCacheable, 370  
IsCachable, 370  
IsDeprecated, 370  
IsDone, 370  
ISelector, 384  
ISelectorDigit, 384  
IsFeature, 370  
IsSelfClearing, 371  
IsStreamable, 371  
IString, 385  
IsValueCacheValid, 371  
IValue, 385  
Length, 385  
Linear, 351  
listIncrement, 349  
LittleEndian, 349  
LoadXMLFromFile, 371  
LoadXMLFromFileInject, 371  
LoadXMLFromString, 371  
LoadXMLFromStringInject, 372  
LoadXMLFromZIPData, 372  
LoadXMLFromZIPFile, 372  
Logarithmic, 351  
MACAddress, 351  
make\_NodeCallback, 372  
MergeXMLFiles, 373  
NA, 347  
NI, 347  
No, 353  
NoCache, 348  
 NodeList\_t, 346  
noIncrement, 349  
None, 352  
ODevFileStream, 347  
operator!=, 373  
operator<<, 373  
operator>>, 375  
operator\*, 373  
operator(), 373  
operator=, 374  
operator==, 375  
PersistFeature, 375  
Poll, 375  
PreprocessXMLFromFile, 375  
PreprocessXMLFromZIPFile, 376  
PureNumber, 351  
Register, 376  
RegisterCallback, 376  
Replay, 377  
Restore, 377  
RO, 347  
RW, 347  
SET\_GUID, 377  
SetCookie, 377  
SetIntValue, 377  
SetNext, 378  
SetNumEnums, 378  
Signed, 351  
Standard, 351  
StopRecording, 378  
StringList\_t, 347  
TAG\_EVENT\_CMD, 354  
TAG\_EVENTDATA\_CMD, 354  
ToString, 378  
U3V\_EVENT\_PREFIX, 385  
Unsigned, 351  
USB, 352  
v1\_0, 349  
v1\_1, 349  
Varying, 352  
Verify, 385  
WO, 347  
Write, 379  
WriteAround, 348  
WriteThrough, 348  
xvAll, 353  
xvCycles, 353  
xvDefault, 353  
xvLoad, 353  
xvSFNC, 353  
Yes, 353  
Spinnaker::GenICam, 386  
DoesEnvironmentVariableExist, 387  
GetFiles, 387  
GetGenICamCacheFolder, 387  
GetGenICamCLProtocolFolder, 388  
GetGenICamLogConfig, 388  
getline, 388  
GetModulePathFromFunction, 388  
GetValueOfEnvironmentVariable, 389  
INTEGRAL\_CAST, 389  
INTEGRAL\_CAST2, 389  
ReplaceEnvironmentVariables, 390  
SetGenICamCacheFolder, 390  
SetGenICamCLProtocolFolder, 390  
SetGenICamLogConfig, 390

ThrowBadAlloc, 390  
Tokenize, 390  
UrlDecode, 391  
UrlEncode, 391  
Spinnaker::Video, 391  
SPINNAKER\_API  
    SpinnakerPlatform.h, 1269  
SPINNAKER\_API\_ABSTRACT  
    SpinnakerPlatform.h, 1270  
SPINNAKER\_ERR\_ABORT  
    Spinnaker, 268  
SPINNAKER\_ERR\_ACCESS\_DENIED  
    Spinnaker, 268  
SPINNAKER\_ERR\_BUFFER\_TOO\_SMALL  
    Spinnaker, 268  
SPINNAKER\_ERR\_BUSY  
    Spinnaker, 268  
SPINNAKER\_ERR\_CUSTOM\_ID  
    Spinnaker, 268  
SPINNAKER\_ERR\_ERROR  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_COLOR\_CONVERSION  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_CONVERT  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_COPY  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_DECOMPRESSION  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_HISTOGRAM\_MEAN  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_HISTOGRAM\_RANGE  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_MALLOC  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_MIN\_MAX  
    Spinnaker, 268  
SPINNAKER\_ERR\_IM\_NOT\_SUPPORTED  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_ADDRESS  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_BUFFER  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_HANDLE  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_ID  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_INDEX  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_PARAMETER  
    Spinnaker, 268  
SPINNAKER\_ERR\_INVALID\_VALUE  
    Spinnaker, 268  
SPINNAKER\_ERR\_IO  
    Spinnaker, 268  
SPINNAKER\_ERR\_NO\_DATA  
    Spinnaker, 268  
SPINNAKER\_ERR\_NOT\_AVAILABLE  
    Spinnaker, 268  
SPINNAKER\_ERR\_NOT\_IMPLEMENTED  
    Spinnaker, 268  
SPINNAKER\_ERR\_NOT\_INITIALIZED  
    Spinnaker, 268  
SPINNAKER\_ERR\_OUT\_OF\_MEMORY  
    Spinnaker, 268  
SPINNAKER\_ERR\_PARSING\_CHUNK\_DATA  
    Spinnaker, 268  
SPINNAKER\_ERR\_RESOURCE\_EXHAUSTED  
    Spinnaker, 268  
SPINNAKER\_ERR\_RESOURCE\_IN\_USE  
    Spinnaker, 268  
SPINNAKER\_ERR\_SUCCESS  
    Spinnaker, 267  
SPINNAKER\_ERR\_TIMEOUT  
    Spinnaker, 268  
SPINNAKER\_EVENT\_ARRIVAL\_REMOVAL  
    Spinnaker, 270  
SPINNAKER\_EVENT\_DEVICE  
    Spinnaker, 270  
SPINNAKER\_EVENT\_DEVICE\_SPECIFIC  
    Spinnaker, 270  
SPINNAKER\_EVENT\_INTERFACE\_ARRIVAL\_REMOVAL  
    Spinnaker, 270  
SPINNAKER\_EVENT\_LOGGING\_EVENT  
    Spinnaker, 270  
SPINNAKER\_EVENT\_NEW\_BUFFER  
    Spinnaker, 270  
SPINNAKER\_EVENT\_UNKNOWN  
    Spinnaker, 270  
SPINNAKER\_EXCEPTION  
    ExceptionHandling.cpp, 1341  
SPINNAKER\_LOCAL  
    SpinnakerPlatform.h, 1270  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_CUSTOM\_ID  
    Spinnaker, 302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_GEV  
    Spinnaker, 301  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_IIDC  
    Spinnaker, 302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_16BIT  
    Spinnaker, 302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_PFNC\_32BIT  
    Spinnaker, 302  
SPINNAKER\_PIXELFORMAT\_NAMESPACE\_UNKNOWN  
    Spinnaker, 301  
SpinnakerLogLevel  
    Spinnaker, 312  
SpinnakerPlatform.h  
    SPINNAKER\_API, 1269  
    SPINNAKER\_API\_ABSTRACT, 1270  
    SPINNAKER\_LOCAL, 1270  
SpinTestCamera, 1031  
SpinTestCamera Class, 173  
SpinUpdate.h  
    GetErrorMessage, 1271  
    SetMessageCallback, 1271

