

Spinnaker C++

1.23.0.27

Generated by Doxygen 1.8.13

Contents

| | | |
|----------|---------------------------------------|-----------|
| 1 | Introduction | 1 |
| 2 | Software Licensing Information | 3 |
| 3 | Module Index | 5 |
| 3.1 | Modules | 5 |
| 4 | Namespace Index | 9 |
| 4.1 | Namespace List | 9 |
| 5 | Hierarchical Index | 11 |
| 5.1 | Class Hierarchy | 11 |
| 6 | Class Index | 17 |
| 6.1 | Class List | 17 |
| 7 | File Index | 23 |
| 7.1 | File List | 23 |
| 8 | Module Documentation | 27 |
| 8.1 | Spinnaker Event Classes | 27 |
| 8.1.1 | Detailed Description | 28 |
| 8.2 | ArrivalEvent Class | 29 |
| 8.2.1 | Detailed Description | 29 |
| 8.3 | Spinnaker Classes | 30 |
| 8.3.1 | Detailed Description | 32 |
| 8.4 | AVI Recorder Class | 33 |

| | | |
|----------|--------------------------------------|----|
| 8.4.1 | Detailed Description | 33 |
| 8.4.2 | Function Documentation | 33 |
| 8.4.2.1 | DEPRECATED_CLASS() | 33 |
| 8.5 | BasePtr Class | 36 |
| 8.5.1 | Detailed Description | 36 |
| 8.6 | Camera Class | 37 |
| 8.6.1 | Detailed Description | 37 |
| 8.7 | Camera Base Class | 38 |
| 8.7.1 | Detailed Description | 38 |
| 8.8 | CameraDefs Class | 39 |
| 8.8.1 | Detailed Description | 71 |
| 8.8.2 | Enumeration Type Documentation | 71 |
| 8.8.2.1 | AcquisitionModeEnums | 71 |
| 8.8.2.2 | AcquisitionStatusSelectorEnums | 71 |
| 8.8.2.3 | ActionUnconditionalModeEnums | 72 |
| 8.8.2.4 | AdcBitDepthEnums | 72 |
| 8.8.2.5 | AutoAlgorithmSelectorEnums | 72 |
| 8.8.2.6 | AutoExposureControlPriorityEnums | 73 |
| 8.8.2.7 | AutoExposureLightingModeEnums | 73 |
| 8.8.2.8 | AutoExposureMeteringModeEnums | 73 |
| 8.8.2.9 | AutoExposureTargetGreyValueAutoEnums | 74 |
| 8.8.2.10 | BalanceRatioSelectorEnums | 74 |
| 8.8.2.11 | BalanceWhiteAutoEnums | 75 |
| 8.8.2.12 | BalanceWhiteAutoProfileEnums | 75 |
| 8.8.2.13 | BinningHorizontalModeEnums | 75 |
| 8.8.2.14 | BinningSelectorEnums | 76 |
| 8.8.2.15 | BinningVerticalModeEnums | 76 |
| 8.8.2.16 | BlackLevelAutoBalanceEnums | 76 |
| 8.8.2.17 | BlackLevelAutoEnums | 77 |
| 8.8.2.18 | BlackLevelSelectorEnums | 77 |

| | | |
|----------|---|----|
| 8.8.2.19 | ChunkBlackLevelSelectorEnums | 77 |
| 8.8.2.20 | ChunkCounterSelectorEnums | 78 |
| 8.8.2.21 | ChunkEncoderSelectorEnums | 78 |
| 8.8.2.22 | ChunkEncoderStatusEnums | 78 |
| 8.8.2.23 | ChunkExposureTimeSelectorEnums | 78 |
| 8.8.2.24 | ChunkGainSelectorEnums | 79 |
| 8.8.2.25 | ChunkImageComponentEnums | 79 |
| 8.8.2.26 | ChunkPixelFormatEnums | 80 |
| 8.8.2.27 | ChunkRegionIDEnums | 80 |
| 8.8.2.28 | ChunkScan3dCoordinateReferenceSelectorEnums | 81 |
| 8.8.2.29 | ChunkScan3dCoordinateSelectorEnums | 81 |
| 8.8.2.30 | ChunkScan3dCoordinateSystemEnums | 81 |
| 8.8.2.31 | ChunkScan3dCoordinateSystemReferenceEnums | 82 |
| 8.8.2.32 | ChunkScan3dCoordinateTransformSelectorEnums | 82 |
| 8.8.2.33 | ChunkScan3dDistanceUnitEnums | 82 |
| 8.8.2.34 | ChunkScan3dOutputModeEnums | 83 |
| 8.8.2.35 | ChunkSelectorEnums | 84 |
| 8.8.2.36 | ChunkSourceIDEnums | 84 |
| 8.8.2.37 | ChunkTimerSelectorEnums | 84 |
| 8.8.2.38 | ChunkTransferStreamIDEnums | 85 |
| 8.8.2.39 | CICongfigurationEnums | 85 |
| 8.8.2.40 | CITimeSlotsCountEnums | 86 |
| 8.8.2.41 | ColorTransformationSelectorEnums | 86 |
| 8.8.2.42 | ColorTransformationValueSelectorEnums | 86 |
| 8.8.2.43 | CounterEventActivationEnums | 87 |
| 8.8.2.44 | CounterEventSourceEnums | 87 |
| 8.8.2.45 | CounterResetActivationEnums | 88 |
| 8.8.2.46 | CounterResetSourceEnums | 88 |
| 8.8.2.47 | CounterSelectorEnums | 89 |
| 8.8.2.48 | CounterStatusEnums | 89 |

| | | |
|----------|------------------------------------|-----|
| 8.8.2.49 | CounterTriggerActivationEnums | 89 |
| 8.8.2.50 | CounterTriggerSourceEnums | 90 |
| 8.8.2.51 | CxpConnectionTestModeEnums | 90 |
| 8.8.2.52 | CxpLinkConfigurationEnums | 90 |
| 8.8.2.53 | CxpLinkConfigurationPreferredEnums | 91 |
| 8.8.2.54 | CxpLinkConfigurationStatusEnums | 92 |
| 8.8.2.55 | CxpPoCxpStatusEnums | 93 |
| 8.8.2.56 | DecimationHorizontalModeEnums | 94 |
| 8.8.2.57 | DecimationSelectorEnums | 94 |
| 8.8.2.58 | DecimationVerticalModeEnums | 94 |
| 8.8.2.59 | DefectCorrectionModeEnums | 95 |
| 8.8.2.60 | DeinterlacingEnums | 95 |
| 8.8.2.61 | DeviceCharacterSetEnums | 95 |
| 8.8.2.62 | DeviceClockSelectorEnums | 96 |
| 8.8.2.63 | DeviceConnectionStatusEnums | 96 |
| 8.8.2.64 | DeviceIndicatorModeEnums | 96 |
| 8.8.2.65 | DeviceLinkHeartbeatModeEnums | 96 |
| 8.8.2.66 | DeviceLinkThroughputLimitModeEnums | 98 |
| 8.8.2.67 | DevicePowerSupplySelectorEnums | 98 |
| 8.8.2.68 | DeviceRegistersEndiannessEnums | 98 |
| 8.8.2.69 | DeviceScanTypeEnums | 99 |
| 8.8.2.70 | DeviceSerialPortBaudRateEnums | 99 |
| 8.8.2.71 | DeviceSerialPortSelectorEnums | 99 |
| 8.8.2.72 | DeviceStreamChannelEndiannessEnums | 100 |
| 8.8.2.73 | DeviceStreamChannelTypeEnums | 100 |
| 8.8.2.74 | DeviceTapGeometryEnums | 100 |
| 8.8.2.75 | DeviceTemperatureSelectorEnums | 101 |
| 8.8.2.76 | DeviceTLTypeEnums | 102 |
| 8.8.2.77 | DeviceTypeEnums | 102 |
| 8.8.2.78 | EncoderModeEnums | 102 |

| | | |
|-----------|--|-----|
| 8.8.2.79 | EncoderOutputModeEnums | 103 |
| 8.8.2.80 | EncoderResetActivationEnums | 103 |
| 8.8.2.81 | EncoderResetSourceEnums | 104 |
| 8.8.2.82 | EncoderSelectorEnums | 105 |
| 8.8.2.83 | EncoderSourceAEnums | 105 |
| 8.8.2.84 | EncoderSourceBEnums | 105 |
| 8.8.2.85 | EncoderStatusEnums | 106 |
| 8.8.2.86 | EventNotificationEnums | 106 |
| 8.8.2.87 | EventSelectorEnums | 106 |
| 8.8.2.88 | ExposureActiveModeEnums | 107 |
| 8.8.2.89 | ExposureAutoEnums | 107 |
| 8.8.2.90 | ExposureModeEnums | 107 |
| 8.8.2.91 | ExposureTimeModeEnums | 108 |
| 8.8.2.92 | ExposureTimeSelectorEnums | 108 |
| 8.8.2.93 | FileOpenModeEnums | 109 |
| 8.8.2.94 | FileOperationSelectorEnums | 109 |
| 8.8.2.95 | FileOperationStatusEnums | 109 |
| 8.8.2.96 | FileSelectorEnums | 110 |
| 8.8.2.97 | GainAutoBalanceEnums | 110 |
| 8.8.2.98 | GainAutoEnums | 110 |
| 8.8.2.99 | GainSelectorEnums | 111 |
| 8.8.2.100 | GevCCPEnums | 111 |
| 8.8.2.101 | GevCurrentPhysicalLinkConfigurationEnums | 111 |
| 8.8.2.102 | GevGVCPExtendedStatusCodesSelectorEnums | 112 |
| 8.8.2.103 | GevGVSPExtendedIDModeEnums | 112 |
| 8.8.2.104 | GevIEEE1588ClockAccuracyEnums | 112 |
| 8.8.2.105 | GevIEEE1588ModeEnums | 112 |
| 8.8.2.106 | GevIEEE1588StatusEnums | 113 |
| 8.8.2.107 | GevIPConfigurationStatusEnums | 113 |
| 8.8.2.108 | GevPhysicalLinkConfigurationEnums | 114 |

| | | |
|-----------|--|-----|
| 8.8.2.109 | GevSupportedOptionSelectorEnums | 114 |
| 8.8.2.110 | ImageComponentSelectorEnums | 115 |
| 8.8.2.111 | ImageCompressionJPEGFormatOptionEnums | 115 |
| 8.8.2.112 | ImageCompressionModeEnums | 116 |
| 8.8.2.113 | ImageCompressionRateOptionEnums | 116 |
| 8.8.2.114 | LineFormatEnums | 116 |
| 8.8.2.115 | LineInputFilterSelectorEnums | 117 |
| 8.8.2.116 | LineModeEnums | 117 |
| 8.8.2.117 | LineSelectorEnums | 117 |
| 8.8.2.118 | LineSourceEnums | 118 |
| 8.8.2.119 | LogicBlockLUTInputActivationEnums | 118 |
| 8.8.2.120 | LogicBlockLUTInputSelectorEnums | 119 |
| 8.8.2.121 | LogicBlockLUTInputSourceEnums | 119 |
| 8.8.2.122 | LogicBlockLUTSelectorEnums | 120 |
| 8.8.2.123 | LogicBlockSelectorEnums | 120 |
| 8.8.2.124 | LUTSelectorEnums | 120 |
| 8.8.2.125 | PixelColorFilterEnums | 121 |
| 8.8.2.126 | PixelFormatEnums | 121 |
| 8.8.2.127 | PixelFormatInfoSelectorEnums | 127 |
| 8.8.2.128 | PixelSizeEnums | 132 |
| 8.8.2.129 | RegionDestinationEnums | 133 |
| 8.8.2.130 | RegionModeEnums | 133 |
| 8.8.2.131 | RegionSelectorEnums | 134 |
| 8.8.2.132 | RgbTransformLightSourceEnums | 134 |
| 8.8.2.133 | Scan3dCoordinateReferenceSelectorEnums | 135 |
| 8.8.2.134 | Scan3dCoordinateSelectorEnums | 135 |
| 8.8.2.135 | Scan3dCoordinateSystemEnums | 135 |
| 8.8.2.136 | Scan3dCoordinateSystemReferenceEnums | 136 |
| 8.8.2.137 | Scan3dCoordinateTransformSelectorEnums | 136 |
| 8.8.2.138 | Scan3dDistanceUnitEnums | 136 |

| | |
|---|-----|
| 8.8.2.139 Scan3dOutputModeEnums | 137 |
| 8.8.2.140 SensorDigitizationTapsEnums | 137 |
| 8.8.2.141 SensorShutterModeEnums | 138 |
| 8.8.2.142 SensorTapsEnums | 138 |
| 8.8.2.143 SequencerConfigurationModeEnums | 139 |
| 8.8.2.144 SequencerConfigurationValidEnums | 139 |
| 8.8.2.145 SequencerModeEnums | 139 |
| 8.8.2.146 SequencerSetValidEnums | 139 |
| 8.8.2.147 SequencerTriggerActivationEnums | 140 |
| 8.8.2.148 SequencerTriggerSourceEnums | 140 |
| 8.8.2.149 SerialPortBaudRateEnums | 140 |
| 8.8.2.150 SerialPortParityEnums | 141 |
| 8.8.2.151 SerialPortSelectorEnums | 141 |
| 8.8.2.152 SerialPortSourceEnums | 142 |
| 8.8.2.153 SerialPortStopBitsEnums | 142 |
| 8.8.2.154 SoftwareSignalSelectorEnums | 142 |
| 8.8.2.155 SourceSelectorEnums | 143 |
| 8.8.2.156 TestPatternEnums | 143 |
| 8.8.2.157 TestPatternGeneratorSelectorEnums | 143 |
| 8.8.2.158 TimerSelectorEnums | 144 |
| 8.8.2.159 TimerStatusEnums | 144 |
| 8.8.2.160 TimerTriggerActivationEnums | 144 |
| 8.8.2.161 TimerTriggerSourceEnums | 145 |
| 8.8.2.162 TransferComponentSelectorEnums | 146 |
| 8.8.2.163 TransferControlModeEnums | 146 |
| 8.8.2.164 TransferOperationModeEnums | 147 |
| 8.8.2.165 TransferQueueModeEnums | 147 |
| 8.8.2.166 TransferSelectorEnums | 147 |
| 8.8.2.167 TransferStatusSelectorEnums | 148 |
| 8.8.2.168 TransferTriggerActivationEnums | 148 |

| | |
|--|-----|
| 8.8.2.169 TransferTriggerModeEnums | 148 |
| 8.8.2.170 TransferTriggerSelectorEnums | 149 |
| 8.8.2.171 TransferTriggerSourceEnums | 149 |
| 8.8.2.172 TriggerActivationEnums | 150 |
| 8.8.2.173 TriggerModeEnums | 151 |
| 8.8.2.174 TriggerOverlapEnums | 151 |
| 8.8.2.175 TriggerSelectorEnums | 151 |
| 8.8.2.176 TriggerSourceEnums | 152 |
| 8.8.2.177 UserOutputSelectorEnums | 152 |
| 8.8.2.178 UserSetDefaultEnums | 152 |
| 8.8.2.179 UserSetSelectorEnums | 153 |
| 8.8.2.180 WhiteClipSelectorEnums | 153 |
| 8.9 Camera List Class | 154 |
| 8.9.1 Detailed Description | 154 |
| 8.10 CameraPtr Class | 155 |
| 8.10.1 Detailed Description | 155 |
| 8.10.2 Function Documentation | 155 |
| 8.10.2.1 CameraPtr() [1/4] | 155 |
| 8.10.2.2 CameraPtr() [2/4] | 155 |
| 8.10.2.3 CameraPtr() [3/4] | 156 |
| 8.10.2.4 CameraPtr() [4/4] | 156 |
| 8.11 ChunkData Class | 157 |
| 8.11.1 Detailed Description | 157 |
| 8.12 DeviceEvent Class | 158 |
| 8.12.1 Detailed Description | 158 |
| 8.13 Event Class | 159 |
| 8.13.1 Detailed Description | 159 |
| 8.14 Exception Class | 160 |
| 8.14.1 Detailed Description | 160 |
| 8.15 Image Class | 161 |