SetProgressCallback, 1271  
SPINUPDATE\_API, 1271  
UpdateFirmware, 1271  
UpdateFirmwareConsole, 1271  
UpdateFirmwareGUI, 1272  
UpdatorMessageCallback, 1272  
UpdatorProgressCallback, 1272  
SPINUPDATE\_API  
    SpinUpdate.h, 1271  
SpinVideo, 1031  
    ~SpinVideo, 1032  
    Append, 1032  
    Close, 1033  
    Open, 1033, 1035  
    SetMaximumFileSize, 1035  
    SpinVideo, 1032  
src/Acquisition/Acquisition.cpp, 1281  
src/Acquisition/resource.h, 1284  
src/Acquisition/stdafx.cpp, 1284  
src/Acquisition/stdafx.h, 1293  
src/Acquisition/targetver.h, 1309  
src/AcquisitionMultipleCameraRecovery/AcquisitionMultipleCameraRecovery/resource.h, 1324  
src/AcquisitionMultipleCameraRecovery/resource.h, 1284  
src/AcquisitionMultipleThread/AcquisitionMultipleThread.cpp, 1326  
src/AcquisitionMultipleThread/resource.h, 1284  
src/ActionCommand/ActionCommand.cpp, 1327  
src/ActionCommand/resource.h, 1284  
src/ActionCommand/stdafx.cpp, 1285  
src/ActionCommand/stdafx.h, 1294  
src/ActionCommand/targetver.h, 1310  
src/BufferHandling/BufferHandling.cpp, 1329  
src/BufferHandling/resource.h, 1284  
src/BufferHandling/stdafx.cpp, 1285  
src/BufferHandling/stdafx.h, 1295  
src/BufferHandling/targetver.h, 1311  
src/ChunkData/ChunkData.cpp, 1332  
src/ChunkData/resource.h, 1284  
src/CounterAndTimer/CounterAndTimer.cpp, 1334  
src/CounterAndTimer/resource.h, 1284  
src/CounterAndTimer/stdafx.cpp, 1286  
src/CounterAndTimer/stdafx.h, 1296  
src/CounterAndTimer/targetver.h, 1312  
src/DeviceEvents/DeviceEvents.cpp, 1336  
src/DeviceEvents/resource.h, 1284  
src/DeviceEvents/stdafx.cpp, 1286  
src/DeviceEvents/stdafx.h, 1297  
src/DeviceEvents/targetver.h, 1313  
src/Enumeration/Enumeration.cpp, 1338  
src/Enumeration/resource.h, 1284  
src/Enumeration/stdafx.cpp, 1287  
src/Enumeration/stdafx.h, 1298  
src/Enumeration/targetver.h, 1314  
src/Enumeration\_QuickSpin/Enumeration\_QuickSpin.cpp, 1339  
src/Enumeration\_QuickSpin/resource.h, 1284  
src/Enumeration\_QuickSpin/stdafx.cpp, 1287  
src/Enumeration\_QuickSpin/stdafx.h, 1299  
src/Enumeration\_QuickSpin/targetver.h, 1315  
src/EnumerationEvents/EnumerationEvents.cpp, 1339  
src/EnumerationEvents/resource.h, 1284  
src/ExceptionHandling/ExceptionHandling.cpp, 1340  
src/ExceptionHandling/resource.h, 1284  
src/ExceptionHandling/stdafx.cpp, 1288  
src/ExceptionHandling/stdafx.h, 1300  
src/ExceptionHandling/targetver.h, 1316  
src/Exposure/Exposure.cpp, 1342  
src/Exposure/resource.h, 1284  
src/Exposure/stdafx.cpp, 1288  
src/Exposure/stdafx.h, 1301  
src/Exposure/targetver.h, 1317  
src/Exposure\_QuickSpin/Exposure\_QuickSpin.cpp, 1343  
src/Exposure\_QuickSpin/resource.h, 1284  
src/Exposure\_QuickSpin/stdafx.cpp, 1289  
src/Exposure\_QuickSpin/stdafx.h, 1302  
src/Exposure\_QuickSpin/targetver.h, 1318  
src/FileAccess/FileAccess\_QuickSpin/FileAccess\_QuickSpin.cpp, 1345  
src/FileAccess\_QuickSpin/resource.h, 1284  
src/FileAccess\_QuickSpin/stdafx.cpp, 1290  
src/FileAccess\_QuickSpin/stdafx.h, 1303  
src/FileAccess\_QuickSpin/targetver.h, 1319  
src/GentLInfo\_GentLInfo\_QuickSpin/GenTLInfo\_QuickSpin.cpp, 1348  
src/GenTLInfo\_QuickSpin/resource.h, 1284  
src/GenTLInfo\_QuickSpin/stdafx.h, 1321  
src/GigEVisionPerformance/CpuUtil.cpp, 1349  
src/GigEVisionPerformance/CpuUtil.h, 1350  
src/GigEVisionPerformance/GigEVisionPerformance.cpp, 1351  
src/GigEVisionPerformance/GigEVisionPerformance.h, 1356  
src/GigEVisionPerformance/resource.h, 1284  
src/GigEVisionPerformance/stdafx.cpp, 1291  
src/GigEVisionPerformance/stdafx.h, 1304  
src/GigEVisionPerformance/targetver.h, 1320  
src/HighDynamicRange/HighDynamicRange.cpp, 1357  
src/HighDynamicRange/resource.h, 1284  
src/ImageEvents/ImageEvents.cpp, 1360  
src/ImageEvents/resource.h, 1284  
src/ImageFormatControl/ImageFormatControl.cpp, 1362  
src/ImageFormatControl/resource.h, 1284  
src/ImageFormatControl/stdafx.h, 1305  
src/ImageFormatControl\_QuickSpin/ImageFormatControl\_QuickSpin.cpp, 1363  
src/ImageFormatControl\_QuickSpin/resource.h, 1284  
src/ImageFormatControl\_QuickSpin/stdafx.h, 1305  
src/Inference/Inference.cpp, 1364  
src/Inference/resource.h, 1284  
src/Logging/Logging.cpp, 1371  
src/Logging/resource.h, 1284  
src/LogicBlock/LogicBlock.cpp, 1372

src/LogicBlock/resource.h, 1284  
 src/LookupTable/LookupTable.cpp, 1374  
 src/LookupTable/resource.h, 1284  
 src/NodeMapCallback/NodeMapCallback.cpp, 1375  
 src/NodeMapCallback/resource.h, 1284  
 src/NodeMapInfo/NodeMapInfo.cpp, 1377  
 src/NodeMapInfo/resource.h, 1284  
 src/NodeMapInfo/stdafx.cpp, 1291  
 src/NodeMapInfo/stdafx.h, 1306  
 src/NodeMapInfo/targetver.h, 1321  
 src/Polarization/Polarization.cpp, 1381  
 src/Polarization/resource.h, 1284  
 src/Polarization/stdafx.h, 1307  
 src/SaveToAvi/resource.h, 1284  
 src/SaveToAvi/SaveToAvi.cpp, 1384  
 src/Sequencer/resource.h, 1284  
 src/Sequencer/Sequencer.cpp, 1386  
 src/Sequencer/stdafx.cpp, 1292  
 src/Sequencer/stdafx.h, 1307  
 src/Sequencer/targetver.h, 1322  
 src/SerialRxTx/resource.h, 1284  
 src/SerialRxTx/SerialRxTx.cpp, 1388  
 src/SerialRxTx/stdafx.cpp, 1292  
 src/SerialRxTx/stdafx.h, 1308  
 src/SerialRxTx/targetver.h, 1323  
 src/Trigger/resource.h, 1284  
 src/Trigger/Trigger.cpp, 1391  
 src/Trigger\_QuickSpin/resource.h, 1284  
 src/Trigger\_QuickSpin/Trigger\_QuickSpin.cpp, 1394  
**Standard**  
 Spinnaker::GenApi, 351  
**STANDARD\_CAST\_TO\_SPINNAKER**  
 ExceptionHandling.cpp, 1341  
**STANDARD\_EXCEPTION**  
 ExceptionHandling.cpp, 1341  
**StartCpuTracing**  
 CpuUtil, 188  
**StartPerformanceCounter**  
 PerformanceCounter, 189  
**StartRecording**  
 PortNode, 1015  
 PortRecorder, 1018  
**StartSecondsCounter**  
 SecondsCounter, 190  
**StartStream**  
 IDataStream, 811  
**startTime**  
 SecondsCounter, 190  
**StatisticsChannel**  
 Spinnaker, 313  
**Status**  
 ActionCommandResult, 393  
**StopCpuTracing**  
 CpuUtil, 188  
**StopRecording**  
 PortNode, 1015  
 PortRecorder, 1019  
 Spinnaker::GenApi, 378  
**StopStream**  
 IDataStream, 811  
**StoreToBag**  
 CFeatureBag, 627  
**Stream**  
 EventHandler, 739  
 IImage, 834  
 Image, 882  
**StreamAnnounceBufferMinimum**  
 TransportLayerStream, 1088  
**StreamAnnouncedBufferCount**  
 TransportLayerStream, 1088  
**StreamBlockTransferSize**  
 TransportLayerStream, 1088  
**StreamBufferAlignment**  
 TransportLayerStream, 1089  
**StreamBufferCountManual**  
 TransportLayerStream, 1089  
**StreamBufferCountMax**  
 TransportLayerStream, 1089  
**StreamBufferCountMode**  
 TransportLayerStream, 1089  
**StreamBufferCountMode\_Auto**  
 Spinnaker, 314  
**StreamBufferCountMode\_Manual**  
 Spinnaker, 313  
**StreamBufferCountModeEnum**  
 Spinnaker, 313  
**StreamBufferCountResult**  
 TransportLayerStream, 1089  
**StreamBufferHandlingMode**  
 TransportLayerStream, 1089  
**StreamBufferHandlingMode\_NewestFirst**  
 Spinnaker, 314  
**StreamBufferHandlingMode\_NewestOnly**  
 Spinnaker, 314  
**StreamBufferHandlingMode\_OldestFirst**  
 Spinnaker, 314  
**StreamBufferHandlingMode\_OldestFirstOverwrite**  
 Spinnaker, 314  
**StreamBufferHandlingModeEnum**  
 Spinnaker, 314  
**StreamChannelId**  
 GVCP\_EVENT\_ITEM, 774  
 GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776  
**StreamChunkCountMaximum**  
 TransportLayerStream, 1090  
**StreamCRCCheckEnable**  
 TransportLayerStream, 1090  
**StreamDeliveredFrameCount**  
 TransportLayerStream, 1090  
**StreamFailedBufferCount**  
 TransportLayerStream, 1090  
**StreamID**  
 TransportLayerStream, 1090  
**StreamInputBufferCount**  
 TransportLayerStream, 1090  
**StreamIsGrabbing**

TransportLayerStream, 1091  
StreamLostFrameCount  
    TransportLayerStream, 1091  
StreamOutputBufferCount  
    TransportLayerStream, 1091  
StreamStartedFrameCount  
    TransportLayerStream, 1091  
StreamType  
    TransportLayerStream, 1091  
StreamType\_CameraLink  
    Spinnaker, 315  
StreamType\_CameraLinkHS  
    Spinnaker, 315  
StreamType\_CoaxPress  
    Spinnaker, 315  
StreamType\_Custom  
    Spinnaker, 315  
StreamType\_GigEVision  
    Spinnaker, 315  
StreamType\_USB3Vision  
    Spinnaker, 315  
StreamTypeEnum  
    Spinnaker, 314  
StringList\_t  
    Spinnaker::GenApi, 347  
StringNode, 1036  
    ~StringNode, 1038  
    GetMaxLength, 1038  
    GetValue, 1038  
    operator\*, 1039  
    operator(), 1039  
    operator=, 1039  
    SetReference, 1039  
    SetValue, 1039  
    StringNode, 1038  
StringNode Class, 174  
StringRegNode, 1040  
    ~StringRegNode, 1042  
    SetReference, 1042  
    StringRegNode, 1041, 1042  
StringRegNode Class, 175  
StructPort Class, 176  
SubMinor  
    Version\_t, 1106  
SUBNET\_MASK\_INVALID  
    AdapterConfig, 184  
subnetLength  
    IpInfo, 954  
subnetMask  
    IpInfo, 954  
substr  
    gcstring, 770  
swap  
    gcstring, 770  
sync  
    ODevFileStreamBuf< CharType, Traits >, 1007  
Synch Class, 177  
System, 1043  
    ~System, 1045  
    GetCameras, 1045  
    GetInstance, 1046  
    GetInterfaces, 1046  
    GetLibraryVersion, 1046  
    GetLoggingEventPriorityLevel, 1047  
    GetTLNodeMap, 1047  
    IsInUse, 1047  
    RegisterEventHandler, 1048  
    RegisterInterfaceEventHandler, 1048  
    RegisterLoggingEventHandler, 1048  
    ReleaseInstance, 1049  
    SendActionCommand, 1049  
    SetLoggingEventPriorityLevel, 1050  
    System, 1045  
    TransportLayerSystem, 1094  
    UnregisterAllLoggingEventHandlers, 1050  
    UnregisterEventHandler, 1051  
    UnregisterInterfaceEventHandler, 1051  
    UnregisterLoggingEventHandler, 1051  
    UpdateCameras, 1052  
    UpdateInterfaceList, 1052  
System Class, 77  
System.h  
    FLIR\_SPINNAKER\_VERSION\_BUILD, 1275  
    FLIR\_SPINNAKER\_VERSION\_MAJOR, 1275  
    FLIR\_SPINNAKER\_VERSION\_MINOR, 1276  
    FLIR\_SPINNAKER\_VERSION\_TYPE, 1276  
SystemEventHandler, 1053  
    ~SystemEventHandler, 1054  
    OnInterfaceArrival, 1054  
    OnInterfaceRemoval, 1055  
    operator=, 1055  
    SystemEventHandler, 1054  
SystemEventHandler Class, 78  
SystemEventHandlerImpl, 1056  
    ~SystemEventHandlerImpl, 1057  
    LockEventHandlerMutex, 1057  
    OnInterfaceArrival, 1057  
    OnInterfaceRemoval, 1058  
    RegisterAllInterfaceEvents, 1058  
    RegisterInterfaceEventToSystem, 1058  
    SystemEventHandlerImpl, 1057  
    UnlockEventHandlerMutex, 1058  
    UnregisterAllInterfaceEvents, 1058  
    UnregisterInterfaceEventFromSystem, 1058  
SystemImpl  
    IInterface, 845  
    InterfaceList, 946  
    LoggingEventData, 972  
SystemPtr, 1059  
    ~SystemPtr, 1060  
    SystemPtr, 1060  
SystemPtr Class, 79  
SystemPtrInternal  
    ISystem, 960  
    TransportLayerSystem, 1094  
TAG\_EVENT\_CMD