| | |
|---|-----|
| 8.15.1 Detailed Description | 161 |
| 8.16 ImageEvent Class | 162 |
| 8.16.1 Detailed Description | 162 |
| 8.17 ImagePtr Class | 163 |
| 8.17.1 Detailed Description | 163 |
| 8.18 ImageStatistics Class | 164 |
| 8.18.1 Detailed Description | 164 |
| 8.19 Image Utility Class | 165 |
| 8.19.1 Detailed Description | 165 |
| 8.20 Image Utility Heatmap Class | 166 |
| 8.20.1 Detailed Description | 166 |
| 8.21 Image Utility Polarization Class | 167 |
| 8.21.1 Detailed Description | 167 |
| 8.22 Interface Class | 168 |
| 8.22.1 Detailed Description | 168 |
| 8.23 InterfaceEvent Class | 169 |
| 8.23.1 Detailed Description | 169 |
| 8.24 InterfaceList Class | 170 |
| 8.24.1 Detailed Description | 170 |
| 8.25 InterfacePtr Class | 171 |
| 8.25.1 Detailed Description | 171 |
| 8.26 LoggingEvent Class | 172 |
| 8.26.1 Detailed Description | 172 |
| 8.27 Logging Event Class | 173 |
| 8.27.1 Detailed Description | 173 |
| 8.28 LoggingEventDataPtr Class | 174 |
| 8.28.1 Detailed Description | 174 |
| 8.29 RemovalEvent Class | 175 |
| 8.29.1 Detailed Description | 175 |
| 8.30 Spinnaker Headers | 176 |

| | |
|---|-----|
| 8.30.1 Detailed Description | 177 |
| 8.30.2 Variable Documentation | 177 |
| 8.30.2.1 EVENT_TIMEOUT_INFINITE | 177 |
| 8.30.2.2 EVENT_TIMEOUT_NONE | 177 |
| 8.31 Spinnaker.h | 178 |
| 8.32 Spinnaker Definitions | 179 |
| 8.32.1 Detailed Description | 182 |
| 8.32.2 Enumeration Type Documentation | 182 |
| 8.32.2.1 ActionCommandStatus | 183 |
| 8.32.2.2 BufferOwnership | 183 |
| 8.32.2.3 ColorProcessingAlgorithm | 183 |
| 8.32.2.4 Error | 184 |
| 8.32.2.5 EventType | 185 |
| 8.32.2.6 ImageFileFormat | 185 |
| 8.32.2.7 ImageStatus | 186 |
| 8.32.2.8 PayloadTypeInfoIDs | 186 |
| 8.32.2.9 PixelFormatIntType | 187 |
| 8.32.2.10 PixelFormatNamespaceID | 187 |
| 8.32.2.11 SpinnakerLogLevel | 188 |
| 8.32.2.12 StatisticsChannel | 188 |
| 8.32.3 Function Documentation | 189 |
| 8.32.3.1 DEPRECATED_CLASS() [1/3] | 189 |
| 8.32.3.2 DEPRECATED_CLASS() [2/3] | 189 |
| 8.32.3.3 DEPRECATED_CLASS() [3/3] | 189 |
| 8.33 Spinnaker Platform | 190 |
| 8.33.1 Detailed Description | 190 |
| 8.33.2 Macro Definition Documentation | 190 |
| 8.33.2.1 SPINNAKER_API | 190 |
| 8.33.2.2 SPINNAKER_API_ABSTRACT | 190 |
| 8.33.2.3 SPINNAKER_LOCAL | 190 |

| | |
|--|-----|
| 8.34 Spinnaker Video Class | 191 |
| 8.34.1 Detailed Description | 191 |
| 8.35 Spinnaker Video Definitions | 192 |
| 8.36 System Class | 193 |
| 8.36.1 Detailed Description | 193 |
| 8.37 SystemPtr Class | 194 |
| 8.37.1 Detailed Description | 194 |
| 8.38 Spinnaker QuickSpin Classes | 195 |
| 8.38.1 Detailed Description | 195 |
| 8.39 TransportLayerDefs Class | 196 |
| 8.39.1 Detailed Description | 197 |
| 8.39.2 Enumeration Type Documentation | 197 |
| 8.39.2.1 DeviceAccessStatusEnum | 197 |
| 8.39.2.2 DeviceCurrentSpeedEnum | 198 |
| 8.39.2.3 DeviceEndiannessMechanismEnum | 198 |
| 8.39.2.4 DeviceTypeEnum | 198 |
| 8.39.2.5 FilterDriverStatusEnum | 199 |
| 8.39.2.6 GenICamXMLLocationEnum | 199 |
| 8.39.2.7 GevCCPEnum | 200 |
| 8.39.2.8 GUIXMLLocationEnum | 200 |
| 8.39.2.9 POEStatusEnum | 200 |
| 8.39.2.10 StreamBufferCountModeEnum | 200 |
| 8.39.2.11 StreamBufferHandlingModeEnum | 201 |
| 8.39.2.12 StreamDefaultBufferCountModeEnum | 201 |
| 8.39.2.13 StreamTypeEnum | 202 |
| 8.40 TransportLayerDevice Class | 203 |
| 8.40.1 Detailed Description | 203 |
| 8.41 TransportLayerInterface Class | 204 |
| 8.41.1 Detailed Description | 204 |
| 8.42 TransportLayerStream Class | 205 |

| | |
|--|-----|
| 8.42.1 Detailed Description | 205 |
| 8.43 TransportLayerSystem Class | 206 |
| 8.43.1 Detailed Description | 206 |
| 8.44 Camera Base Interface Class | 207 |
| 8.44.1 Detailed Description | 207 |
| 8.45 IChunkData Class | 208 |
| 8.45.1 Detailed Description | 208 |
| 8.46 IImage Class | 209 |
| 8.46.1 Detailed Description | 209 |
| 8.47 IImageStatistics Class | 210 |
| 8.47.1 Detailed Description | 210 |
| 8.48 IInterface Class | 211 |
| 8.48.1 Detailed Description | 211 |
| 8.49 IInterfaceList Class | 212 |
| 8.49.1 Detailed Description | 212 |
| 8.50 ISystem Class | 213 |
| 8.50.1 Detailed Description | 213 |
| 8.51 Spinnaker GenApi Classes | 214 |
| 8.51.1 Detailed Description | 220 |
| 8.51.2 Typedef Documentation | 220 |
| 8.51.2.1 CNodeMapRef | 220 |
| 8.51.2.2 CNodeRef | 220 |
| 8.51.2.3 CSelectorRef | 220 |
| 8.51.3 Function Documentation | 220 |
| 8.51.3.1 _ClearXMLCache() | 220 |
| 8.51.3.2 _Connect() [1/2] | 221 |
| 8.51.3.3 _Connect() [2/2] | 221 |
| 8.51.3.4 _Destroy() | 221 |
| 8.51.3.5 _GetDeviceName() | 221 |
| 8.51.3.6 _GetNode() | 221 |

| | | |
|-----------|---|-----|
| 8.51.3.7 | _GetNodes() | 221 |
| 8.51.3.8 | _GetSupportedSchemaVersions() | 221 |
| 8.51.3.9 | _InvalidateNodes() | 222 |
| 8.51.3.10 | _LoadXMLFromFile() | 222 |
| 8.51.3.11 | _LoadXMLFromFileInject() | 222 |
| 8.51.3.12 | _LoadXMLFromString() | 222 |
| 8.51.3.13 | _LoadXMLFromStringInject() | 222 |
| 8.51.3.14 | _LoadXMLFromZIPData() | 222 |
| 8.51.3.15 | _LoadXMLFromZIPFile() | 222 |
| 8.51.3.16 | _Poll() | 223 |
| 8.51.3.17 | CastToIDestroy() | 223 |
| 8.51.3.18 | CNodeMapRefT() [1/3] | 223 |
| 8.51.3.19 | CNodeMapRefT() [2/3] | 223 |
| 8.51.3.20 | CNodeMapRefT() [3/3] | 223 |
| 8.51.3.21 | EatComments() | 223 |
| 8.51.3.22 | operator<<() | 224 |
| 8.51.3.23 | operator=() [1/2] | 224 |
| 8.51.3.24 | operator=() [2/2] | 224 |
| 8.51.3.25 | operator>>() | 224 |
| 8.51.3.26 | ~CNodeMapRefT() | 224 |
| 8.52 | AutoVector Class | 225 |
| 8.52.1 | Detailed Description | 225 |
| 8.53 | Spinnaker GenApi Interfaces | 226 |
| 8.53.1 | Detailed Description | 227 |
| 8.53.2 | Typedef Documentation | 227 |
| 8.53.2.1 | CallbackHandleType | 227 |
| 8.53.2.2 | NodeList_t | 227 |
| 8.54 | IBase Interface | 228 |
| 8.54.1 | Detailed Description | 228 |
| 8.54.2 | Variable Documentation | 228 |

| | |
|--|-----|
| 8.54.2.1 IBase | 228 |
| 8.55 BooleanNode Class | 229 |
| 8.55.1 Detailed Description | 229 |
| 8.55.2 Typedef Documentation | 229 |
| 8.55.2.1 CBooleanRef | 229 |
| 8.56 CategoryNode Class | 230 |
| 8.56.1 Detailed Description | 230 |
| 8.56.2 Typedef Documentation | 230 |
| 8.56.2.1 CCategoryRef | 230 |
| 8.57 ChunkAdapter Class | 231 |
| 8.57.1 Detailed Description | 231 |
| 8.58 ChunkAdapterDcam Class | 232 |
| 8.58.1 Detailed Description | 232 |
| 8.59 ChunkAdapterGeneric Class | 233 |
| 8.59.1 Detailed Description | 233 |
| 8.60 ChunkAdapterGEV Class | 234 |
| 8.60.1 Detailed Description | 234 |
| 8.61 ChunkPort Class | 235 |
| 8.61.1 Detailed Description | 235 |
| 8.62 CommandNode Class | 236 |
| 8.62.1 Detailed Description | 236 |
| 8.62.2 Typedef Documentation | 236 |
| 8.62.2.1 CCommandRef | 236 |
| 8.63 Container Class | 237 |
| 8.64 Counter Class | 238 |
| 8.64.1 Detailed Description | 238 |
| 8.65 EnumClasses Class | 239 |
| 8.65.1 Detailed Description | 240 |
| 8.66 EnumEntryNode Class | 241 |
| 8.66.1 Detailed Description | 241 |

| | | |
|----------|---------------------------|-----|
| 8.66.2 | Typedef Documentation | 241 |
| 8.66.2.1 | CEnumEntryRef | 241 |
| 8.67 | EnumNode Class | 242 |
| 8.67.1 | Detailed Description | 242 |
| 8.67.2 | Typedef Documentation | 242 |
| 8.67.2.1 | CEnumerationRef | 242 |
| 8.68 | EnumNodeT Class | 243 |
| 8.68.1 | Detailed Description | 243 |
| 8.69 | EventAdapter Class | 244 |
| 8.69.1 | Detailed Description | 244 |
| 8.70 | EventAdapter1394 Class | 245 |
| 8.70.1 | Detailed Description | 245 |
| 8.71 | EventAdapterGeneric Class | 246 |
| 8.71.1 | Detailed Description | 246 |
| 8.72 | EventAdapterGEV Class | 247 |
| 8.72.1 | Detailed Description | 247 |
| 8.73 | EventAdapterU3V Class | 248 |
| 8.73.1 | Detailed Description | 248 |
| 8.74 | EventPort Class | 249 |
| 8.74.1 | Detailed Description | 249 |
| 8.75 | Filestream Class | 250 |
| 8.75.1 | Detailed Description | 250 |
| 8.76 | FloatNode Class | 251 |
| 8.76.1 | Detailed Description | 251 |
| 8.76.2 | Typedef Documentation | 251 |
| 8.76.2.1 | CFloatRef | 251 |
| 8.77 | FloatRegNode Class | 252 |
| 8.77.1 | Detailed Description | 252 |
| 8.78 | GCString Class | 253 |
| 8.78.1 | Detailed Description | 253 |

| | | |
|-----------|---------------------------------------|-----|
| 8.79 | GCSynch Class | 254 |
| 8.79.1 | Detailed Description | 254 |
| 8.80 | GCTypes Class | 255 |
| 8.80.1 | Detailed Description | 255 |
| 8.80.2 | Typedef Documentation | 255 |
| 8.80.2.1 | float32_t | 255 |
| 8.80.2.2 | float64_t | 255 |
| 8.81 | Spinnaker GenApi Utilities | 256 |
| 8.81.1 | Detailed Description | 256 |
| 8.82 | GCUtilities Utility | 257 |
| 8.82.1 | Detailed Description | 258 |
| 8.82.2 | Function Documentation | 258 |
| 8.82.2.1 | DoesEnvironmentVariableExist() | 258 |
| 8.82.2.2 | GetFiles() | 258 |
| 8.82.2.3 | GetGenICamCacheFolder() | 258 |
| 8.82.2.4 | GetGenICamCLProtocolFolder() | 259 |
| 8.82.2.5 | GetGenICamLogConfig() | 259 |
| 8.82.2.6 | GetModulePathFromFunction() | 259 |
| 8.82.2.7 | GetValueOfEnvironmentVariable() [1/2] | 259 |
| 8.82.2.8 | GetValueOfEnvironmentVariable() [2/2] | 260 |
| 8.82.2.9 | INTEGRAL_CAST() | 260 |
| 8.82.2.10 | INTEGRAL_CAST2() | 260 |
| 8.82.2.11 | ReplaceEnvironmentVariables() | 260 |
| 8.82.2.12 | SetGenICamCacheFolder() | 261 |
| 8.82.2.13 | SetGenICamCLProtocolFolder() | 261 |
| 8.82.2.14 | SetGenICamLogConfig() | 261 |
| 8.82.2.15 | Tokenize() | 261 |
| 8.82.2.16 | UrlDecode() | 261 |
| 8.82.2.17 | UrlEncode() | 262 |
| 8.83 | IBoolean Interface | 263 |

| | |
|--|-----|
| 8.83.1 Detailed Description | 263 |
| 8.83.2 Function Documentation | 263 |
| 8.83.2.1 GetValue() | 263 |
| 8.83.2.2 operator>() | 264 |
| 8.83.2.3 operator=() | 264 |
| 8.83.3 Variable Documentation | 264 |
| 8.83.3.1 IBoolean | 264 |
| 8.83.3.2 Verify | 264 |
| 8.84 ICategory Interfaces | 265 |
| 8.84.1 Detailed Description | 265 |
| 8.84.2 Variable Documentation | 265 |
| 8.84.2.1 ICategory | 265 |
| 8.85 IChunkPort Interface | 266 |
| 8.85.1 Detailed Description | 266 |
| 8.85.2 Macro Definition Documentation | 266 |
| 8.85.2.1 CHUNK_BASE_ADDRESS_REGISTER | 266 |
| 8.85.2.2 CHUNK_BASE_ADDRESS_REGISTER_LEN | 267 |
| 8.85.2.3 CHUNK_LENGTH_REGISTER | 267 |
| 8.85.2.4 CHUNK_LENGTH_REGISTER_LEN | 267 |
| 8.85.3 Function Documentation | 267 |
| 8.85.3.1 CacheChunkData() | 267 |
| 8.85.4 Variable Documentation | 267 |
| 8.85.4.1 IChunkPort | 267 |
| 8.86 ICommand Interface | 268 |
| 8.86.1 Detailed Description | 268 |
| 8.86.2 Function Documentation | 268 |
| 8.86.2.1 IsDone() | 268 |
| 8.86.3 Variable Documentation | 268 |
| 8.86.3.1 ICommand | 269 |
| 8.87 IDestroy Interface | 270 |

| | |
|---|-----|
| 8.87.1 Detailed Description | 270 |
| 8.87.2 Variable Documentation | 270 |
| 8.87.2.1 IDestroy | 270 |
| 8.88 IDeviceInfo Interface | 271 |
| 8.88.1 Detailed Description | 271 |
| 8.88.2 Function Documentation | 271 |
| 8.88.2.1 GetDeviceVersion() | 271 |
| 8.88.2.2 GetGenApiVersion() | 272 |
| 8.88.2.3 GetProductGuid() | 272 |
| 8.88.2.4 GetSchemaVersion() | 272 |
| 8.88.2.5 GetStandardNameSpace() | 272 |
| 8.88.2.6 GetToolTip() | 272 |
| 8.88.2.7 GetVendorName() | 272 |
| 8.88.2.8 GetVersionGuid() | 273 |
| 8.88.3 Variable Documentation | 273 |
| 8.88.3.1 IDeviceInfo | 273 |
| 8.89 IEnumEntry Interface | 274 |
| 8.89.1 Detailed Description | 274 |
| 8.89.2 Function Documentation | 274 |
| 8.89.2.1 GetNumericValue() | 274 |
| 8.89.2.2 GetSymbolic() | 274 |
| 8.89.2.3 IsSelfClearing() | 275 |
| 8.89.3 Variable Documentation | 275 |
| 8.89.3.1 IEnumEntry | 275 |
| 8.90 IEnumeration Interface | 276 |
| 8.90.1 Detailed Description | 276 |
| 8.90.2 Function Documentation | 276 |
| 8.90.2.1 GetCurrentEntry() | 276 |
| 8.90.2.2 GetEntries() | 277 |
| 8.90.2.3 GetEntry() | 277 |