Spinnaker::GenApi, 354  
 TAG\_EVENTDATA\_CMD  
     Spinnaker::GenApi, 354  
 Test0001  
     Camera, 548  
 TestDuration  
     GigEVisionPerformance.cpp, 1356  
 TestEventGenerate  
     Camera, 549  
 TestPattern  
     Camera, 549  
 TestPattern\_Increment  
     Spinnaker, 315  
 TestPattern\_Off  
     Spinnaker, 315  
 TestPattern\_SensorTestPattern  
     Spinnaker, 315  
 TestPatternEnums  
     Spinnaker, 315  
 TestPatternGeneratorSelector  
     Camera, 549  
 TestPatternGeneratorSelector\_PipelineStart  
     Spinnaker, 315  
 TestPatternGeneratorSelector\_Sensor  
     Spinnaker, 315  
 TestPatternGeneratorSelectorEnums  
     Spinnaker, 315  
 TestPendingAck  
     Camera, 549  
 ThrowBadAlloc  
     Spinnaker::GenICam, 390  
 TIFF  
     Spinnaker, 283  
 TIFFOption, 1061  
     ADOBEDFLATE, 1062  
     CCITTFAZ3, 1062  
     CCITTFAZ4, 1062  
     compression, 1062  
     CompressionMethod, 1061  
     DEFLATE, 1062  
     JPEG, 1062  
     LZW, 1062  
     NONE, 1062  
     PACKBITS, 1062  
     reserved, 1062  
     TIFFOption, 1062  
 timeDiff  
     SecondsCounter, 190  
 TimerDelay  
     Camera, 549  
 TimerDuration  
     Camera, 550  
 TimerReset  
     Camera, 550  
 TimerSelector  
     Camera, 550  
 TimerSelector\_Timer0  
     Spinnaker, 317  
 TimerSelector\_Timer1  
     Spinnaker, 317  
 TimerSelector\_Timer2  
     Spinnaker, 317  
 TimerSelectorEnums  
     Spinnaker, 315  
 TimerStatus  
     Camera, 550  
 TimerStatus\_TimerActive  
     Spinnaker, 317  
 TimerStatus\_TimerCompleted  
     Spinnaker, 317  
 TimerStatus\_TimerIdle  
     Spinnaker, 317  
 TimerStatus\_TimerTriggerWait  
     Spinnaker, 317  
 TimerStatusEnums  
     Spinnaker, 317  
 TimerTriggerActivation  
     Camera, 550  
 TimerTriggerActivation\_AnyEdge  
     Spinnaker, 317  
 TimerTriggerActivation\_FallingEdge  
     Spinnaker, 317  
 TimerTriggerActivation\_LevelHigh  
     Spinnaker, 317  
 TimerTriggerActivation\_LevelLow  
     Spinnaker, 317  
 TimerTriggerActivation\_RisingEdge  
     Spinnaker, 317  
 TimerTriggerActivationEnums  
     Spinnaker, 317  
 TimerTriggerSource  
     Camera, 550  
 TimerTriggerSource\_AcquisitionEnd  
     Spinnaker, 318  
 TimerTriggerSource\_AcquisitionStart  
     Spinnaker, 318  
 TimerTriggerSource\_AcquisitionTrigger  
     Spinnaker, 318  
 TimerTriggerSource\_Action0  
     Spinnaker, 318  
 TimerTriggerSource\_Action1  
     Spinnaker, 319  
 TimerTriggerSource\_Action2  
     Spinnaker, 319  
 TimerTriggerSource\_Counter0End  
     Spinnaker, 318  
 TimerTriggerSource\_Counter0Start  
     Spinnaker, 318  
 TimerTriggerSource\_Counter1End  
     Spinnaker, 318  
 TimerTriggerSource\_Counter1Start  
     Spinnaker, 318  
 TimerTriggerSource\_Counter2End  
     Spinnaker, 318  
 TimerTriggerSource\_Counter2Start  
     Spinnaker, 318

TimerTriggerSource\_Encoder0  
    Spinnaker, 318  
TimerTriggerSource\_Encoder1  
    Spinnaker, 318  
TimerTriggerSource\_Encoder2  
    Spinnaker, 318  
TimerTriggerSource\_ExposureEnd  
    Spinnaker, 318  
TimerTriggerSource\_ExposureStart  
    Spinnaker, 318  
TimerTriggerSource\_FrameBurstEnd  
    Spinnaker, 318  
TimerTriggerSource\_FrameBurstStart  
    Spinnaker, 318  
TimerTriggerSource\_FrameEnd  
    Spinnaker, 318  
TimerTriggerSource\_FrameStart  
    Spinnaker, 318  
TimerTriggerSource\_FrameTrigger  
    Spinnaker, 318  
TimerTriggerSource\_Line0  
    Spinnaker, 318  
TimerTriggerSource\_Line1  
    Spinnaker, 318  
TimerTriggerSource\_Line2  
    Spinnaker, 318  
TimerTriggerSource\_LineEnd  
    Spinnaker, 318  
TimerTriggerSource\_LineStart  
    Spinnaker, 318  
TimerTriggerSource\_LineTrigger  
    Spinnaker, 318  
TimerTriggerSource\_LinkTrigger0  
    Spinnaker, 319  
TimerTriggerSource\_LinkTrigger1  
    Spinnaker, 319  
TimerTriggerSource\_LinkTrigger2  
    Spinnaker, 319  
TimerTriggerSource\_Off  
    Spinnaker, 318  
TimerTriggerSource\_SoftwareSignal0  
    Spinnaker, 318  
TimerTriggerSource\_SoftwareSignal1  
    Spinnaker, 318  
TimerTriggerSource\_SoftwareSignal2  
    Spinnaker, 318  
TimerTriggerSource\_Timer0End  
    Spinnaker, 318  
TimerTriggerSource\_Timer0Start  
    Spinnaker, 318  
TimerTriggerSource\_Timer1End  
    Spinnaker, 318  
TimerTriggerSource\_Timer1Start  
    Spinnaker, 318  
TimerTriggerSource\_Timer2End  
    Spinnaker, 318  
TimerTriggerSource\_Timer2Start  
    Spinnaker, 318