| | | |
|-----------|-------------------------|-----|
| 8.90.2.4 | GetEntryByName() | 277 |
| 8.90.2.5 | GetIntValue() | 277 |
| 8.90.2.6 | operator*() | 277 |
| 8.90.2.7 | SetIntValue() | 278 |
| 8.90.3 | Variable Documentation | 278 |
| 8.90.3.1 | IEnumeration | 278 |
| 8.91 | IEnumerationT Interface | 279 |
| 8.91.1 | Detailed Description | 279 |
| 8.91.2 | Function Documentation | 279 |
| 8.91.2.1 | GetEntry() | 279 |
| 8.91.2.2 | operator=() [1/2] | 280 |
| 8.91.2.3 | operator=() [2/2] | 280 |
| 8.91.3 | Variable Documentation | 280 |
| 8.91.3.1 | IEnumerationT | 280 |
| 8.91.3.2 | IEnumReference | 280 |
| 8.92 | IFloat Interface | 281 |
| 8.92.1 | Detailed Description | 282 |
| 8.92.2 | Function Documentation | 282 |
| 8.92.2.1 | GetDisplayNotation() | 282 |
| 8.92.2.2 | GetDisplayPrecision() | 282 |
| 8.92.2.3 | GetInc() | 282 |
| 8.92.2.4 | GetIncMode() | 282 |
| 8.92.2.5 | GetListOfValidValues() | 282 |
| 8.92.2.6 | GetMax() | 283 |
| 8.92.2.7 | GetMin() | 283 |
| 8.92.2.8 | GetRepresentation() | 283 |
| 8.92.2.9 | GetUnit() | 283 |
| 8.92.2.10 | HasInc() | 283 |
| 8.92.2.11 | ImposeMax() | 283 |
| 8.92.2.12 | ImposeMin() | 284 |

| | |
|-------------------------------|-----|
| 8.92.2.13 operator=() | 284 |
| 8.92.3 Variable Documentation | 284 |
| 8.92.3.1 IFloat | 284 |
| 8.93 Integer Interface | 285 |
| 8.93.1 Detailed Description | 285 |
| 8.93.2 Function Documentation | 285 |
| 8.93.2.1 ImposeMax() | 285 |
| 8.93.2.2 ImposeMin() | 285 |
| 8.93.2.3 operator=() | 286 |
| 8.93.3 Variable Documentation | 286 |
| 8.93.3.1 Integer | 286 |
| 8.94 INode Interface | 287 |
| 8.94.1 Detailed Description | 289 |
| 8.94.2 Function Documentation | 289 |
| 8.94.2.1 Combine() [1/3] | 289 |
| 8.94.2.2 Combine() [2/3] | 289 |
| 8.94.2.3 Combine() [3/3] | 289 |
| 8.94.2.4 DeregisterCallback() | 290 |
| 8.94.2.5 GetAlias() | 290 |
| 8.94.2.6 GetCachingMode() | 290 |
| 8.94.2.7 GetCastAlias() | 290 |
| 8.94.2.8 GetChildren() | 290 |
| 8.94.2.9 GetDescription() | 291 |
| 8.94.2.10 GetDisplayName() | 291 |
| 8.94.2.11 GetDocuURL() | 291 |
| 8.94.2.12 GetEventID() | 291 |
| 8.94.2.13 GetNamespace() | 291 |
| 8.94.2.14 GetNodeMap() | 291 |
| 8.94.2.15 GetParents() | 291 |
| 8.94.2.16 GetPollingTime() | 292 |

| | |
|---------------------------------------|-----|
| 8.94.2.17 GetPrincipalInterfaceType() | 292 |
| 8.94.2.18 GetProperty() | 292 |
| 8.94.2.19 GetPropertyNames() | 292 |
| 8.94.2.20 GetVisibility() | 292 |
| 8.94.2.21 ImposeAccessMode() | 293 |
| 8.94.2.22 ImposeVisibility() | 293 |
| 8.94.2.23 InvalidateNode() | 293 |
| 8.94.2.24 IsAccessModeCacheable() | 293 |
| 8.94.2.25 IsAvailable() [1/3] | 293 |
| 8.94.2.26 IsAvailable() [2/3] | 293 |
| 8.94.2.27 IsAvailable() [3/3] | 294 |
| 8.94.2.28 IsCachable() | 294 |
| 8.94.2.29 IsCacheable() | 294 |
| 8.94.2.30 IsDeprecated() | 294 |
| 8.94.2.31 IsFeature() | 294 |
| 8.94.2.32 IsImplemented() [1/3] | 294 |
| 8.94.2.33 IsImplemented() [2/3] | 295 |
| 8.94.2.34 IsImplemented() [3/3] | 295 |
| 8.94.2.35 IsReadable() [1/3] | 295 |
| 8.94.2.36 IsReadable() [2/3] | 295 |
| 8.94.2.37 IsReadable() [3/3] | 295 |
| 8.94.2.38 IsStreamable() | 295 |
| 8.94.2.39 IsVisible() | 296 |
| 8.94.2.40 IsWritable() [1/3] | 296 |
| 8.94.2.41 IsWritable() [2/3] | 296 |
| 8.94.2.42 IsWritable() [3/3] | 296 |
| 8.94.2.43 operator!=(()) | 296 |
| 8.94.2.44 operator==(()) | 296 |
| 8.94.2.45 RegisterCallback() | 297 |
| 8.94.3 Variable Documentation | 297 |

| | | |
|-----------|------------------------------|-----|
| 8.94.3.1 | Inode | 297 |
| 8.94.3.2 | IReference | 297 |
| 8.95 | INodeMap Interface | 298 |
| 8.95.1 | Detailed Description | 298 |
| 8.95.2 | Function Documentation | 298 |
| 8.95.2.1 | Connect() [1/2] | 299 |
| 8.95.2.2 | Connect() [2/2] | 299 |
| 8.95.2.3 | GetDeviceName() | 299 |
| 8.95.2.4 | GetLock() | 299 |
| 8.95.2.5 | GetNode() | 299 |
| 8.95.2.6 | GetNumNodes() | 300 |
| 8.95.2.7 | InvalidateNodes() | 300 |
| 8.95.2.8 | Poll() | 300 |
| 8.95.3 | Variable Documentation | 300 |
| 8.95.3.1 | INodeMap | 300 |
| 8.96 | INodeMapDyn Interface | 301 |
| 8.96.1 | Detailed Description | 302 |
| 8.96.2 | Function Documentation | 302 |
| 8.96.2.1 | ExtractIndependentSubtree() | 302 |
| 8.96.2.2 | GetSupportedSchemaVersions() | 302 |
| 8.96.2.3 | LoadXMLFromFile() | 302 |
| 8.96.2.4 | LoadXMLFromFileInject() | 303 |
| 8.96.2.5 | LoadXMLFromString() | 303 |
| 8.96.2.6 | LoadXMLFromStringInject() | 303 |
| 8.96.2.7 | LoadXMLFromZIPData() | 303 |
| 8.96.2.8 | LoadXMLFromZIPFile() | 303 |
| 8.96.2.9 | MergeXMLFiles() | 303 |
| 8.96.2.10 | PreprocessXMLFromFile() | 304 |
| 8.96.2.11 | PreprocessXMLFromZIPFile() | 304 |
| 8.96.3 | Variable Documentation | 305 |

| | | |
|-----------|--------------------------|-----|
| 8.96.3.1 | InodeMapDyn | 305 |
| 8.97 | IntegerNode Class | 306 |
| 8.97.1 | Detailed Description | 306 |
| 8.97.2 | Typedef Documentation | 306 |
| 8.97.2.1 | CIntegerRef | 306 |
| 8.98 | IntRegNode Class | 307 |
| 8.98.1 | Detailed Description | 307 |
| 8.99 | IPort Interface | 308 |
| 8.99.1 | Detailed Description | 308 |
| 8.99.2 | Function Documentation | 308 |
| 8.99.2.1 | Write() | 308 |
| 8.99.3 | Variable Documentation | 308 |
| 8.99.3.1 | Address | 309 |
| 8.99.3.2 | IPort | 309 |
| 8.99.3.3 | Length | 309 |
| 8.100 | IPortConstruct Interface | 310 |
| 8.100.1 | Detailed Description | 310 |
| 8.100.2 | Function Documentation | 310 |
| 8.100.2.1 | GetSwapEndianness() | 310 |
| 8.100.3 | Variable Documentation | 310 |
| 8.100.3.1 | IPortConstruct | 310 |
| 8.101 | IPortRecorder Interface | 311 |
| 8.101.1 | Detailed Description | 311 |
| 8.101.2 | Function Documentation | 311 |
| 8.101.2.1 | GetCookie() | 311 |
| 8.101.2.2 | Replay() | 312 |
| 8.101.2.3 | SetCookie() | 312 |
| 8.101.2.4 | StopRecording() | 312 |
| 8.101.3 | Variable Documentation | 312 |
| 8.101.3.1 | Invalidate | 312 |

| | |
|--|-----|
| 8.101.3.2 IPortRecorder | 312 |
| 8.101.3.3 IPortReplay | 312 |
| 8.101.3.4 IPortWriteList | 312 |
| 8.102IRegister Interfaces | 313 |
| 8.102.1 Detailed Description | 313 |
| 8.102.2 Function Documentation | 313 |
| 8.102.2.1 Get() | 313 |
| 8.102.2.2 GetAddress() | 314 |
| 8.102.2.3 GetLength() | 314 |
| 8.102.3 Variable Documentation | 314 |
| 8.102.3.1 IRegister | 314 |
| 8.103ISelector Interface | 315 |
| 8.103.1 Detailed Description | 315 |
| 8.103.2 Function Documentation | 315 |
| 8.103.2.1 GetSelectedFeatures() | 315 |
| 8.103.2.2 GetSelectingFeatures() | 315 |
| 8.103.3 Variable Documentation | 315 |
| 8.103.3.1 ISelector | 315 |
| 8.104ISelectorDigit Interface | 316 |
| 8.104.1 Detailed Description | 316 |
| 8.104.2 Function Documentation | 316 |
| 8.104.2.1 GetSelectorList() | 316 |
| 8.104.2.2 Restore() | 317 |
| 8.104.2.3 SetNext() | 317 |
| 8.104.2.4 ToString() | 317 |
| 8.104.3 Variable Documentation | 317 |
| 8.104.3.1 ISelectorDigit | 318 |
| 8.105IString Class | 319 |
| 8.105.1 Detailed Description | 319 |
| 8.105.2 Function Documentation | 319 |

| | |
|--|-----|
| 8.105.2.1 GetMaxLength() | 319 |
| 8.105.3 Variable Documentation | 319 |
| 8.105.3.1 IString | 319 |
| 8.106IValue Class | 320 |
| 8.106.1 Detailed Description | 320 |
| 8.106.2 Function Documentation | 320 |
| 8.106.2.1 FromString() | 320 |
| 8.106.2.2 IsValueCacheValid() | 321 |
| 8.106.2.3 ToString() | 321 |
| 8.106.3 Variable Documentation | 321 |
| 8.106.3.1 IValue | 321 |
| 8.107Node Class | 322 |
| 8.107.1 Detailed Description | 322 |
| 8.108NodeCallback Class | 323 |
| 8.108.1 Detailed Description | 324 |
| 8.108.2 Enumeration Type Documentation | 324 |
| 8.108.2.1 ECallbackType | 324 |
| 8.108.3 Function Documentation | 324 |
| 8.108.3.1 Deregister() | 324 |
| 8.108.3.2 make_NodeCallback() [1/2] | 324 |
| 8.108.3.3 make_NodeCallback() [2/2] | 325 |
| 8.108.3.4 Register() [1/2] | 325 |
| 8.108.3.5 Register() [2/2] | 325 |
| 8.109NodeMap Class | 326 |
| 8.109.1 Detailed Description | 326 |
| 8.110NodeMapFactory Class | 327 |
| 8.110.1 Detailed Description | 327 |
| 8.110.2 Enumeration Type Documentation | 327 |
| 8.110.2.1 ECacheUsage_t | 327 |
| 8.110.2.2 EContentType_t | 328 |

| | |
|--|-----|
| 8.111 NodeMapRef Class | 329 |
| 8.111.1 Detailed Description | 329 |
| 8.112 Persistence Class | 330 |
| 8.112.1 Detailed Description | 330 |
| 8.113 Pointer Class | 331 |
| 8.113.1 Detailed Description | 332 |
| 8.113.2 Typedef Documentation | 332 |
| 8.113.2.1 CBasePtr | 332 |
| 8.113.2.2 CBooleanPtr | 333 |
| 8.113.2.3 CCategoryPtr | 333 |
| 8.113.2.4 CChunkPortPtr | 333 |
| 8.113.2.5 CCommandPtr | 333 |
| 8.113.2.6 CDeviceInfoPtr | 333 |
| 8.113.2.7 CEnumEntryPtr | 333 |
| 8.113.2.8 CEnumerationPtr | 334 |
| 8.113.2.9 CIntegerPtr | 334 |
| 8.113.2.10 CNodeMapDynPtr | 334 |
| 8.113.2.11 CNodeMapPtr | 334 |
| 8.113.2.12 CNodePtr | 334 |
| 8.113.2.13 CPortConstructPtr | 334 |
| 8.113.2.14 CPortPtr | 335 |
| 8.113.2.15 CPortRecorderPtr | 335 |
| 8.113.2.16 CPortReplayPtr | 335 |
| 8.113.2.17 CPortWriteListPtr | 335 |
| 8.113.2.18 CRegisterPtr | 335 |
| 8.113.2.19 CSelectorPtr | 335 |
| 8.113.2.20 CStringPtr | 336 |
| 8.113.2.21 CValuePtr | 336 |
| 8.113.3 Function Documentation | 336 |
| 8.113.3.1 GetInterfaceName() | 336 |

| | |
|--------------------------------|-----|
| 8.113.3.2 IsAvailable() | 336 |
| 8.113.3.3 IsImplemented() | 336 |
| 8.113.3.4 IsReadable() | 336 |
| 8.113.3.5 IsWritable() | 336 |
| 8.114PortImpl Class | 337 |
| 8.114.1 Detailed Description | 337 |
| 8.115PortNode Class | 338 |
| 8.115.1 Detailed Description | 338 |
| 8.115.2 Typedef Documentation | 338 |
| 8.115.2.1 CPortRef | 338 |
| 8.116PortRecorder Class | 339 |
| 8.116.1 Detailed Description | 339 |
| 8.116.2 Typedef Documentation | 339 |
| 8.116.2.1 CPortRecorderRef | 339 |
| 8.117PortReplay Class | 340 |
| 8.117.1 Detailed Description | 340 |
| 8.118PortWriteList Class | 341 |
| 8.118.1 Detailed Description | 341 |
| 8.119Reference Interfaces | 342 |
| 8.119.1 Detailed Description | 342 |
| 8.119.2 Function Documentation | 342 |
| 8.119.2.1 SetNumEnums() | 342 |
| 8.120RegisterNode Class | 343 |
| 8.120.1 Detailed Description | 343 |
| 8.120.2 Typedef Documentation | 343 |
| 8.120.2.1 CRegisterRef | 343 |
| 8.121RegisterPortImpl Class | 344 |
| 8.121.1 Detailed Description | 344 |
| 8.122SelectorSet Class | 345 |
| 8.122.1 Detailed Description | 345 |

| | |
|--|-----|
| 8.123SpinTestCamera Class | 346 |
| 8.123.1 Detailed Description | 346 |
| 8.124StringNode Class | 347 |
| 8.124.1 Detailed Description | 347 |
| 8.124.2 Typedef Documentation | 347 |
| 8.124.2.1 CStringRef | 347 |
| 8.125StringRegNode Class | 348 |
| 8.125.1 Detailed Description | 348 |
| 8.126StructPort Class | 349 |
| 8.126.1 Detailed Description | 349 |
| 8.127Synch Class | 350 |
| 8.127.1 Detailed Description | 350 |
| 8.128Spinnaker GenApi Enums | 351 |
| 8.128.1 Detailed Description | 351 |
| 8.129Types Enums | 352 |
| 8.129.1 Detailed Description | 354 |
| 8.129.2 Macro Definition Documentation | 354 |
| 8.129.2.1 _UndefinedRepresentation | 354 |
| 8.129.3 Typedef Documentation | 354 |
| 8.129.3.1 StringList_t | 355 |
| 8.129.4 Enumeration Type Documentation | 355 |
| 8.129.4.1 EAccessMode | 355 |
| 8.129.4.2 ECachingMode | 355 |
| 8.129.4.3 EDisplayNotation | 355 |
| 8.129.4.4 EEndianess | 357 |
| 8.129.4.5 EGenApiSchemaVersion | 357 |
| 8.129.4.6 EIncMode | 357 |
| 8.129.4.7 EInputDirection | 358 |
| 8.129.4.8 EInterfaceType | 358 |
| 8.129.4.9 ELinkType | 358 |

| | | |
|------------|--|------------|
| 8.129.4.10 | ENamespace | 359 |
| 8.129.4.11 | ERepresentation | 359 |
| 8.129.4.12 | ESign | 359 |
| 8.129.4.13 | ESlope | 360 |
| 8.129.4.14 | EStandardNamespace | 360 |
| 8.129.4.15 | EVisibility | 360 |
| 8.129.4.16 | EXMLValidation | 361 |
| 8.129.4.17 | EYesNo | 361 |
| 8.130 | ValueNode Class | 362 |
| 8.130.1 | Detailed Description | 362 |
| 8.130.2 | Typedef Documentation | 362 |
| 8.130.2.1 | CValueRef | 362 |
| 8.131 | ChunkAdapterU3V Class | 363 |
| 8.131.1 | Detailed Description | 363 |
| 9 | Namespace Documentation | 365 |
| 9.1 | Spinnaker Namespace Reference | 365 |
| 9.2 | Spinnaker::GenApi Namespace Reference | 404 |
| 9.2.1 | Typedef Documentation | 418 |
| 9.2.1.1 | IDevFileStream | 418 |
| 9.2.1.2 | ODevFileStream | 419 |
| 9.2.2 | Enumeration Type Documentation | 419 |
| 9.2.2.1 | GVCP_MESSAGE_TAGS | 419 |
| 9.2.3 | Function Documentation | 419 |
| 9.2.3.1 | PersistFeature() | 419 |
| 9.2.3.2 | SET_GUID() | 419 |
| 9.2.4 | Variable Documentation | 420 |
| 9.2.4.1 | COMMAND_MAGIC | 420 |
| 9.2.4.2 | GENCP_COMMAND_HEADER_SIZE | 420 |
| 9.2.4.3 | GENCP_EVENT_BASIC_SIZE | 420 |
| 9.2.4.4 | GENCP_EVENT_CMD_ID | 420 |
| 9.2.4.5 | IPersistScript | 420 |
| 9.2.4.6 | U3V_EVENT_PREFIX | 420 |
| 9.3 | Spinnaker::GenICam Namespace Reference | 421 |
| 9.3.1 | Function Documentation | 422 |
| 9.3.1.1 | getline() [1/2] | 422 |
| 9.3.1.2 | getline() [2/2] | 422 |
| 9.3.1.3 | ThrowBadAlloc() | 422 |
| 9.4 | Spinnaker::Video Namespace Reference | 423 |

| | |
|---|------------|
| 10 Class Documentation | 425 |
| 10.1 ActionCommandResult Struct Reference | 425 |
| 10.1.1 Detailed Description | 425 |
| 10.1.2 Member Data Documentation | 425 |
| 10.1.2.1 DeviceAddress | 425 |
| 10.1.2.2 Status | 425 |
| 10.2 ArrivalEvent Class Reference | 426 |
| 10.2.1 Detailed Description | 427 |
| 10.2.2 Constructor & Destructor Documentation | 427 |
| 10.2.2.1 ArrivalEvent() | 427 |
| 10.2.2.2 ~ArrivalEvent() | 427 |
| 10.2.3 Member Function Documentation | 427 |
| 10.2.3.1 OnDeviceArrival() | 427 |
| 10.2.3.2 operator=() | 428 |
| 10.3 AttachStatistics_t Struct Reference | 428 |
| 10.3.1 Detailed Description | 428 |
| 10.3.2 Member Data Documentation | 428 |
| 10.3.2.1 NumAttachedChunks | 428 |
| 10.3.2.2 NumChunkPorts | 428 |
| 10.3.2.3 NumChunks | 429 |
| 10.4 AutoLock Class Reference | 429 |
| 10.4.1 Constructor & Destructor Documentation | 429 |
| 10.4.1.1 AutoLock() | 429 |
| 10.4.1.2 ~AutoLock() | 429 |
| 10.5 AutoLock Class Reference | 429 |
| 10.5.1 Constructor & Destructor Documentation | 430 |
| 10.5.1.1 AutoLock() | 430 |
| 10.5.1.2 ~AutoLock() | 430 |
| 10.6 AVIOption Struct Reference | 430 |
| 10.6.1 Detailed Description | 430 |

| | | |
|-----------|--|-----|
| 10.6.2 | Constructor & Destructor Documentation | 430 |
| 10.6.2.1 | AVIOption() | 431 |
| 10.6.3 | Member Data Documentation | 431 |
| 10.6.3.1 | frameRate | 431 |
| 10.6.3.2 | reserved | 431 |
| 10.7 | BasePtr< T, B > Class Template Reference | 431 |
| 10.7.1 | Detailed Description | 432 |
| 10.7.2 | Constructor & Destructor Documentation | 432 |
| 10.7.2.1 | BasePtr() [1/2] | 432 |
| 10.7.2.2 | ~BasePtr() | 432 |
| 10.7.2.3 | BasePtr() [2/2] | 433 |
| 10.7.3 | Member Function Documentation | 433 |
| 10.7.3.1 | get() | 433 |
| 10.7.3.2 | IsValid() | 433 |
| 10.7.3.3 | operator bool() | 433 |
| 10.7.3.4 | operator T*() | 433 |
| 10.7.3.5 | operator->() | 434 |
| 10.7.3.6 | operator=() [1/4] | 434 |
| 10.7.3.7 | operator=() [2/4] | 434 |
| 10.7.3.8 | operator=() [3/4] | 434 |
| 10.7.3.9 | operator=() [4/4] | 434 |
| 10.7.3.10 | operator==() [1/4] | 434 |
| 10.7.3.11 | operator==() [2/4] | 435 |
| 10.7.3.12 | operator==() [3/4] | 435 |
| 10.7.3.13 | operator==() [4/4] | 435 |
| 10.7.4 | Member Data Documentation | 435 |
| 10.7.4.1 | m_pT | 435 |
| 10.8 | BMPOption Struct Reference | 435 |
| 10.8.1 | Detailed Description | 436 |
| 10.8.2 | Constructor & Destructor Documentation | 436 |

| | | |
|-----------|--|-----|
| 10.8.2.1 | BMPOption() | 436 |
| 10.8.3 | Member Data Documentation | 436 |
| 10.8.3.1 | indexedColor_8bit | 436 |
| 10.8.3.2 | reserved | 436 |
| 10.9 | BooleanNode Class Reference | 437 |
| 10.9.1 | Detailed Description | 438 |
| 10.9.2 | Constructor & Destructor Documentation | 438 |
| 10.9.2.1 | BooleanNode() [1/2] | 438 |
| 10.9.2.2 | BooleanNode() [2/2] | 438 |
| 10.9.2.3 | ~BooleanNode() | 438 |
| 10.9.3 | Member Function Documentation | 438 |
| 10.9.3.1 | GetValue() | 438 |
| 10.9.3.2 | operator=() | 439 |
| 10.9.3.3 | SetReference() | 439 |
| 10.9.3.4 | SetValue() | 439 |
| 10.10 | Camera Class Reference | 440 |
| 10.10.1 | Detailed Description | 470 |
| 10.10.2 | Constructor & Destructor Documentation | 470 |
| 10.10.2.1 | ~Camera() | 470 |
| 10.10.2.2 | Camera() | 470 |
| 10.10.3 | Member Function Documentation | 470 |
| 10.10.3.1 | Init() | 470 |
| 10.10.4 | Member Data Documentation | 470 |
| 10.10.4.1 | AasRoiEnable | 470 |
| 10.10.4.2 | AasRoiHeight | 471 |
| 10.10.4.3 | AasRoiOffsetX | 471 |
| 10.10.4.4 | AasRoiOffsetY | 471 |
| 10.10.4.5 | AasRoiWidth | 471 |
| 10.10.4.6 | AcquisitionAbort | 472 |
| 10.10.4.7 | AcquisitionArm | 472 |

| | |
|---|-----|
| 10.10.4.8 AcquisitionBurstFrameCount | 472 |
| 10.10.4.9 AcquisitionFrameCount | 472 |
| 10.10.4.10 AcquisitionFrameRate | 472 |
| 10.10.4.11 AcquisitionFrameRateEnable | 473 |
| 10.10.4.12 AcquisitionLineRate | 473 |
| 10.10.4.13 AcquisitionMode | 473 |
| 10.10.4.14 AcquisitionResultingFrameRate | 473 |
| 10.10.4.15 AcquisitionStart | 473 |
| 10.10.4.16 AcquisitionStatus | 473 |
| 10.10.4.17 AcquisitionStatusSelector | 474 |
| 10.10.4.18 AcquisitionStop | 474 |
| 10.10.4.19 ActionDeviceKey | 474 |
| 10.10.4.20 ActionGroupKey | 474 |
| 10.10.4.21 ActionGroupMask | 474 |
| 10.10.4.22 ActionQueueSize | 474 |
| 10.10.4.23 ActionSelector | 475 |
| 10.10.4.24 ActionUnconditionalMode | 475 |
| 10.10.4.25 AdaptiveCompressionEnable | 475 |
| 10.10.4.26 AdcBitDepth | 475 |
| 10.10.4.27a PAUSEMACCtrlFramesReceived | 475 |
| 10.10.4.28a PAUSEMACCtrlFramesTransmitted | 476 |
| 10.10.4.29 AutoAlgorithmSelector | 476 |
| 10.10.4.30 AutoExposureControlLoopDamping | 476 |
| 10.10.4.31 AutoExposureControlPriority | 476 |
| 10.10.4.32 AutoExposureEVCompensation | 477 |
| 10.10.4.33 AutoExposureExposureTimeLowerLimit | 477 |
| 10.10.4.34 AutoExposureExposureTimeUpperLimit | 477 |
| 10.10.4.35 AutoExposureGainLowerLimit | 477 |
| 10.10.4.36 AutoExposureGainUpperLimit | 478 |
| 10.10.4.37 AutoExposureGreyValueLowerLimit | 478 |

| | |
|---|-----|
| 10.10.4.38AutoExposureGreyValueUpperLimit | 478 |
| 10.10.4.39AutoExposureLightingMode | 478 |
| 10.10.4.40AutoExposureMeteringMode | 479 |
| 10.10.4.41AutoExposureTargetGreyValue | 479 |
| 10.10.4.42AutoExposureTargetGreyValueAuto | 479 |
| 10.10.4.43BalanceRatio | 480 |
| 10.10.4.44BalanceRatioSelector | 480 |
| 10.10.4.45BalanceWhiteAuto | 480 |
| 10.10.4.46BalanceWhiteAutoDamping | 480 |
| 10.10.4.47BalanceWhiteAutoLowerLimit | 481 |
| 10.10.4.48BalanceWhiteAutoProfile | 481 |
| 10.10.4.49BalanceWhiteAutoUpperLimit | 481 |
| 10.10.4.50BinningHorizontal | 481 |
| 10.10.4.51BinningHorizontalMode | 482 |
| 10.10.4.52BinningSelector | 482 |
| 10.10.4.53BinningVertical | 482 |
| 10.10.4.54BinningVerticalMode | 482 |
| 10.10.4.55BlackLevel | 482 |
| 10.10.4.56BlackLevelAuto | 483 |
| 10.10.4.57BlackLevelAutoBalance | 483 |
| 10.10.4.58BlackLevelClampingEnable | 483 |
| 10.10.4.59BlackLevelRaw | 483 |
| 10.10.4.60BlackLevelSelector | 483 |
| 10.10.4.61ChunkBlackLevel | 484 |
| 10.10.4.62ChunkBlackLevelSelector | 484 |
| 10.10.4.63ChunkCounterSelector | 484 |
| 10.10.4.64ChunkCounterValue | 484 |
| 10.10.4.65ChunkCRC | 484 |
| 10.10.4.66ChunkEnable | 484 |
| 10.10.4.67ChunkEncoderSelector | 485 |

| | | |
|------------|--|-----|
| 10.10.4.68 | ChunkEncoderStatus | 485 |
| 10.10.4.69 | ChunkEncoderValue | 485 |
| 10.10.4.70 | ChunkExposureEndLineStatusAll | 485 |
| 10.10.4.71 | ChunkExposureTime | 485 |
| 10.10.4.72 | ChunkExposureTimeSelector | 485 |
| 10.10.4.73 | ChunkFrameID | 486 |
| 10.10.4.74 | ChunkGain | 486 |
| 10.10.4.75 | ChunkGainSelector | 486 |
| 10.10.4.76 | ChunkHeight | 486 |
| 10.10.4.77 | ChunkImage | 486 |
| 10.10.4.78 | ChunkImageComponent | 486 |
| 10.10.4.79 | ChunkInferenceConfidence | 487 |
| 10.10.4.80 | ChunkInferenceResult | 487 |
| 10.10.4.81 | ChunkLinePitch | 487 |
| 10.10.4.82 | ChunkLineStatusAll | 487 |
| 10.10.4.83 | ChunkModeActive | 487 |
| 10.10.4.84 | ChunkOffsetX | 487 |
| 10.10.4.85 | ChunkOffsetY | 488 |
| 10.10.4.86 | ChunkPartSelector | 488 |
| 10.10.4.87 | ChunkPixelDynamicRangeMax | 488 |
| 10.10.4.88 | ChunkPixelDynamicRangeMin | 488 |
| 10.10.4.89 | ChunkPixelFormat | 488 |
| 10.10.4.90 | ChunkRegionID | 488 |
| 10.10.4.91 | ChunkScan3dAxisMax | 489 |
| 10.10.4.92 | ChunkScan3dAxisMin | 489 |
| 10.10.4.93 | ChunkScan3dCoordinateOffset | 489 |
| 10.10.4.94 | ChunkScan3dCoordinateReferenceSelector | 489 |
| 10.10.4.95 | ChunkScan3dCoordinateReferenceValue | 489 |
| 10.10.4.96 | ChunkScan3dCoordinateScale | 489 |
| 10.10.4.97 | ChunkScan3dCoordinateSelector | 490 |

| | | |
|-------------|--|-----|
| 10.10.4.98 | ChunkScan3dCoordinateSystem | 490 |
| 10.10.4.99 | ChunkScan3dCoordinateSystemReference | 490 |
| 10.10.4.100 | ChunkScan3dCoordinateTransformSelector | 490 |
| 10.10.4.101 | ChunkScan3dDistanceUnit | 490 |
| 10.10.4.102 | ChunkScan3dInvalidDataFlag | 490 |
| 10.10.4.103 | ChunkScan3dInvalidDataValue | 491 |
| 10.10.4.104 | ChunkScan3dOutputMode | 491 |
| 10.10.4.105 | ChunkScan3dTransformValue | 491 |
| 10.10.4.106 | ChunkScanLineSelector | 491 |
| 10.10.4.107 | ChunkSelector | 491 |
| 10.10.4.108 | ChunkSequencerSetActive | 491 |
| 10.10.4.109 | ChunkSerialData | 492 |
| 10.10.4.110 | ChunkSerialDataLength | 492 |
| 10.10.4.111 | ChunkSerialReceiveOverflow | 492 |
| 10.10.4.112 | ChunkSourceID | 492 |
| 10.10.4.113 | ChunkStreamChannelID | 492 |
| 10.10.4.114 | ChunkTimerSelector | 492 |
| 10.10.4.115 | ChunkTimerValue | 493 |
| 10.10.4.116 | ChunkTimestamp | 493 |
| 10.10.4.117 | ChunkTimestampLatchValue | 493 |
| 10.10.4.118 | ChunkTransferBlockID | 493 |
| 10.10.4.119 | ChunkTransferQueueCurrentBlockCount | 493 |
| 10.10.4.120 | ChunkTransferStreamID | 493 |
| 10.10.4.121 | ChunkWidth | 494 |
| 10.10.4.122 | Configuration | 494 |
| 10.10.4.123 | TimeSlotsCount | 494 |
| 10.10.4.124 | ColorTransformationEnable | 494 |
| 10.10.4.125 | ColorTransformationSelector | 494 |
| 10.10.4.126 | ColorTransformationValue | 495 |
| 10.10.4.127 | ColorTransformationValueSelector | 495 |