TimerTriggerSource\_UserOutput0  
    Spinnaker, 318  
TimerTriggerSource\_UserOutput1  
    Spinnaker, 318  
TimerTriggerSource\_UserOutput2  
    Spinnaker, 318  
TimerTriggerSourceEnums  
    Spinnaker, 317  
TimerValue  
    Camera, 551  
Timestamp  
    Camera, 551  
    U3V\_EVENT\_DATA, 1101  
TimestampHigh  
    GVCP\_EVENT\_ITEM, 774  
    GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776  
TimestampLatch  
    Camera, 551  
TimestampLatchValue  
    Camera, 551  
TimestampLow  
    GVCP\_EVENT\_ITEM, 774  
    GVCP\_EVENT\_ITEM\_EXTENDED\_ID, 776  
TimestampReset  
    Camera, 551  
TLDevice  
    ICameraBase, 792  
TLD displayName  
    TransportLayerSystem, 1097  
TLFileName  
    TransportLayerSystem, 1097  
TLID  
    TransportLayerSystem, 1097  
TLInterface  
    IInterface, 845  
TLModelName  
    TransportLayerSystem, 1098  
TLP ParamsLocked  
    Camera, 551  
TLP Path  
    TransportLayerSystem, 1098  
TLStream  
    ICameraBase, 792  
TLS System  
    ISystem, 960  
TLType  
    TransportLayerSystem, 1098  
TLType\_CameraLink  
    Spinnaker, 319  
TLType\_CameraLinkHS  
    Spinnaker, 319  
TLType\_CoaxPress  
    Spinnaker, 319  
TLType\_Custom  
    Spinnaker, 319  
TLType\_GigEVision  
    Spinnaker, 319  
TLType\_Mixed

Spinnaker, 319  
TLType\_USB3Vision  
    Spinnaker, 319  
TLTypeEnum  
    Spinnaker, 319  
TLVendorName  
    TransportLayerSystem, 1098  
TLVersion  
    TransportLayerSystem, 1098  
ToggleHDRMode  
    HighDynamicRange.cpp, 1358  
Tokenize  
    Spinnaker::GenICam, 390  
topLeftXCoord  
    InferenceBoxRect, 919  
    InferenceBoxRotatedRect, 920  
topLeftYCoord  
    InferenceBoxRect, 919  
    InferenceBoxRotatedRect, 920  
ToString  
    CNodeMapFactory, 665  
    CSelectorSet, 697  
    EAccessModeClass, 716  
    ECachingModeClass, 717  
    EDisplayNotationClass, 718  
    EEndianessClass, 719  
    EGenApiSchemaVersionClass, 720  
    EInputDirectionClass, 721  
    ENamespaceClass, 722  
    ERepresentationClass, 732  
    ESignClass, 733  
    ESlopeClass, 734  
    EStandardNameSpaceClass, 735  
    EVisibilityClass, 740  
    EYesNoClass, 746  
    Spinnaker::GenApi, 378  
    ValueNode, 1105  
ToXml  
    CNodeMapFactory, 665  
TransferAbort  
    Camera, 552  
TransferBlockCount  
    Camera, 552  
TransferBurstCount  
    Camera, 552  
TransferComponentSelector  
    Camera, 552  
TransferComponentSelector\_All  
    Spinnaker, 319  
TransferComponentSelector\_Blue  
    Spinnaker, 319  
TransferComponentSelector\_Green  
    Spinnaker, 319  
TransferComponentSelector\_Red  
    Spinnaker, 319  
TransferComponentSelectorEnums  
    Spinnaker, 319  
TransferControlMode  
    Camera, 552  
    TransferControlMode\_Automatic  
        Spinnaker, 320  
    TransferControlMode\_Basic  
        Spinnaker, 320  
    TransferControlMode\_UserControlled  
        Spinnaker, 320  
    TransferControlModeEnums  
        Spinnaker, 320  
    TransferOperationMode  
        Camera, 552  
    TransferOperationMode\_Continuous  
        Spinnaker, 320  
    TransferOperationMode\_MultiBlock  
        Spinnaker, 320  
    TransferOperationModeEnums  
        Spinnaker, 320  
    TransferPause  
        Camera, 553  
    TransferQueueCurrentBlockCount  
        Camera, 553  
    TransferQueueMaxBlockCount  
        Camera, 553  
    TransferQueueMode  
        Camera, 553  
    TransferQueueMode\_FirstInFirstOut  
        Spinnaker, 320  
    TransferQueueModeEnums  
        Spinnaker, 320  
    TransferQueueOverflowCount  
        Camera, 553  
    TransferResume  
        Camera, 553  
    TransferSelector  
        Camera, 554  
    TransferSelector\_All  
        Spinnaker, 321  
    TransferSelector\_Stream0  
        Spinnaker, 321  
    TransferSelector\_Stream1  
        Spinnaker, 321  
    TransferSelector\_Stream2  
        Spinnaker, 321  
    TransferSelectorEnums  
        Spinnaker, 320  
    TransferStart  
        Camera, 554  
    TransferStatus  
        Camera, 554  
    TransferStatusSelector  
        Camera, 554  
    TransferStatusSelector\_Paused  
        Spinnaker, 321  
    TransferStatusSelector\_QueueOverflow  
        Spinnaker, 321  
    TransferStatusSelector\_Stopped  
        Spinnaker, 321  
    TransferStatusSelector\_Stopping