| | | |
|-------------|------------------------------|-----|
| 10.10.4.128 | ompressionRatio | 495 |
| 10.10.4.129 | ounterDelay | 495 |
| 10.10.4.130 | ounterDuration | 495 |
| 10.10.4.131 | ounterEventActivation | 496 |
| 10.10.4.132 | ounterEventSource | 496 |
| 10.10.4.133 | ounterReset | 496 |
| 10.10.4.134 | ounterResetActivation | 496 |
| 10.10.4.135 | ounterResetSource | 496 |
| 10.10.4.136 | ounterSelector | 496 |
| 10.10.4.137 | ounterStatus | 497 |
| 10.10.4.138 | ounterTriggerActivation | 497 |
| 10.10.4.139 | ounterTriggerSource | 497 |
| 10.10.4.140 | ounterValue | 497 |
| 10.10.4.141 | ounterValueAtReset | 497 |
| 10.10.4.142 | xpConnectionSelector | 497 |
| 10.10.4.143 | xpConnectionTestErrorCount | 498 |
| 10.10.4.144 | xpConnectionTestMode | 498 |
| 10.10.4.145 | xpConnectionTestPacketCount | 498 |
| 10.10.4.146 | xpLinkConfiguration | 498 |
| 10.10.4.147 | xpLinkConfigurationPreferred | 498 |
| 10.10.4.148 | xpLinkConfigurationStatus | 498 |
| 10.10.4.149 | xpPoCxpAuto | 499 |
| 10.10.4.150 | xpPoCxpStatus | 499 |
| 10.10.4.151 | xpPoCxpTripReset | 499 |
| 10.10.4.152 | xpPoCxpTurnOff | 499 |
| 10.10.4.153 | ecimationHorizontal | 499 |
| 10.10.4.154 | ecimationHorizontalMode | 500 |
| 10.10.4.155 | ecimationSelector | 500 |
| 10.10.4.156 | ecimationVertical | 500 |
| 10.10.4.157 | ecimationVerticalMode | 500 |

| | | |
|-------------|-------------------------------|-----|
| 10.10.4.158 | DefectCorrectionMode | 501 |
| 10.10.4.159 | DefectCorrectStaticEnable | 501 |
| 10.10.4.160 | DefectTableApply | 501 |
| 10.10.4.161 | DefectTableCoordinateX | 501 |
| 10.10.4.162 | DefectTableCoordinateY | 501 |
| 10.10.4.163 | DefectTableFactoryRestore | 502 |
| 10.10.4.164 | DefectTableIndex | 502 |
| 10.10.4.165 | DefectTablePixelCount | 502 |
| 10.10.4.166 | DefectTableSave | 502 |
| 10.10.4.167 | Deinterlacing | 502 |
| 10.10.4.168 | DeviceCharacterSet | 503 |
| 10.10.4.169 | DeviceClockFrequency | 503 |
| 10.10.4.170 | DeviceClockSelector | 503 |
| 10.10.4.171 | DeviceConnectionSelector | 503 |
| 10.10.4.172 | DeviceConnectionSpeed | 503 |
| 10.10.4.173 | DeviceConnectionStatus | 503 |
| 10.10.4.174 | DeviceEventChannelCount | 504 |
| 10.10.4.175 | DeviceFamilyName | 504 |
| 10.10.4.176 | DeviceFeaturePersistenceEnd | 504 |
| 10.10.4.177 | DeviceFeaturePersistenceStart | 504 |
| 10.10.4.178 | DeviceFirmwareVersion | 504 |
| 10.10.4.179 | DeviceGenCPVersionMajor | 504 |
| 10.10.4.180 | DeviceGenCPVersionMinor | 505 |
| 10.10.4.181 | DeviceID | 505 |
| 10.10.4.182 | DeviceIndicatorMode | 505 |
| 10.10.4.183 | DeviceLinkBandwidthReserve | 505 |
| 10.10.4.184 | DeviceLinkCommandTimeout | 505 |
| 10.10.4.185 | DeviceLinkConnectionCount | 505 |
| 10.10.4.186 | DeviceLinkCurrentThroughput | 506 |
| 10.10.4.187 | DeviceLinkHeartbeatMode | 506 |

| | |
|---|-----|
| 10.10.4.188DeviceLinkHeartbeatTimeout | 506 |
| 10.10.4.189DeviceLinkSelector | 506 |
| 10.10.4.190DeviceLinkSpeed | 506 |
| 10.10.4.191DeviceLinkThroughputLimit | 507 |
| 10.10.4.192DeviceLinkThroughputLimitMode | 507 |
| 10.10.4.193DeviceManifestEntrySelector | 507 |
| 10.10.4.194DeviceManifestPrimaryURL | 507 |
| 10.10.4.195DeviceManifestSchemaMajorVersion | 507 |
| 10.10.4.196DeviceManifestSchemaMinorVersion | 508 |
| 10.10.4.197DeviceManifestSecondaryURL | 508 |
| 10.10.4.198DeviceManifestXMLMajorVersion | 508 |
| 10.10.4.199DeviceManifestXMLMinorVersion | 508 |
| 10.10.4.200DeviceManifestXMLSubMinorVersion | 508 |
| 10.10.4.201DeviceManufacturerInfo | 508 |
| 10.10.4.202DeviceMaxThroughput | 509 |
| 10.10.4.203DeviceModelName | 509 |
| 10.10.4.204DevicePowerSupplySelector | 509 |
| 10.10.4.205DeviceRegistersCheck | 509 |
| 10.10.4.206DeviceRegistersEndianness | 509 |
| 10.10.4.207DeviceRegistersStreamingEnd | 510 |
| 10.10.4.208DeviceRegistersStreamingStart | 510 |
| 10.10.4.209DeviceRegistersValid | 510 |
| 10.10.4.210DeviceReset | 510 |
| 10.10.4.211DeviceScanType | 510 |
| 10.10.4.212DeviceSerialNumber | 510 |
| 10.10.4.213DeviceSerialPortBaudRate | 511 |
| 10.10.4.214DeviceSerialPortSelector | 511 |
| 10.10.4.215DeviceSFNCVersionMajor | 511 |
| 10.10.4.216DeviceSFNCVersionMinor | 511 |
| 10.10.4.217DeviceSFNCVersionSubMinor | 511 |

| | | |
|------------|-------------------------------|-----|
| 10.10.4.21 | DeviceStreamChannelCount | 511 |
| 10.10.4.21 | DeviceStreamChannelEndianness | 512 |
| 10.10.4.22 | DeviceStreamChannelLink | 512 |
| 10.10.4.22 | DeviceStreamChannelPacketSize | 512 |
| 10.10.4.22 | DeviceStreamChannelSelector | 512 |
| 10.10.4.22 | DeviceStreamChannelType | 512 |
| 10.10.4.22 | DeviceTapGeometry | 512 |
| 10.10.4.22 | DeviceTemperature | 513 |
| 10.10.4.22 | DeviceTemperatureSelector | 513 |
| 10.10.4.22 | DeviceTLType | 513 |
| 10.10.4.22 | DeviceTLVersionMajor | 513 |
| 10.10.4.22 | DeviceTLVersionMinor | 513 |
| 10.10.4.23 | DeviceTLVersionSubMinor | 514 |
| 10.10.4.23 | DeviceType | 514 |
| 10.10.4.23 | DeviceUptime | 514 |
| 10.10.4.23 | DeviceUserID | 514 |
| 10.10.4.23 | DeviceVendorName | 514 |
| 10.10.4.23 | DeviceVersion | 514 |
| 10.10.4.23 | EncoderDivider | 515 |
| 10.10.4.23 | EncoderMode | 515 |
| 10.10.4.23 | EncoderOutputMode | 515 |
| 10.10.4.23 | EncoderReset | 515 |
| 10.10.4.24 | EncoderResetActivation | 515 |
| 10.10.4.24 | EncoderResetSource | 515 |
| 10.10.4.24 | EncoderSelector | 516 |
| 10.10.4.24 | EncoderSourceA | 516 |
| 10.10.4.24 | EncoderSourceB | 516 |
| 10.10.4.24 | EncoderStatus | 516 |
| 10.10.4.24 | EncoderTimeout | 516 |
| 10.10.4.24 | EncoderValue | 516 |

| | | |
|-------------|--|-----|
| 10.10.4.248 | EncoderValueAtReset | 517 |
| 10.10.4.249 | EnumerationCount | 517 |
| 10.10.4.250 | EventAcquisitionEnd | 517 |
| 10.10.4.251 | EventAcquisitionEndFrameID | 517 |
| 10.10.4.252 | EventAcquisitionEndTimestamp | 517 |
| 10.10.4.253 | EventAcquisitionError | 517 |
| 10.10.4.254 | EventAcquisitionErrorFrameID | 518 |
| 10.10.4.255 | EventAcquisitionErrorTimestamp | 518 |
| 10.10.4.256 | EventAcquisitionStart | 518 |
| 10.10.4.257 | EventAcquisitionStartFrameID | 518 |
| 10.10.4.258 | EventAcquisitionStartTimestamp | 518 |
| 10.10.4.259 | EventAcquisitionTransferEnd | 518 |
| 10.10.4.260 | EventAcquisitionTransferEndFrameID | 519 |
| 10.10.4.261 | EventAcquisitionTransferEndTimestamp | 519 |
| 10.10.4.262 | EventAcquisitionTransferStart | 519 |
| 10.10.4.263 | EventAcquisitionTransferStartFrameID | 519 |
| 10.10.4.264 | EventAcquisitionTransferStartTimestamp | 519 |
| 10.10.4.265 | EventAcquisitionTrigger | 519 |
| 10.10.4.266 | EventAcquisitionTriggerFrameID | 520 |
| 10.10.4.267 | EventAcquisitionTriggerTimestamp | 520 |
| 10.10.4.268 | EventActionLate | 520 |
| 10.10.4.269 | EventActionLateFrameID | 520 |
| 10.10.4.270 | EventActionLateTimestamp | 520 |
| 10.10.4.271 | EventCounter0End | 520 |
| 10.10.4.272 | EventCounter0EndFrameID | 521 |
| 10.10.4.273 | EventCounter0EndTimestamp | 521 |
| 10.10.4.274 | EventCounter0Start | 521 |
| 10.10.4.275 | EventCounter0StartFrameID | 521 |
| 10.10.4.276 | EventCounter0StartTimestamp | 521 |
| 10.10.4.277 | EventCounter1End | 521 |

| | | |
|------------|---------------------------------|-----|
| 10.10.4.27 | EventCounter1EndFrameID | 522 |
| 10.10.4.27 | EventCounter1EndTimestamp | 522 |
| 10.10.4.28 | EventCounter1Start | 522 |
| 10.10.4.28 | EventCounter1StartFrameID | 522 |
| 10.10.4.28 | EventCounter1StartTimestamp | 522 |
| 10.10.4.28 | EventEncoder0Restarted | 522 |
| 10.10.4.28 | EventEncoder0RestartedFrameID | 523 |
| 10.10.4.28 | EventEncoder0RestartedTimestamp | 523 |
| 10.10.4.28 | EventEncoder0Stopped | 523 |
| 10.10.4.28 | EventEncoder0StoppedFrameID | 523 |
| 10.10.4.28 | EventEncoder0StoppedTimestamp | 523 |
| 10.10.4.28 | EventEncoder1Restarted | 523 |
| 10.10.4.29 | EventEncoder1RestartedFrameID | 524 |
| 10.10.4.29 | EventEncoder1RestartedTimestamp | 524 |
| 10.10.4.29 | EventEncoder1Stopped | 524 |
| 10.10.4.29 | EventEncoder1StoppedFrameID | 524 |
| 10.10.4.29 | EventEncoder1StoppedTimestamp | 524 |
| 10.10.4.29 | EventError | 524 |
| 10.10.4.29 | EventErrorCode | 525 |
| 10.10.4.29 | EventErrorFrameID | 525 |
| 10.10.4.29 | EventErrorTimestamp | 525 |
| 10.10.4.29 | EventExposureEnd | 525 |
| 10.10.4.30 | EventExposureEndFrameID | 525 |
| 10.10.4.30 | EventExposureEndTimestamp | 525 |
| 10.10.4.30 | EventExposureStart | 526 |
| 10.10.4.30 | EventExposureStartFrameID | 526 |
| 10.10.4.30 | EventExposureStartTimestamp | 526 |
| 10.10.4.30 | EventFrameBurstEnd | 526 |
| 10.10.4.30 | EventFrameBurstEndFrameID | 526 |
| 10.10.4.30 | EventFrameBurstEndTimestamp | 526 |

| | | |
|------------|----------------------------------|-----|
| 10.10.4.30 | EventFrameBurstStart | 527 |
| 10.10.4.30 | EventFrameBurstStartFrameID | 527 |
| 10.10.4.31 | EventFrameBurstStartTimestamp | 527 |
| 10.10.4.31 | EventFrameEnd | 527 |
| 10.10.4.31 | EventFrameEndFrameID | 527 |
| 10.10.4.31 | EventFrameEndTimestamp | 527 |
| 10.10.4.31 | EventFrameStart | 528 |
| 10.10.4.31 | EventFrameStartFrameID | 528 |
| 10.10.4.31 | EventFrameStartTimestamp | 528 |
| 10.10.4.31 | EventFrameTransferEnd | 528 |
| 10.10.4.31 | EventFrameTransferEndFrameID | 528 |
| 10.10.4.31 | EventFrameTransferEndTimestamp | 528 |
| 10.10.4.32 | EventFrameTransferStart | 529 |
| 10.10.4.32 | EventFrameTransferStartFrameID | 529 |
| 10.10.4.32 | EventFrameTransferStartTimestamp | 529 |
| 10.10.4.32 | EventFrameTrigger | 529 |
| 10.10.4.32 | EventFrameTriggerFrameID | 529 |
| 10.10.4.32 | EventFrameTriggerTimestamp | 529 |
| 10.10.4.32 | EventLine0AnyEdge | 530 |
| 10.10.4.32 | EventLine0AnyEdgeFrameID | 530 |
| 10.10.4.32 | EventLine0AnyEdgeTimestamp | 530 |
| 10.10.4.32 | EventLine0FallingEdge | 530 |
| 10.10.4.33 | EventLine0FallingEdgeFrameID | 530 |
| 10.10.4.33 | EventLine0FallingEdgeTimestamp | 530 |
| 10.10.4.33 | EventLine0RisingEdge | 531 |
| 10.10.4.33 | EventLine0RisingEdgeFrameID | 531 |
| 10.10.4.33 | EventLine0RisingEdgeTimestamp | 531 |
| 10.10.4.33 | EventLine1AnyEdge | 531 |
| 10.10.4.33 | EventLine1AnyEdgeFrameID | 531 |
| 10.10.4.33 | EventLine1AnyEdgeTimestamp | 531 |

| | | |
|------------|---------------------------------------|-----|
| 10.10.4.33 | EventLine1FallingEdge | 532 |
| 10.10.4.33 | EventLine1FallingEdgeFrameID | 532 |
| 10.10.4.34 | EventLine1FallingEdgeTimestamp | 532 |
| 10.10.4.34 | EventLine1RisingEdge | 532 |
| 10.10.4.34 | EventLine1RisingEdgeFrameID | 532 |
| 10.10.4.34 | EventLine1RisingEdgeTimestamp | 532 |
| 10.10.4.34 | EventLinkSpeedChange | 533 |
| 10.10.4.34 | EventLinkSpeedChangeFrameID | 533 |
| 10.10.4.34 | EventLinkSpeedChangeTimestamp | 533 |
| 10.10.4.34 | EventLinkTrigger0 | 533 |
| 10.10.4.34 | EventLinkTrigger0FrameID | 533 |
| 10.10.4.34 | EventLinkTrigger0Timestamp | 533 |
| 10.10.4.35 | EventLinkTrigger1 | 534 |
| 10.10.4.35 | EventLinkTrigger1FrameID | 534 |
| 10.10.4.35 | EventLinkTrigger1Timestamp | 534 |
| 10.10.4.35 | EventNotification | 534 |
| 10.10.4.35 | EventSelector | 534 |
| 10.10.4.35 | EventSequencerSetChange | 534 |
| 10.10.4.35 | EventSequencerSetChangeFrameID | 535 |
| 10.10.4.35 | EventSequencerSetChangeTimestamp | 535 |
| 10.10.4.35 | EventSerialData | 535 |
| 10.10.4.35 | EventSerialDataLength | 535 |
| 10.10.4.36 | EventSerialPortReceive | 535 |
| 10.10.4.36 | EventSerialPortReceiveTimestamp | 535 |
| 10.10.4.36 | EventSerialReceiveOverflow | 536 |
| 10.10.4.36 | EventStream0TransferBlockEnd | 536 |
| 10.10.4.36 | EventStream0TransferBlockEndFrameID | 536 |
| 10.10.4.36 | EventStream0TransferBlockEndTimestamp | 536 |
| 10.10.4.36 | EventStream0TransferBlockStart | 536 |
| 10.10.4.36 | EventStream0TransferBlockStartFrameID | 536 |