Spinnaker, 321  
TransferStatusSelector\_Streaming  
    Spinnaker, 321  
TransferStatusSelectorEnums  
    Spinnaker, 321  
TransferStop  
    Camera, 554  
TransferStreamChannel  
    Camera, 554  
TransferTriggerActivation  
    Camera, 555  
TransferTriggerActivation\_AnyEdge  
    Spinnaker, 321  
TransferTriggerActivation\_FallingEdge  
    Spinnaker, 321  
TransferTriggerActivation\_LevelHigh  
    Spinnaker, 322  
TransferTriggerActivation\_LevelLow  
    Spinnaker, 322  
TransferTriggerActivation\_RisingEdge  
    Spinnaker, 321  
TransferTriggerActivationEnums  
    Spinnaker, 321  
TransferTriggerMode  
    Camera, 555  
TransferTriggerMode\_Off  
    Spinnaker, 322  
TransferTriggerMode\_On  
    Spinnaker, 322  
TransferTriggerModeEnums  
    Spinnaker, 322  
TransferTriggerSelector  
    Camera, 555  
TransferTriggerSelector\_TransferAbort  
    Spinnaker, 322  
TransferTriggerSelector\_TransferActive  
    Spinnaker, 322  
TransferTriggerSelector\_TransferBurstStart  
    Spinnaker, 322  
TransferTriggerSelector\_TransferBurstStop  
    Spinnaker, 322  
TransferTriggerSelector\_TransferPause  
    Spinnaker, 322  
TransferTriggerSelector\_TransferResume  
    Spinnaker, 322  
TransferTriggerSelector\_TransferStart  
    Spinnaker, 322  
TransferTriggerSelector\_TransferStop  
    Spinnaker, 322  
TransferTriggerSelectorEnums  
    Spinnaker, 322  
TransferTriggerSource  
    Camera, 555  
TransferTriggerSource\_Action0  
    Spinnaker, 323  
TransferTriggerSource\_Action1  
    Spinnaker, 323  
TransferTriggerSource\_Action2  
    Spinnaker, 323  
TransferTriggerSource\_Counter0End  
    Spinnaker, 323  
TransferTriggerSource\_Counter0Start  
    Spinnaker, 323  
TransferTriggerSource\_Counter1End  
    Spinnaker, 323  
TransferTriggerSource\_Counter1Start  
    Spinnaker, 323  
TransferTriggerSource\_Counter2End  
    Spinnaker, 323  
TransferTriggerSource\_Counter2Start  
    Spinnaker, 323  
TransferTriggerSource\_Line0  
    Spinnaker, 323  
TransferTriggerSource\_Line1  
    Spinnaker, 323  
TransferTriggerSource\_Line2  
    Spinnaker, 323  
TransferTriggerSource\_SoftwareSignal0  
    Spinnaker, 323  
TransferTriggerSource\_SoftwareSignal1  
    Spinnaker, 323  
TransferTriggerSource\_SoftwareSignal2  
    Spinnaker, 323  
TransferTriggerSource\_Timer0End  
    Spinnaker, 323  
TransferTriggerSource\_Timer0Start  
    Spinnaker, 323  
TransferTriggerSource\_Timer1End  
    Spinnaker, 323  
TransferTriggerSource\_Timer1Start  
    Spinnaker, 323  
TransferTriggerSource\_Timer2End  
    Spinnaker, 323  
TransferTriggerSource\_Timer2Start  
    Spinnaker, 323  
TransferTriggerSourceEnums  
    Spinnaker, 322  
transmitBuffers  
    AdapterInfo, 398  
transmitBuffersMax  
    AdapterInfo, 398  
transmitBuffersMin  
    AdapterInfo, 398  
transmitBuffersRegKey  
    AdapterInfo, 398  
transmitBuffersStep  
    AdapterInfo, 398  
TransportLayerDefs Class, 81  
TransportLayerDevice, 1063  
    ~TransportLayerDevice, 1065  
    CameraBase, 1065  
    CameraInternal, 1065  
    DeviceAccessStatus, 1066  
    DeviceCurrentSpeed, 1066  
    DeviceDisplayName, 1066  
    DeviceDriverVersion, 1066

DeviceEndianessMechanism, 1066  
 DeviceID, 1067  
 DeviceInstanceId, 1067  
 DeviceIsUpdater, 1067  
 DeviceLinkSpeed, 1067  
 DeviceLocation, 1067  
 DevicemodelName, 1067  
 DeviceMulticastMonitorMode, 1068  
 DeviceSerialNumber, 1068  
 DeviceType, 1068  
 DeviceU3VProtocol, 1068  
 DeviceUserID, 1068  
 DeviceVendorName, 1068  
 DeviceVersion, 1069  
 GenICamXMLLocation, 1069  
 GenICamXMLPath, 1069  
 GevCCP, 1069  
 GevDeviceAutoForceIP, 1069  
 GevDeviceDiscoverMaximumPacketSize, 1069  
 GevDeviceForceGateway, 1070  
 GevDeviceForceIP, 1070  
 GevDeviceForceIPAddress, 1070  
 GevDeviceForceSubnetMask, 1070  
 GevDeviceGateway, 1070  
 GevDeviceIPAddress, 1070  
 GevDeviceIsWrongSubnet, 1071  
 GevDeviceMACAddress, 1071  
 GevDeviceMaximumPacketSize, 1071  
 GevDeviceMaximumRetryCount, 1071  
 GevDeviceModelsBigEndian, 1071  
 GevDevicePort, 1071  
 GevDeviceReadWriteTimeout, 1072  
 GevDeviceSubnetMask, 1072  
 GevVersionMajor, 1072  
 GevVersionMinor, 1072  
 GUIXMLLocation, 1072  
 GUIXMLPath, 1072  
 ICameraBase, 1066  
 TransportLayerDevice, 1065  
 TransportLayerDevice Class, 82  
 TransportLayerInterface, 1073  
 ~TransportLayerInterface, 1075  
 ActionCommand, 1076  
 DeviceAccessStatus, 1076  
 DeviceCount, 1077  
 DeviceID, 1077  
 DevicemodelName, 1077  
 DeviceSelector, 1077  
 DeviceSerialNumber, 1077  
 DeviceUnlock, 1077  
 DeviceUpdateList, 1078  
 DeviceVendorName, 1078  
 FilterDriverStatus, 1078  
 GevActionDeviceKey, 1078  
 GevActionGroupKey, 1078  
 GevActionGroupMask, 1078  
 GevActionTime, 1079  
 GevDeviceAutoForceIP, 1079  
 GevDeviceForceGateway, 1079  
 GevDeviceForceIP, 1079  
 GevDeviceForceIPAddress, 1079  
 GevDeviceForceSubnetMask, 1079  
 GevDeviceGateway, 1080  
 GevDeviceIPAddress, 1080  
 GevDeviceMACAddress, 1080  
 GevDeviceSubnetMask, 1080  
 GevInterfaceGateway, 1080  
 GevInterfaceGatewaySelector, 1080  
 GevInterfaceMACAddress, 1081  
 GevInterfaceMTU, 1081  
 GevInterfaceReceiveLinkSpeed, 1081  
 GevInterfaceSubnetIPAddress, 1081  
 GevInterfaceSubnetMask, 1081  
 GevInterfaceSubnetSelector, 1081  
 GevInterfaceTransmitLinkSpeed, 1082  
 HostAdapterDriverVersion, 1082  
 HostAdapterName, 1082  
 HostAdapterVendor, 1082  
 IInterface, 1076  
 IncompatibleDeviceCount, 1082  
 IncompatibleDeviceID, 1082  
 IncompatibleDevicemodelName, 1083  
 IncompatibleDeviceSelector, 1083  
 IncompatibleDeviceVendorName, 1083  
 IncompatibleGevDeviceIPAddress, 1083  
 IncompatibleGevDeviceMACAddress, 1083  
 IncompatibleGevDeviceSubnetMask, 1083  
 Interface, 1076  
 InterfaceDisplayName, 1084  
 InterfaceID, 1084  
 InterfaceInternal, 1076  
 InterfaceType, 1084  
 POEStatus, 1084  
 TransportLayerInterface, 1075, 1076  
 TransportLayerInterface Class, 83  
 TransportLayerStream, 1084  
 ~TransportLayerStream, 1086  
 CameraBase, 1087  
 CameraInternal, 1087  
 GevFailedPacketCount, 1087  
 GevMaximumNumberResendRequests, 1087  
 GevPacketResendMode, 1087  
 GevPacketResendTimeout, 1087  
 GevResendPacketCount, 1088  
 GevResendRequestCount, 1088  
 GevTotalPacketCount, 1088  
 ICameraBase, 1087  
 StreamAnnounceBufferMinimum, 1088  
 StreamAnnouncedBufferCount, 1088  
 StreamBlockTransferSize, 1088  
 StreamBufferAlignment, 1089  
 StreamBufferCountManual, 1089  
 StreamBufferCountMax, 1089  
 StreamBufferCountMode, 1089  
 StreamBufferCountResult, 1089  
 StreamBufferHandlingMode, 1089