| | | |
|-------------|---|-----|
| 10.10.4.368 | EventStream0TransferBlockStartTimestamp | 537 |
| 10.10.4.369 | EventStream0TransferBlockTrigger | 537 |
| 10.10.4.370 | EventStream0TransferBlockTriggerFrameID | 537 |
| 10.10.4.371 | EventStream0TransferBlockTriggerTimestamp | 537 |
| 10.10.4.372 | EventStream0TransferBurstEnd | 537 |
| 10.10.4.373 | EventStream0TransferBurstEndFrameID | 537 |
| 10.10.4.374 | EventStream0TransferBurstEndTimestamp | 538 |
| 10.10.4.375 | EventStream0TransferBurstStart | 538 |
| 10.10.4.376 | EventStream0TransferBurstStartFrameID | 538 |
| 10.10.4.377 | EventStream0TransferBurstStartTimestamp | 538 |
| 10.10.4.378 | EventStream0TransferEnd | 538 |
| 10.10.4.379 | EventStream0TransferEndFrameID | 538 |
| 10.10.4.380 | EventStream0TransferEndTimestamp | 539 |
| 10.10.4.381 | EventStream0TransferOverflow | 539 |
| 10.10.4.382 | EventStream0TransferOverflowFrameID | 539 |
| 10.10.4.383 | EventStream0TransferOverflowTimestamp | 539 |
| 10.10.4.384 | EventStream0TransferPause | 539 |
| 10.10.4.385 | EventStream0TransferPauseFrameID | 539 |
| 10.10.4.386 | EventStream0TransferPauseTimestamp | 540 |
| 10.10.4.387 | EventStream0TransferResume | 540 |
| 10.10.4.388 | EventStream0TransferResumeFrameID | 540 |
| 10.10.4.389 | EventStream0TransferResumeTimestamp | 540 |
| 10.10.4.390 | EventStream0TransferStart | 540 |
| 10.10.4.391 | EventStream0TransferStartFrameID | 540 |
| 10.10.4.392 | EventStream0TransferStartTimestamp | 541 |
| 10.10.4.393 | EventTest | 541 |
| 10.10.4.394 | EventTestTimestamp | 541 |
| 10.10.4.395 | EventTimer0End | 541 |
| 10.10.4.396 | EventTimer0EndFrameID | 541 |
| 10.10.4.397 | EventTimer0EndTimestamp | 541 |

| | | |
|------------|---------------------------|-----|
| 10.10.4.39 | EventTimer0Start | 542 |
| 10.10.4.39 | EventTimer0StartFrameID | 542 |
| 10.10.4.40 | EventTimer0StartTimestamp | 542 |
| 10.10.4.40 | EventTimer1End | 542 |
| 10.10.4.40 | EventTimer1EndFrameID | 542 |
| 10.10.4.40 | EventTimer1EndTimestamp | 542 |
| 10.10.4.40 | EventTimer1Start | 543 |
| 10.10.4.40 | EventTimer1StartFrameID | 543 |
| 10.10.4.40 | EventTimer1StartTimestamp | 543 |
| 10.10.4.40 | ExposureActiveMode | 543 |
| 10.10.4.40 | ExposureAuto | 543 |
| 10.10.4.40 | ExposureMode | 543 |
| 10.10.4.41 | ExposureTime | 544 |
| 10.10.4.41 | ExposureTimeMode | 544 |
| 10.10.4.41 | ExposureTimeSelector | 544 |
| 10.10.4.41 | FactoryReset | 544 |
| 10.10.4.41 | FileAccessBuffer | 544 |
| 10.10.4.41 | FileAccessLength | 544 |
| 10.10.4.41 | FileAccessOffset | 545 |
| 10.10.4.41 | FileOpenMode | 545 |
| 10.10.4.41 | FileOperationExecute | 545 |
| 10.10.4.41 | FileOperationResult | 545 |
| 10.10.4.42 | FileOperationSelector | 545 |
| 10.10.4.42 | FileOperationStatus | 546 |
| 10.10.4.42 | FileSelector | 546 |
| 10.10.4.42 | FileSize | 546 |
| 10.10.4.42 | Gain | 546 |
| 10.10.4.42 | GainAuto | 546 |
| 10.10.4.42 | GainAutoBalance | 547 |
| 10.10.4.42 | GainSelector | 547 |

| | | |
|-------------|---------------------------------------|-----|
| 10.10.4.428 | Gamma | 547 |
| 10.10.4.429 | GammaEnable | 547 |
| 10.10.4.430 | DevActiveLinkCount | 547 |
| 10.10.4.431 | DevCCP | 547 |
| 10.10.4.432 | DevCurrentDefaultGateway | 548 |
| 10.10.4.433 | DevCurrentIPAddress | 548 |
| 10.10.4.434 | DevCurrentIPConfigurationDHCP | 548 |
| 10.10.4.435 | DevCurrentIPConfigurationLLA | 548 |
| 10.10.4.436 | DevCurrentIPConfigurationPersistentIP | 548 |
| 10.10.4.437 | DevCurrentPhysicalLinkConfiguration | 548 |
| 10.10.4.438 | DevCurrentSubnetMask | 549 |
| 10.10.4.439 | DevDiscoveryAckDelay | 549 |
| 10.10.4.440 | DevFirstURL | 549 |
| 10.10.4.441 | DevGVCPExtendedStatusCodes | 549 |
| 10.10.4.442 | DevGVCPExtendedStatusCodesSelector | 549 |
| 10.10.4.443 | DevGVCPHeartbeatDisable | 549 |
| 10.10.4.444 | DevGVCPPendingAck | 550 |
| 10.10.4.445 | DevGVCPPendingTimeout | 550 |
| 10.10.4.446 | DevGVSPExtendedIDMode | 550 |
| 10.10.4.447 | DevHeartbeatTimeout | 550 |
| 10.10.4.448 | DevIEEE1588 | 550 |
| 10.10.4.449 | DevIEEE1588ClockAccuracy | 550 |
| 10.10.4.450 | DevIEEE1588Mode | 551 |
| 10.10.4.451 | DevIEEE1588Status | 551 |
| 10.10.4.452 | DevInterfaceSelector | 551 |
| 10.10.4.453 | DevIPConfigurationStatus | 551 |
| 10.10.4.454 | DevMACAddress | 551 |
| 10.10.4.455 | DevMCDA | 551 |
| 10.10.4.456 | DevMCPHostPort | 552 |
| 10.10.4.457 | DevMCRC | 552 |

| | | |
|-------------|-----------------------------------|-----|
| 10.10.4.458 | evMCSP | 552 |
| 10.10.4.459 | evMCTT | 552 |
| 10.10.4.460 | evNumberOfInterfaces | 552 |
| 10.10.4.461 | evPAUSEFrameReception | 552 |
| 10.10.4.462 | evPAUSEFrameTransmission | 553 |
| 10.10.4.463 | evPersistentDefaultGateway | 553 |
| 10.10.4.464 | evPersistentIPAddress | 553 |
| 10.10.4.465 | evPersistentSubnetMask | 553 |
| 10.10.4.466 | evPhysicalLinkConfiguration | 553 |
| 10.10.4.467 | evPrimaryApplicationIPAddress | 553 |
| 10.10.4.468 | evPrimaryApplicationSocket | 554 |
| 10.10.4.469 | evPrimaryApplicationSwitchoverKey | 554 |
| 10.10.4.470 | evSCCFGAllInTransmission | 554 |
| 10.10.4.471 | evSCCFGExtendedChunkData | 554 |
| 10.10.4.472 | evSCCFGPacketResendDestination | 554 |
| 10.10.4.473 | evSCCFGUnconditionalStreaming | 554 |
| 10.10.4.474 | evSCDA | 555 |
| 10.10.4.475 | evSCPD | 555 |
| 10.10.4.476 | evSCPDDirection | 555 |
| 10.10.4.477 | evSCPHostPort | 555 |
| 10.10.4.478 | evSCPIInterfaceIndex | 555 |
| 10.10.4.479 | evSCPSBigEndian | 555 |
| 10.10.4.480 | evSCPSDoNotFragment | 556 |
| 10.10.4.481 | evSCPSFireTestPacket | 556 |
| 10.10.4.482 | evSCPSPacketSize | 556 |
| 10.10.4.483 | evSCSP | 556 |
| 10.10.4.484 | evSCZoneConfigurationLock | 556 |
| 10.10.4.485 | evSCZoneCount | 556 |
| 10.10.4.486 | evSCZoneDirectionAll | 557 |
| 10.10.4.487 | evSecondURL | 557 |

| | | |
|-------------|----------------------------------|-----|
| 10.10.4.488 | DevStreamChannelSelector | 557 |
| 10.10.4.489 | DevSupportedOption | 557 |
| 10.10.4.490 | DevSupportedOptionSelector | 557 |
| 10.10.4.491 | DevTimestampTickFrequency | 557 |
| 10.10.4.492 | GuiXmlManifestAddress | 558 |
| 10.10.4.493 | Height | 558 |
| 10.10.4.494 | HeightMax | 558 |
| 10.10.4.495 | ImageComponentEnable | 558 |
| 10.10.4.496 | ImageComponentSelector | 558 |
| 10.10.4.497 | ImageCompressionBitrate | 558 |
| 10.10.4.498 | ImageCompressionJPEGFormatOption | 559 |
| 10.10.4.499 | ImageCompressionMode | 559 |
| 10.10.4.500 | ImageCompressionQuality | 559 |
| 10.10.4.501 | ImageCompressionRateOption | 559 |
| 10.10.4.502 | OpEnable | 559 |
| 10.10.4.503 | OneFilterWidth | 560 |
| 10.10.4.504 | OneFormat | 560 |
| 10.10.4.505 | OneInputFilterSelector | 560 |
| 10.10.4.506 | OneInverter | 560 |
| 10.10.4.507 | OneMode | 560 |
| 10.10.4.508 | OnePitch | 560 |
| 10.10.4.509 | OneSelector | 561 |
| 10.10.4.510 | OneSource | 561 |
| 10.10.4.511 | OneStatus | 561 |
| 10.10.4.512 | OneStatusAll | 561 |
| 10.10.4.513 | OneErrorCount | 561 |
| 10.10.4.514 | OneUptime | 561 |
| 10.10.4.515 | LogicBlockLUTInputActivation | 562 |
| 10.10.4.516 | LogicBlockLUTInputSelector | 562 |
| 10.10.4.517 | LogicBlockLUTInputSource | 562 |

| | | |
|------------|-----------------------------|-----|
| 10.10.4.51 | LogicBlockLUTOutputValue | 562 |
| 10.10.4.51 | LogicBlockLUTOutputValueAll | 562 |
| 10.10.4.52 | LogicBlockLUTRowIndex | 562 |
| 10.10.4.52 | LogicBlockLUTSelector | 563 |
| 10.10.4.52 | LogicBlockSelector | 563 |
| 10.10.4.52 | LUTEnable | 563 |
| 10.10.4.52 | LUTIndex | 563 |
| 10.10.4.52 | LUTSelector | 563 |
| 10.10.4.52 | LUTValue | 564 |
| 10.10.4.52 | LUTValueAll | 564 |
| 10.10.4.52 | MaxDeviceResetTime | 564 |
| 10.10.4.52 | OffsetX | 564 |
| 10.10.4.53 | OffsetY | 564 |
| 10.10.4.53 | PacketResendRequestCount | 565 |
| 10.10.4.53 | PayloadSize | 565 |
| 10.10.4.53 | PixelColorFilter | 565 |
| 10.10.4.53 | PixelDynamicRangeMax | 565 |
| 10.10.4.53 | PixelDynamicRangeMin | 565 |
| 10.10.4.53 | PixelFormat | 566 |
| 10.10.4.53 | PixelFormatInfoID | 566 |
| 10.10.4.53 | PixelFormatInfoSelector | 566 |
| 10.10.4.53 | PixelSize | 566 |
| 10.10.4.54 | PowerSupplyCurrent | 566 |
| 10.10.4.54 | PowerSupplyVoltage | 566 |
| 10.10.4.54 | RegionDestination | 567 |
| 10.10.4.54 | RegionMode | 567 |
| 10.10.4.54 | RegionSelector | 567 |
| 10.10.4.54 | ReverseX | 567 |
| 10.10.4.54 | ReverseY | 567 |
| 10.10.4.54 | RgbTransformLightSource | 568 |

| | | |
|-------------|-----------------------------------|-----|
| 10.10.4.548 | Saturation | 568 |
| 10.10.4.549 | SaturationEnable | 568 |
| 10.10.4.550 | Scan3dAxisMax | 568 |
| 10.10.4.551 | Scan3dAxisMin | 568 |
| 10.10.4.552 | Scan3dCoordinateOffset | 569 |
| 10.10.4.553 | Scan3dCoordinateReferenceSelector | 569 |
| 10.10.4.554 | Scan3dCoordinateReferenceValue | 569 |
| 10.10.4.555 | Scan3dCoordinateScale | 569 |
| 10.10.4.556 | Scan3dCoordinateSelector | 569 |
| 10.10.4.557 | Scan3dCoordinateSystem | 569 |
| 10.10.4.558 | Scan3dCoordinateSystemReference | 570 |
| 10.10.4.559 | Scan3dCoordinateTransformSelector | 570 |
| 10.10.4.560 | Scan3dDistanceUnit | 570 |
| 10.10.4.561 | Scan3dInvalidDataFlag | 570 |
| 10.10.4.562 | Scan3dInvalidDataValue | 570 |
| 10.10.4.563 | Scan3dOutputMode | 570 |
| 10.10.4.564 | Scan3dTransformValue | 571 |
| 10.10.4.565 | SensorDescription | 571 |
| 10.10.4.566 | SensorDigitizationTaps | 571 |
| 10.10.4.567 | SensorHeight | 571 |
| 10.10.4.568 | SensorShutterMode | 571 |
| 10.10.4.569 | SensorTaps | 571 |
| 10.10.4.570 | SensorWidth | 572 |
| 10.10.4.571 | SequencerConfigurationMode | 572 |
| 10.10.4.572 | SequencerConfigurationValid | 572 |
| 10.10.4.573 | SequencerFeatureEnable | 572 |
| 10.10.4.574 | SequencerMode | 572 |
| 10.10.4.575 | SequencerPathSelector | 573 |
| 10.10.4.576 | SequencerSetActive | 573 |
| 10.10.4.577 | SequencerSetLoad | 573 |

| | | |
|-------------|--|-----|
| 10.10.4.578 | SequencerSetNext | 573 |
| 10.10.4.579 | SequencerSetSave | 573 |
| 10.10.4.580 | SequencerSetSelector | 574 |
| 10.10.4.581 | SequencerSetStart | 574 |
| 10.10.4.582 | SequencerSetValid | 574 |
| 10.10.4.583 | SequencerTriggerActivation | 574 |
| 10.10.4.584 | SequencerTriggerSource | 574 |
| 10.10.4.585 | SerialPortBaudRate | 575 |
| 10.10.4.586 | SerialPortDataBits | 575 |
| 10.10.4.587 | SerialPortParity | 575 |
| 10.10.4.588 | SerialPortSelector | 575 |
| 10.10.4.589 | SerialPortSource | 575 |
| 10.10.4.590 | SerialPortStopBits | 575 |
| 10.10.4.591 | SerialReceiveFramingErrorCount | 576 |
| 10.10.4.592 | SerialReceiveParityErrorCount | 576 |
| 10.10.4.593 | SerialReceiveQueueClear | 576 |
| 10.10.4.594 | SerialReceiveQueueCurrentCharacterCount | 576 |
| 10.10.4.595 | SerialReceiveQueueMaxCharacterCount | 576 |
| 10.10.4.596 | SerialTransmitQueueCurrentCharacterCount | 576 |
| 10.10.4.597 | SerialTransmitQueueMaxCharacterCount | 577 |
| 10.10.4.598 | Sharpening | 577 |
| 10.10.4.599 | SharpeningAuto | 577 |
| 10.10.4.600 | SharpeningEnable | 577 |
| 10.10.4.601 | SharpeningThreshold | 578 |
| 10.10.4.602 | SoftwareSignalPulse | 578 |
| 10.10.4.603 | SoftwareSignalSelector | 578 |
| 10.10.4.604 | SourceCount | 578 |
| 10.10.4.605 | SourceSelector | 578 |
| 10.10.4.606 | Test0001 | 579 |
| 10.10.4.607 | TestEventGenerate | 579 |

| | |
|--|-----|
| 10.10.4.60TestPattern | 579 |
| 10.10.4.60TestPatternGeneratorSelector | 579 |
| 10.10.4.61TestPendingAck | 579 |
| 10.10.4.61TimerDelay | 580 |
| 10.10.4.61TimerDuration | 580 |
| 10.10.4.61TimerReset | 580 |
| 10.10.4.61TimerSelector | 580 |
| 10.10.4.61TimerStatus | 580 |
| 10.10.4.61TimerTriggerActivation | 580 |
| 10.10.4.61TimerTriggerSource | 581 |
| 10.10.4.61TimerValue | 581 |
| 10.10.4.61Timestamp | 581 |
| 10.10.4.62TimestampLatch | 581 |
| 10.10.4.62TimestampLatchValue | 581 |
| 10.10.4.62TimestampReset | 581 |
| 10.10.4.62TLParamsLocked | 582 |
| 10.10.4.62TransferAbort | 582 |
| 10.10.4.62TransferBlockCount | 582 |
| 10.10.4.62TransferBurstCount | 582 |
| 10.10.4.62TransferComponentSelector | 582 |
| 10.10.4.62TransferControlMode | 582 |
| 10.10.4.62TransferOperationMode | 583 |
| 10.10.4.63TransferPause | 583 |
| 10.10.4.63TransferQueueCurrentBlockCount | 583 |
| 10.10.4.63TransferQueueMaxBlockCount | 583 |
| 10.10.4.63TransferQueueMode | 583 |
| 10.10.4.63TransferQueueOverflowCount | 583 |
| 10.10.4.63TransferResume | 584 |
| 10.10.4.63TransferSelector | 584 |
| 10.10.4.63TransferStart | 584 |

| | |
|-------------------------------------|-----|
| 10.10.4.63TransferStatus | 584 |
| 10.10.4.63TransferStatusSelector | 584 |
| 10.10.4.64TransferStop | 584 |
| 10.10.4.64TransferStreamChannel | 585 |
| 10.10.4.64TransferTriggerActivation | 585 |
| 10.10.4.64TransferTriggerMode | 585 |
| 10.10.4.64TransferTriggerSelector | 585 |
| 10.10.4.64TransferTriggerSource | 585 |
| 10.10.4.64TriggerActivation | 585 |
| 10.10.4.64TriggerDelay | 586 |
| 10.10.4.64TriggerDivider | 586 |
| 10.10.4.64TriggerEventTest | 586 |
| 10.10.4.65TriggerMode | 586 |
| 10.10.4.65TriggerMultiplier | 586 |
| 10.10.4.65TriggerOverlap | 587 |
| 10.10.4.65TriggerSelector | 587 |
| 10.10.4.65TriggerSoftware | 587 |
| 10.10.4.65TriggerSource | 587 |
| 10.10.4.65UserOutputSelector | 587 |
| 10.10.4.65UserOutputValue | 588 |
| 10.10.4.65UserOutputValueAll | 588 |
| 10.10.4.65UserOutputValueAllMask | 588 |
| 10.10.4.66UserSetDefault | 588 |
| 10.10.4.66UserSetFeatureEnable | 588 |
| 10.10.4.66UserSetLoad | 589 |
| 10.10.4.66UserSetSave | 589 |
| 10.10.4.66UserSetSelector | 589 |
| 10.10.4.66_3Enable | 589 |
| 10.10.4.66WhiteClip | 589 |
| 10.10.4.66WhiteClipSelector | 590 |

| | |
|--|-----|
| 10.10.4.600Width | 590 |
| 10.10.4.600WidthMax | 590 |
| 10.11 CameraBase Class Reference | 591 |
| 10.11.1 Detailed Description | 593 |
| 10.11.2 Constructor & Destructor Documentation | 593 |
| 10.11.2.1 ~CameraBase() | 593 |
| 10.11.2.2 CameraBase() [1/2] | 594 |
| 10.11.2.3 CameraBase() [2/2] | 594 |
| 10.11.3 Member Function Documentation | 594 |
| 10.11.3.1 BeginAcquisition() | 594 |
| 10.11.3.2 Delnit() | 594 |
| 10.11.3.3 DiscoverMaxPacketSize() | 595 |
| 10.11.3.4 EndAcquisition() | 595 |
| 10.11.3.5 ForceIP() | 595 |
| 10.11.3.6 GetAccessMode() | 596 |
| 10.11.3.7 GetBufferOwnership() | 596 |
| 10.11.3.8 GetGuiXml() | 596 |
| 10.11.3.9 GetNextImage() | 597 |
| 10.11.3.10GetNodeMap() | 597 |
| 10.11.3.11GetNumDataStreams() | 598 |
| 10.11.3.12GetNumImagesInUse() | 598 |
| 10.11.3.13GetTLDeviceNodeMap() | 598 |
| 10.11.3.14GetTLStreamNodeMap() | 599 |
| 10.11.3.15GetUniqueID() | 599 |
| 10.11.3.16GetUserBufferCount() | 599 |
| 10.11.3.17GetUserBufferSize() | 600 |
| 10.11.3.18GetUserBufferTotalSize() | 600 |
| 10.11.3.19nit() | 601 |
| 10.11.3.20sInitialized() | 601 |
| 10.11.3.21IsStreaming() | 601 |

| | | |
|------------|--|-----|
| 10.11.3.22 | IsValid() | 602 |
| 10.11.3.23 | operator=() | 602 |
| 10.11.3.24 | ReadPort() | 602 |
| 10.11.3.25 | RegisterEvent() [1/2] | 602 |
| 10.11.3.26 | RegisterEvent() [2/2] | 603 |
| 10.11.3.27 | SetBufferOwnership() | 603 |
| 10.11.3.28 | SetUserBuffers() [1/2] | 604 |
| 10.11.3.29 | SetUserBuffers() [2/2] | 604 |
| 10.11.3.30 | UnregisterEvent() | 605 |
| 10.11.3.31 | WritePort() | 605 |
| 10.11.4 | Friends And Related Function Documentation | 605 |
| 10.11.4.1 | InterfaceImpl | 605 |
| 10.12 | CameraList Class Reference | 606 |
| 10.12.1 | Detailed Description | 607 |
| 10.12.2 | Constructor & Destructor Documentation | 607 |
| 10.12.2.1 | CameraList() [1/2] | 607 |
| 10.12.2.2 | ~CameraList() | 607 |
| 10.12.2.3 | CameraList() [2/2] | 607 |
| 10.12.3 | Member Function Documentation | 608 |
| 10.12.3.1 | Append() | 608 |
| 10.12.3.2 | Clear() | 609 |
| 10.12.3.3 | GetByIndex() | 609 |
| 10.12.3.4 | GetBySerial() | 610 |
| 10.12.3.5 | GetSize() | 610 |
| 10.12.3.6 | operator=() | 610 |
| 10.12.3.7 | operator[]() | 611 |
| 10.12.3.8 | RemoveByIndex() | 611 |
| 10.12.3.9 | RemoveBySerial() | 611 |
| 10.13 | CameraPtr Class Reference | 612 |
| 10.13.1 | Detailed Description | 613 |

| | | |
|-----------|--|-----|
| 10.14 | CategoryNode Class Reference | 613 |
| 10.14.1 | Detailed Description | 614 |
| 10.14.2 | Constructor & Destructor Documentation | 614 |
| 10.14.2.1 | CategoryNode() [1/2] | 614 |
| 10.14.2.2 | CategoryNode() [2/2] | 615 |
| 10.14.2.3 | ~CategoryNode() | 615 |
| 10.14.3 | Member Function Documentation | 615 |
| 10.14.3.1 | GetFeatures() | 615 |
| 10.14.3.2 | SetReference() | 615 |
| 10.15 | CChunkAdapter Class Reference | 615 |
| 10.15.1 | Detailed Description | 616 |
| 10.15.2 | Constructor & Destructor Documentation | 616 |
| 10.15.2.1 | ~CChunkAdapter() | 616 |
| 10.15.2.2 | CChunkAdapter() | 617 |
| 10.15.3 | Member Function Documentation | 617 |
| 10.15.3.1 | AttachBuffer() | 617 |
| 10.15.3.2 | AttachNodeMap() | 617 |
| 10.15.3.3 | CheckBufferLayout() | 617 |
| 10.15.3.4 | ClearCaches() | 618 |
| 10.15.3.5 | DetachBuffer() | 618 |
| 10.15.3.6 | DetachNodeMap() | 618 |
| 10.15.3.7 | UpdateBuffer() | 618 |
| 10.15.4 | Member Data Documentation | 618 |
| 10.15.4.1 | m_pChunkAdapter | 618 |
| 10.16 | CChunkAdapterDcam Class Reference | 619 |
| 10.16.1 | Detailed Description | 620 |
| 10.16.2 | Constructor & Destructor Documentation | 620 |
| 10.16.2.1 | CChunkAdapterDcam() | 620 |
| 10.16.2.2 | ~CChunkAdapterDcam() | 620 |
| 10.16.3 | Member Function Documentation | 620 |

| | |
|--|-----|
| 10.16.3.1 AttachBuffer() | 620 |
| 10.16.3.2 CheckBufferLayout() | 621 |
| 10.16.3.3 CheckCRC() | 621 |
| 10.16.3.4 HasCRC() | 621 |
| 10.17 CChunkAdapterGeneric Class Reference | 621 |
| 10.17.1 Constructor & Destructor Documentation | 622 |
| 10.17.1.1 CChunkAdapterGeneric() | 622 |
| 10.17.1.2 ~CChunkAdapterGeneric() | 622 |
| 10.17.2 Member Function Documentation | 623 |
| 10.17.2.1 AttachBuffer() [1/3] | 623 |
| 10.17.2.2 AttachBuffer() [2/3] | 623 |
| 10.17.2.3 AttachBuffer() [3/3] | 623 |
| 10.17.2.4 CheckBufferLayout() | 623 |
| 10.18 CChunkAdapterGEV Class Reference | 624 |
| 10.18.1 Detailed Description | 624 |
| 10.18.2 Constructor & Destructor Documentation | 625 |
| 10.18.2.1 CChunkAdapterGEV() | 625 |
| 10.18.2.2 ~CChunkAdapterGEV() | 625 |
| 10.18.3 Member Function Documentation | 625 |
| 10.18.3.1 AttachBuffer() | 625 |
| 10.18.3.2 CheckBufferLayout() | 625 |
| 10.19 CChunkAdapterU3V Class Reference | 626 |
| 10.19.1 Detailed Description | 626 |
| 10.19.2 Constructor & Destructor Documentation | 627 |
| 10.19.2.1 CChunkAdapterU3V() | 627 |
| 10.19.2.2 ~CChunkAdapterU3V() | 627 |
| 10.19.3 Member Function Documentation | 627 |
| 10.19.3.1 AttachBuffer() | 627 |
| 10.19.3.2 CheckBufferLayout() | 627 |
| 10.20 CChunkPort Class Reference | 628 |

| | |
|--|-----|
| 10.20.1 Detailed Description | 629 |
| 10.20.2 Constructor & Destructor Documentation | 629 |
| 10.20.2.1 CChunkPort() | 629 |
| 10.20.2.2 ~CChunkPort() | 630 |
| 10.20.3 Member Function Documentation | 630 |
| 10.20.3.1 AttachChunk() | 630 |
| 10.20.3.2 AttachPort() | 630 |
| 10.20.3.3 CheckChunkID() [1/2] | 630 |
| 10.20.3.4 CheckChunkID() [2/2] | 630 |
| 10.20.3.5 ClearCache() | 631 |
| 10.20.3.6 DetachChunk() | 631 |
| 10.20.3.7 DetachPort() | 631 |
| 10.20.3.8 GetAccessMode() | 631 |
| 10.20.3.9 GetChunkIDLength() | 631 |
| 10.20.3.10 GetPrincipalInterfaceType() | 631 |
| 10.20.3.11 GetSwapEndianness() | 632 |
| 10.20.3.12 InvalidateNode() | 632 |
| 10.20.3.13 Read() | 632 |
| 10.20.3.14 SetPortImpl() | 632 |
| 10.20.3.15 UpdateBuffer() | 632 |
| 10.20.3.16 Write() | 632 |
| 10.20.4 Member Data Documentation | 633 |
| 10.20.4.1 m_pChunkPort | 633 |
| 10.20.4.2 m_pPort | 633 |
| 10.20.4.3 m_pPortAdapter | 633 |
| 10.21 CEnumerationTRef< EnumT > Class Template Reference | 633 |
| 10.21.1 Detailed Description | 635 |
| 10.21.2 Constructor & Destructor Documentation | 635 |
| 10.21.2.1 CEnumerationTRef() [1/2] | 635 |
| 10.21.2.2 CEnumerationTRef() [2/2] | 635 |

| | |
|--|-----|
| 10.21.2.3 ~CEnumerationTRef() | 635 |
| 10.21.3 Member Function Documentation | 635 |
| 10.21.3.1 GetCurrentEntry() | 636 |
| 10.21.3.2 GetEntry() [1/2] | 636 |
| 10.21.3.3 GetEntry() [2/2] | 636 |
| 10.21.3.4 GetValue() | 636 |
| 10.21.3.5 operator() | 637 |
| 10.21.3.6 operator=() [1/2] | 637 |
| 10.21.3.7 operator=() [2/2] | 637 |
| 10.21.3.8 SetEnumReference() | 637 |
| 10.21.3.9 SetNumEnums() | 637 |
| 10.21.3.10 SetReference() | 638 |
| 10.21.3.11 SetValue() | 638 |
| 10.22 CEventAdapter Class Reference | 638 |
| 10.22.1 Detailed Description | 639 |
| 10.22.2 Constructor & Destructor Documentation | 639 |
| 10.22.2.1 CEventAdapter() | 639 |
| 10.22.2.2 ~CEventAdapter() | 639 |
| 10.22.3 Member Function Documentation | 639 |
| 10.22.3.1 AttachNodeMap() | 639 |
| 10.22.3.2 DeliverMessage() | 640 |
| 10.22.3.3 DetachNodeMap() | 640 |
| 10.22.4 Member Data Documentation | 640 |
| 10.22.4.1 m_pEventAdapter | 640 |
| 10.23 CEventAdapter1394 Class Reference | 640 |
| 10.23.1 Detailed Description | 641 |
| 10.23.2 Constructor & Destructor Documentation | 641 |
| 10.23.2.1 CEventAdapter1394() | 641 |
| 10.23.2.2 ~CEventAdapter1394() | 641 |
| 10.23.3 Member Function Documentation | 642 |

| | |
|--|-----|
| 10.23.3.1 DeliverEventMessage() | 642 |
| 10.23.3.2 DeliverMessage() | 642 |
| 10.24 CEventAdapterGeneric Class Reference | 642 |
| 10.24.1 Detailed Description | 643 |
| 10.24.2 Constructor & Destructor Documentation | 643 |
| 10.24.2.1 CEventAdapterGeneric() | 643 |
| 10.24.2.2 ~CEventAdapterGeneric() | 644 |
| 10.24.3 Member Function Documentation | 644 |
| 10.24.3.1 DeliverMessage() [1/3] | 644 |
| 10.24.3.2 DeliverMessage() [2/3] | 644 |
| 10.24.3.3 DeliverMessage() [3/3] | 644 |
| 10.25 CEventAdapterGEV Class Reference | 645 |
| 10.25.1 Detailed Description | 646 |
| 10.25.2 Constructor & Destructor Documentation | 646 |
| 10.25.2.1 CEventAdapterGEV() | 646 |
| 10.25.2.2 ~CEventAdapterGEV() | 646 |
| 10.25.3 Member Function Documentation | 646 |
| 10.25.3.1 DeliverEventMessage() [1/2] | 646 |
| 10.25.3.2 DeliverEventMessage() [2/2] | 646 |
| 10.25.3.3 DeliverMessage() | 647 |
| 10.26 CEventAdapterU3V Class Reference | 647 |
| 10.26.1 Detailed Description | 648 |
| 10.26.2 Constructor & Destructor Documentation | 648 |
| 10.26.2.1 CEventAdapterU3V() | 648 |
| 10.26.2.2 ~CEventAdapterU3V() | 648 |
| 10.26.3 Member Function Documentation | 648 |
| 10.26.3.1 DeliverEventMessage() | 648 |
| 10.26.3.2 DeliverMessage() | 649 |
| 10.27 CEventPort Class Reference | 649 |
| 10.27.1 Detailed Description | 650 |

| | |
|--|-----|
| 10.27.2 Constructor & Destructor Documentation | 650 |
| 10.27.2.1 CEventPort() | 651 |
| 10.27.2.2 ~CEventPort() | 651 |
| 10.27.3 Member Function Documentation | 651 |
| 10.27.3.1 AttachEvent() | 651 |
| 10.27.3.2 AttachNode() | 651 |
| 10.27.3.3 CheckEventID() [1/2] | 651 |
| 10.27.3.4 CheckEventID() [2/2] | 652 |
| 10.27.3.5 DetachEvent() | 652 |
| 10.27.3.6 DetachNode() | 652 |
| 10.27.3.7 GetAccessMode() | 652 |
| 10.27.3.8 GetEventIDLength() | 652 |
| 10.27.3.9 GetPrincipalInterfaceType() | 652 |
| 10.27.3.10 GetSwapEndianess() | 653 |
| 10.27.3.11 InvalidateNode() | 653 |
| 10.27.3.12 Read() | 653 |
| 10.27.3.13 SetPortImpl() | 653 |
| 10.27.3.14 Write() | 653 |
| 10.27.4 Member Data Documentation | 653 |
| 10.27.4.1 m_pEventPort | 654 |
| 10.27.4.2 m_pNode | 654 |
| 10.27.4.3 m_pPortAdapter | 654 |
| 10.28 CFeatureBag Class Reference | 654 |
| 10.28.1 Detailed Description | 655 |
| 10.28.2 Constructor & Destructor Documentation | 655 |
| 10.28.2.1 CFeatureBag() | 655 |
| 10.28.2.2 ~CFeatureBag() | 655 |
| 10.28.3 Member Function Documentation | 655 |
| 10.28.3.1 GetFeatureBagHandle() | 655 |
| 10.28.3.2 LoadFromBag() | 655 |