StreamChunkCountMaximum, 1090  
StreamCRCCheckEnable, 1090  
StreamDeliveredFrameCount, 1090  
StreamFailedBufferCount, 1090  
StreamID, 1090  
StreamInputBufferCount, 1090  
StreamIsGrabbing, 1091  
StreamLostFrameCount, 1091  
StreamOutputBufferCount, 1091  
StreamStartedFrameCount, 1091  
StreamType, 1091  
TransportLayerStream, 1086  
TransportLayerStream Class, 84  
TransportLayerStreamInfo  
    IDataStream, 811  
TransportLayerSystem, 1092  
    ~TransportLayerSystem, 1094  
    EnumerateGEVInterfaces, 1094  
    GenTLSFNCVersionMajor, 1095  
    GenTLSFNCVersionMinor, 1095  
    GenTLSFNCVersionSubMinor, 1095  
    GenTLVersionMajor, 1095  
    GenTLVersionMinor, 1095  
    GevInterfaceDefaultGateway, 1095  
    GevInterfaceDefaultIPAddress, 1096  
    GevInterfaceDefaultSubnetMask, 1096  
    GevInterfaceMACAddress, 1096  
    GevVersionMajor, 1096  
    GevVersionMinor, 1096  
    InterfaceDisplayName, 1096  
    InterfaceID, 1097  
    InterfaceSelector, 1097  
    InterfaceUpdateList, 1097  
    ISystem, 1094  
    System, 1094  
    SystemPtrInternal, 1094  
    TLDisplayName, 1097  
    TLFileName, 1097  
    TLID, 1097  
    TLmodelName, 1098  
    TLPath, 1098  
    TLType, 1098  
    TLVendorName, 1098  
    TLVersion, 1098  
    TransportLayerSystem, 1093, 1094  
TransportLayerSystem Class, 85  
Trigger.cpp  
    AcquireImages, 1393  
    chosenTrigger, 1394  
    ConfigureTrigger, 1393  
    GrabNextImageByTrigger, 1393  
    HARDWARE, 1393  
    main, 1393  
    PrintDeviceInfo, 1393  
    ResetTrigger, 1393  
    RunSingleCamera, 1394  
    SOFTWARE, 1393  
    triggerType, 1392  
Trigger\_QuickSpin.cpp  
    AcquireImages, 1395  
    chosenTrigger, 1396  
    ConfigureTrigger, 1395  
    GrabNextImageByTrigger, 1395  
    HARDWARE, 1395  
    main, 1395  
    PrintDeviceInfo, 1396  
    ResetTrigger, 1396  
    RunSingleCamera, 1396  
    SOFTWARE, 1395  
    triggerType, 1395  
TriggerActivation  
    Camera, 555  
TriggerActivation\_AnyEdge  
    Spinnaker, 324  
TriggerActivation\_FallingEdge  
    Spinnaker, 324  
TriggerActivation\_LevelHigh  
    Spinnaker, 324  
TriggerActivation\_LevelLow  
    Spinnaker, 324  
TriggerActivation\_RisingEdge  
    Spinnaker, 324  
TriggerActivationEnums  
    Spinnaker, 324  
TriggerDelay  
    Camera, 555  
TriggerDivider  
    Camera, 556  
TriggerEventTest  
    Camera, 556  
TriggerMode  
    Camera, 556  
TriggerMode\_Off  
    Spinnaker, 324  
TriggerMode\_On  
    Spinnaker, 324  
TriggerModeEnums  
    Spinnaker, 324  
TriggerMultiplier  
    Camera, 556  
TriggerOverlap  
    Camera, 556  
TriggerOverlap\_Off  
    Spinnaker, 324  
TriggerOverlap\_PreviousFrame  
    Spinnaker, 324  
TriggerOverlap\_ReadOut  
    Spinnaker, 324  
TriggerOverlapEnums  
    Spinnaker, 324  
TriggerSelector  
    Camera, 557  
TriggerSelector\_AcquisitionStart  
    Spinnaker, 325  
TriggerSelector\_FrameBurstStart  
    Spinnaker, 325

**TriggerSelector\_FrameStart**  
 Spinnaker, 325  
**TriggerSelectorEnums**  
 Spinnaker, 324  
**TriggerSoftware**  
 Camera, 557  
**TriggerSource**  
 Camera, 557  
**TriggerSource\_Action0**  
 Spinnaker, 325  
**TriggerSource\_Counter0End**  
 Spinnaker, 325  
**TriggerSource\_Counter0Start**  
 Spinnaker, 325  
**TriggerSource\_Counter1End**  
 Spinnaker, 325  
**TriggerSource\_Counter1Start**  
 Spinnaker, 325  
**TriggerSource\_Line0**  
 Spinnaker, 325  
**TriggerSource\_Line1**  
 Spinnaker, 325  
**TriggerSource\_Line2**  
 Spinnaker, 325  
**TriggerSource\_Line3**  
 Spinnaker, 325  
**TriggerSource\_LogicBlock0**  
 Spinnaker, 325  
**TriggerSource\_LogicBlock1**  
 Spinnaker, 325  
**TriggerSource\_Software**  
 Spinnaker, 325  
**TriggerSource\_UserOutput0**  
 Spinnaker, 325  
**TriggerSource\_UserOutput1**  
 Spinnaker, 325  
**TriggerSource\_UserOutput2**  
 Spinnaker, 325  
**TriggerSource\_UserOutput3**  
 Spinnaker, 325  
**TriggerSourceEnums**  
 Spinnaker, 325  
**triggerType**  
 Trigger.cpp, 1392  
 Trigger\_QuickSpin.cpp, 1395  
**TryLock**  
 CGlobalLock, 632  
 CLock, 649, 651  
**TWO\_SECOND\_DELAY**  
 SerialRxTx.cpp, 1389  
**type**  
 LibraryVersion, 966  
**Types Enums**, 179  
**Types.h**  
 \_UndefinedRepresentation, 1264  
 interface, 1264  
**U3V\_CHUNK\_TRAILER**, 1099  
 ChunkID, 1099  
 ChunkLength, 1099  
**U3V\_COMMAND\_HEADER**, 1099  
 CommandId, 1100  
 Flags, 1100  
 Length, 1100  
 Prefix, 1100  
 ReqId, 1100  
**U3V\_EVENT\_DATA**, 1100  
 EventId, 1101  
 Reserved, 1101  
 Timestamp, 1101  
**U3V\_EVENT\_MESSAGE**, 1101  
 CommandHeader, 1102  
 EventData, 1102  
**U3V\_EVENT\_PREFIX**  
 Spinnaker::GenApi, 385  
**UNCOMPRESSED**  
 SaveToAvi.cpp, 1384  
**underflow**  
 IDevFileStreamBuf< CharType, Traits >, 816  
**UNKNOWN\_PIXELFORMAT**  
 Spinnaker, 295  
**Unlock**  
 CGlobalLock, 633  
 CLock, 649, 651  
**UnlockEarly**  
 CGlobalLockUnlocker, 634  
**UnlockEventHandlerMutex**  
 SystemEventHandlerImpl, 1058  
**UnregisterAllInterfaceEvents**  
 SystemEventHandlerImpl, 1058  
**UnregisterAllLoggingEventHandlers**  
 ISystem, 959  
 System, 1050  
**UnregisterEventHandler**  
 CameraBase, 575  
 ICameraBase, 791  
 IInterface, 844  
 Interface, 933  
 ISystem, 959  
 System, 1051  
**UnregisterImageEventHandler**  
 IDataStream, 811  
**UnregisterInterfaceEventFromSystem**  
 SystemEventHandlerImpl, 1058  
**UnregisterInterfaceEventHandler**  
 ISystem, 959  
 System, 1051  
**UnregisterLoggingEventHandler**  
 ISystem, 960  
 System, 1051  
**Unsigned**  
 Spinnaker::GenApi, 351  
**UpdateBuffer**  
 CChunkAdapter, 588  
 CChunkPort, 603  
**UpdateCameras**  
 IInterface, 844

Interface, 933  
ISystem, 960  
System, 1052  
UpdateFirmware  
    SpinUpdate.h, 1271  
UpdateFirmwareConsole  
    SpinUpdate.h, 1271  
UpdateFirmwareGUI  
    SpinUpdate.h, 1272  
UpdateInterfaceList  
    ISystem, 960  
    System, 1052  
UpdatorMessageCallback  
    SpinUpdate.h, 1272  
UpdatorProgressCallback  
    SpinUpdate.h, 1272  
UploadFileToCamera  
    Inference.cpp, 1369  
UploadImage  
     FileAccess\_QuickSpin.cpp, 1347  
UrlDecode  
    Spinnaker::GenICam, 391  
UrlEncode  
    Spinnaker::GenICam, 391  
USB  
    Spinnaker::GenApi, 352  
USE\_TEMP\_CACHE\_FILE  
    GCUtilities.h, 1211  
UseDuration  
    GigEVisionPerformance.cpp, 1356  
UseMaxFramerate  
    GigEVisionPerformance.cpp, 1356  
UserOutputSelector  
    Camera, 557  
UserOutputSelector\_UserOutput0  
    Spinnaker, 326  
UserOutputSelector\_UserOutput1  
    Spinnaker, 326  
UserOutputSelector\_UserOutput2  
    Spinnaker, 326  
UserOutputSelector\_UserOutput3  
    Spinnaker, 326  
UserOutputSelectorEnums  
    Spinnaker, 325  
UserOutputValue  
    Camera, 557  
UserOutputValueAll  
    Camera, 558  
UserOutputValueAllMask  
    Camera, 558  
UserSetDefault  
    Camera, 558  
UserSetDefault\_Default  
    Spinnaker, 326  
UserSetDefault\_UserSet0  
    Spinnaker, 326  
UserSetDefault\_UserSet1  
    Spinnaker, 326  
UserSetDefaultEnums  
    Spinnaker, 326  
UserSetFeatureEnable  
    Camera, 558  
UserSetFramerate  
    GigEVisionPerformance.cpp, 1356  
UserSetLoad  
    Camera, 558  
UserSetSave  
    Camera, 559  
UserSetSelector  
    Camera, 559  
UserSetSelector\_Default  
    Spinnaker, 326  
UserSetSelector\_UserSet0  
    Spinnaker, 326  
UserSetSelector\_UserSet1  
    Spinnaker, 326  
UserSetSelectorEnums  
    Spinnaker, 326  
v1\_0  
    Spinnaker::GenApi, 349  
v1\_1  
    Spinnaker::GenApi, 349  
V3\_3Enable  
    Camera, 559  
VALID\_SUBNET\_NOT\_FOUND  
    AdapterConfig, 184  
ValidateIpAddress  
    AdapterConfig, 187  
VALUE  
    NodeMapInfo.cpp, 1378  
ValueNode, 1102  
    ~ValueNode, 1104  
    FromString, 1104  
    GetNode, 1104  
    IsValueCacheValid, 1104  
    SetReference, 1105  
    ToString, 1105  
    ValueNode, 1103, 1104  
ValueNode Class, 180  
Varying  
    Spinnaker::GenApi, 352  
Verify  
    Spinnaker::GenApi, 385  
Version\_t, 1105  
    Major, 1106  
    Minor, 1106  
    SubMinor, 1106  
videoType  
    SaveToAvi.cpp, 1384  
WaitForImages  
    ImageEvents.cpp, 1361  
WaitOnImageEvent  
    IDataStream, 812  
WEIGHTED\_DIRECTIONAL\_FILTER  
    Spinnaker, 246

what  
    Exception, [745](#)

WhiteClip  
    Camera, [559](#)

WhiteClipSelector  
    Camera, [560](#)

WhiteClipSelector\_All  
    Spinnaker, [327](#)

WhiteClipSelector\_Blue  
    Spinnaker, [327](#)

WhiteClipSelector\_Green  
    Spinnaker, [327](#)

WhiteClipSelector\_Red  
    Spinnaker, [327](#)

WhiteClipSelector\_Tap1  
    Spinnaker, [327](#)

WhiteClipSelector\_Tap2  
    Spinnaker, [327](#)

WhiteClipSelector\_U  
    Spinnaker, [327](#)

WhiteClipSelector\_V  
    Spinnaker, [327](#)

WhiteClipSelector\_Y  
    Spinnaker, [327](#)

WhiteClipSelectorEnums  
    Spinnaker, [326](#)

Width  
    Camera, [560](#)

width  
    H264Option, [783](#)

WidthMax  
    Camera, [560](#)

WO  
    Spinnaker::GenApi, [347](#)

Write  
    CChunkPort, [603](#)  
    CEventPort, [624](#)  
    CPortImpl, [686](#)  
    CPortWriteList, [689](#)  
    CRegisterPortImpl, [693](#)  
    CTestPortStruct< CDataStruct >, [700](#)  
    PortNode, [1015](#)  
    PortRecorder, [1019](#)  
    PortReplay, [1023](#)  
    Spinnaker::GenApi, [379](#)

write  
    FileProtocolAdapter, [750](#)

WriteAround  
    Spinnaker::GenApi, [348](#)

WritePort  
    CameraBase, [575](#)  
    ICameraBase, [791](#)

WriteRegister  
    CRegisterPortImpl, [694](#)

WriteThrough  
    Spinnaker::GenApi, [348](#)

xsputn  
    ODevFileStreamBuf< CharType, Traits >, [1007](#)