| | |
|--|-----|
| 10.28.3.3 operator==() | 656 |
| 10.28.3.4 PersistFeature() | 656 |
| 10.28.3.5 SetInfo() | 656 |
| 10.28.3.6 StoreToBag() | 656 |
| 10.29CFloatPtr Class Reference | 657 |
| 10.29.1 Detailed Description | 658 |
| 10.29.2 Constructor & Destructor Documentation | 658 |
| 10.29.2.1 CFloatPtr() [1/2] | 658 |
| 10.29.2.2 CFloatPtr() [2/2] | 658 |
| 10.29.3 Member Function Documentation | 658 |
| 10.29.3.1 GetEnumAlias() | 658 |
| 10.29.3.2 GetIntAlias() | 658 |
| 10.29.3.3 operator=() | 659 |
| 10.30CGeneric_XMLLoaderParams Class Reference | 659 |
| 10.30.1 Detailed Description | 659 |
| 10.30.2 Member Function Documentation | 659 |
| 10.30.2.1 _Initialize() | 660 |
| 10.31CGlobalLock Class Reference | 660 |
| 10.31.1 Detailed Description | 660 |
| 10.31.2 Constructor & Destructor Documentation | 660 |
| 10.31.2.1 CGlobalLock() [1/2] | 661 |
| 10.31.2.2 CGlobalLock() [2/2] | 661 |
| 10.31.2.3 ~CGlobalLock() | 661 |
| 10.31.3 Member Function Documentation | 661 |
| 10.31.3.1 IsValid() | 661 |
| 10.31.3.2 Lock() | 661 |
| 10.31.3.3 TryLock() | 662 |
| 10.31.3.4 Unlock() | 662 |
| 10.31.4 Member Data Documentation | 662 |
| 10.31.4.1 m_DebugCount | 662 |

| | |
|--|-----|
| 10.32CGlobalLockUnlocker Class Reference | 662 |
| 10.32.1 Detailed Description | 663 |
| 10.32.2 Constructor & Destructor Documentation | 663 |
| 10.32.2.1 CGlobalLockUnlocker() | 663 |
| 10.32.2.2 ~CGlobalLockUnlocker() | 663 |
| 10.32.3 Member Function Documentation | 663 |
| 10.32.3.1 UnlockEarly() | 663 |
| 10.32.4 Member Data Documentation | 664 |
| 10.32.4.1 m_enabled | 664 |
| 10.32.4.2 m_Lock | 664 |
| 10.33ChunkData Class Reference | 664 |
| 10.33.1 Detailed Description | 666 |
| 10.33.2 Constructor & Destructor Documentation | 666 |
| 10.33.2.1 ChunkData() [1/2] | 666 |
| 10.33.2.2 ChunkData() [2/2] | 667 |
| 10.33.2.3 ~ChunkData() | 667 |
| 10.33.3 Member Function Documentation | 667 |
| 10.33.3.1 GetBlackLevel() | 667 |
| 10.33.3.2 GetCounterValue() | 667 |
| 10.33.3.3 GetCRC() | 667 |
| 10.33.3.4 GetEncoderValue() | 668 |
| 10.33.3.5 GetExposureEndLineStatusAll() | 668 |
| 10.33.3.6 GetExposureTime() | 668 |
| 10.33.3.7 GetFrameID() | 668 |
| 10.33.3.8 GetGain() | 669 |
| 10.33.3.9 GetHeight() | 669 |
| 10.33.3.10GetImage() | 669 |
| 10.33.3.11GetInferenceConfidence() | 669 |
| 10.33.3.12GetInferenceResult() | 669 |
| 10.33.3.13GetLinePitch() | 670 |

| | | |
|------------|--|-----|
| 10.33.3.14 | GetLineStatusAll() | 670 |
| 10.33.3.15 | GetOffsetX() | 670 |
| 10.33.3.16 | GetOffsetY() | 670 |
| 10.33.3.17 | GetPartSelector() | 671 |
| 10.33.3.18 | GetPixelDynamicRangeMax() | 671 |
| 10.33.3.19 | GetPixelDynamicRangeMin() | 671 |
| 10.33.3.20 | GetScan3dAxisMax() | 671 |
| 10.33.3.21 | GetScan3dAxisMin() | 672 |
| 10.33.3.22 | GetScan3dCoordinateOffset() | 672 |
| 10.33.3.23 | GetScan3dCoordinateReferenceValue() | 672 |
| 10.33.3.24 | GetScan3dCoordinateScale() | 672 |
| 10.33.3.25 | GetScan3dInvalidDataValue() | 673 |
| 10.33.3.26 | GetScan3dTransformValue() | 673 |
| 10.33.3.27 | GetScanLineSelector() | 673 |
| 10.33.3.28 | GetSequencerSetActive() | 673 |
| 10.33.3.29 | GetSerialDataLength() | 674 |
| 10.33.3.30 | GetStreamChannelID() | 674 |
| 10.33.3.31 | GetTimerValue() | 674 |
| 10.33.3.32 | GetTimestamp() | 674 |
| 10.33.3.33 | GetTimestampLatchValue() | 675 |
| 10.33.3.34 | GetTransferBlockID() | 675 |
| 10.33.3.35 | GetTransferQueueCurrentBlockCount() | 675 |
| 10.33.3.36 | GetWidth() | 675 |
| 10.33.3.37 | SetChunks() | 676 |
| 10.34 | Clock Class Reference | 676 |
| 10.34.1 | Detailed Description | 677 |
| 10.34.2 | Constructor & Destructor Documentation | 677 |
| 10.34.2.1 | Clock() [1/2] | 677 |
| 10.34.2.2 | Clock() [2/2] | 677 |
| 10.34.2.3 | ~Clock() | 677 |

| | |
|--|-----|
| 10.34.3 Member Function Documentation | 677 |
| 10.34.3.1 Lock() | 677 |
| 10.34.3.2 TryLock() | 678 |
| 10.34.3.3 Unlock() | 678 |
| 10.34.4 Friends And Related Function Documentation | 678 |
| 10.34.4.1 NodeMap | 678 |
| 10.34.5 Member Data Documentation | 678 |
| 10.34.5.1 m_bOwnLock | 678 |
| 10.34.5.2 m_lock | 678 |
| 10.35 CLock Class Reference | 679 |
| 10.35.1 Detailed Description | 679 |
| 10.35.2 Constructor & Destructor Documentation | 679 |
| 10.35.2.1 CLock() | 679 |
| 10.35.2.2 ~CLock() | 680 |
| 10.35.3 Member Function Documentation | 680 |
| 10.35.3.1 Lock() | 680 |
| 10.35.3.2 TryLock() | 680 |
| 10.35.3.3 Unlock() | 680 |
| 10.36 CLockEx Class Reference | 681 |
| 10.36.1 Detailed Description | 681 |
| 10.36.2 Member Data Documentation | 682 |
| 10.36.2.1 m_lockEx | 682 |
| 10.37 CLockEx Class Reference | 682 |
| 10.37.1 Detailed Description | 683 |
| 10.38 CNodeCallback Class Reference | 683 |
| 10.38.1 Detailed Description | 684 |
| 10.38.2 Constructor & Destructor Documentation | 684 |
| 10.38.2.1 CNodeCallback() | 684 |
| 10.38.2.2 ~CNodeCallback() | 684 |
| 10.38.3 Member Function Documentation | 684 |

| | |
|---|-----|
| 10.38.3.1 Destroy() | 684 |
| 10.38.3.2 GetCallbackType() | 685 |
| 10.38.3.3 GetNode() | 685 |
| 10.38.3.4 operator()() | 685 |
| 10.38.4 Member Data Documentation | 685 |
| 10.38.4.1 m_CallbackType | 685 |
| 10.38.4.2 m_pNode | 685 |
| 10.39CNodeMapFactory Class Reference | 686 |
| 10.39.1 Detailed Description | 687 |
| 10.39.2 Constructor & Destructor Documentation | 688 |
| 10.39.2.1 CNodeMapFactory() [1/5] | 688 |
| 10.39.2.2 ~CNodeMapFactory() | 688 |
| 10.39.2.3 CNodeMapFactory() [2/5] | 688 |
| 10.39.2.4 CNodeMapFactory() [3/5] | 688 |
| 10.39.2.5 CNodeMapFactory() [4/5] | 689 |
| 10.39.2.6 CNodeMapFactory() [5/5] | 690 |
| 10.39.3 Member Function Documentation | 690 |
| 10.39.3.1 AddInjectionData() | 690 |
| 10.39.3.2 ApplyStyleSheet() | 690 |
| 10.39.3.3 ClearCache() | 691 |
| 10.39.3.4 CreateEmptyNodeMap() | 691 |
| 10.39.3.5 CreateNodeDataFromNodeMap() | 691 |
| 10.39.3.6 CreateNodeMap() [1/2] | 691 |
| 10.39.3.7 CreateNodeMap() [2/2] | 691 |
| 10.39.3.8 ExtractSubtree() | 692 |
| 10.39.3.9 GetNodeStatistics() | 692 |
| 10.39.3.10GetSupportedSchemaVersions() | 692 |
| 10.39.3.11IsCameraDescriptionFileDataReleased() | 692 |
| 10.39.3.12IsEmpty() | 692 |
| 10.39.3.13IsLoaded() | 693 |

| | | |
|------------|--|-----|
| 10.39.3.14 | IsPreprocessed() | 693 |
| 10.39.3.15 | LoadAndInject() | 693 |
| 10.39.3.16 | operator=() | 693 |
| 10.39.3.17 | Preprocess() | 693 |
| 10.39.3.18 | ReleaseCameraDescriptionFileData() | 694 |
| 10.39.3.19 | ToString() | 694 |
| 10.39.3.20 | ToXml() | 694 |
| 10.40 | CNodeMapRef Class Reference | 694 |
| 10.40.1 | Detailed Description | 695 |
| 10.40.2 | Constructor & Destructor Documentation | 695 |
| 10.40.2.1 | CNodeMapRef() [1/3] | 696 |
| 10.40.2.2 | CNodeMapRef() [2/3] | 696 |
| 10.40.2.3 | CNodeMapRef() [3/3] | 696 |
| 10.40.3 | Member Function Documentation | 696 |
| 10.40.3.1 | operator=() [1/2] | 696 |
| 10.40.3.2 | operator=() [2/2] | 696 |
| 10.41 | CNodeMapRefT< TCameraParams > Class Template Reference | 697 |
| 10.41.1 | Detailed Description | 698 |
| 10.41.2 | Member Function Documentation | 699 |
| 10.41.2.1 | _ClearXMLCache() | 699 |
| 10.41.2.2 | _Connect() [1/2] | 699 |
| 10.41.2.3 | _Connect() [2/2] | 699 |
| 10.41.2.4 | _GetDeviceName() | 699 |
| 10.41.2.5 | _GetNode() | 700 |
| 10.41.2.6 | _GetNodes() | 700 |
| 10.41.2.7 | _GetSupportedSchemaVersions() | 700 |
| 10.41.2.8 | _InvalidateNodes() | 700 |
| 10.41.2.9 | _LoadXMLFromFile() | 700 |
| 10.41.2.10 | _LoadXMLFromFileInject() | 701 |
| 10.41.2.11 | _LoadXMLFromString() | 701 |

| | |
|---|-----|
| 10.41.2.12 _LoadXMLFromStringInject() | 701 |
| 10.41.2.13 _LoadXMLFromZIPData() | 701 |
| 10.41.2.14 _LoadXMLFromZIPFile() | 701 |
| 10.41.2.15 _Poll() | 702 |
| 10.41.3 Member Data Documentation | 702 |
| 10.41.3.1 _Ptr | 702 |
| 10.42 CommandNode Class Reference | 702 |
| 10.42.1 Detailed Description | 703 |
| 10.42.2 Constructor & Destructor Documentation | 703 |
| 10.42.2.1 CommandNode() [1/2] | 704 |
| 10.42.2.2 CommandNode() [2/2] | 704 |
| 10.42.2.3 ~CommandNode() | 704 |
| 10.42.3 Member Function Documentation | 704 |
| 10.42.3.1 Execute() | 704 |
| 10.42.3.2 IsDone() | 704 |
| 10.42.3.3 operator()() | 705 |
| 10.42.3.4 SetReference() | 705 |
| 10.43 Counter Class Reference | 705 |
| 10.43.1 Detailed Description | 705 |
| 10.43.2 Constructor & Destructor Documentation | 706 |
| 10.43.2.1 Counter() | 706 |
| 10.43.3 Member Function Documentation | 706 |
| 10.43.3.1 GetValue() | 706 |
| 10.43.3.2 IsZero() | 706 |
| 10.43.3.3 operator unsigned int() | 706 |
| 10.43.3.4 operator++() [1/2] | 706 |
| 10.43.3.5 operator++() [2/2] | 706 |
| 10.43.3.6 operator--() [1/2] | 707 |
| 10.43.3.7 operator--() [2/2] | 707 |
| 10.44 CPointer< T, B > Class Template Reference | 707 |

| | |
|--|-----|
| 10.44.1 Detailed Description | 708 |
| 10.44.2 Constructor & Destructor Documentation | 708 |
| 10.44.2.1 CPointer() [1/2] | 708 |
| 10.44.2.2 CPointer() [2/2] | 709 |
| 10.44.2.3 ~CPointer() | 709 |
| 10.44.3 Member Function Documentation | 709 |
| 10.44.3.1 IsValid() | 709 |
| 10.44.3.2 operator bool() | 709 |
| 10.44.3.3 operator T*() | 709 |
| 10.44.3.4 operator"!=() [1/5] | 710 |
| 10.44.3.5 operator"!=() [2/5] | 710 |
| 10.44.3.6 operator"!=() [3/5] | 710 |
| 10.44.3.7 operator"!=() [4/5] | 710 |
| 10.44.3.8 operator"!=() [5/5] | 710 |
| 10.44.3.9 operator()() | 710 |
| 10.44.3.10operator*() | 711 |
| 10.44.3.11operator->() | 711 |
| 10.44.3.12operator=() | 711 |
| 10.44.3.13operator==() [1/3] | 711 |
| 10.44.3.14operator==() [2/3] | 711 |
| 10.44.3.15operator==() [3/3] | 711 |
| 10.44.4 Member Data Documentation | 712 |
| 10.44.4.1 m_pT | 712 |
| 10.45CPortImpl Class Reference | 712 |
| 10.45.1 Detailed Description | 713 |
| 10.45.2 Constructor & Destructor Documentation | 713 |
| 10.45.2.1 CPortImpl() | 713 |
| 10.45.2.2 ~CPortImpl() | 713 |
| 10.45.3 Member Function Documentation | 714 |
| 10.45.3.1 GetAccessMode() | 714 |

| | |
|--|-----|
| 10.45.3.2 GetSwapEndianness() | 714 |
| 10.45.3.3 InvalidateNode() | 714 |
| 10.45.3.4 Read() | 714 |
| 10.45.3.5 Replay() | 714 |
| 10.45.3.6 SetPortImpl() | 715 |
| 10.45.3.7 Write() | 715 |
| 10.45.4 Member Data Documentation | 715 |
| 10.45.4.1 m_ptrPort | 715 |
| 10.46CPortWriteList Class Reference | 716 |
| 10.46.1 Detailed Description | 717 |
| 10.46.2 Constructor & Destructor Documentation | 717 |
| 10.46.2.1 CPortWriteList() | 717 |
| 10.46.2.2 ~CPortWriteList() | 717 |
| 10.46.3 Member Function Documentation | 717 |
| 10.46.3.1 GetCookie() | 717 |
| 10.46.3.2 GetPortWriteListHandle() | 717 |
| 10.46.3.3 Replay() | 718 |
| 10.46.3.4 SetCookie() | 718 |
| 10.46.3.5 Write() | 718 |
| 10.46.4 Member Data Documentation | 718 |
| 10.46.4.1 m_pWriteList | 718 |
| 10.47CRegisterPortImpl Class Reference | 719 |
| 10.47.1 Detailed Description | 720 |
| 10.47.2 Constructor & Destructor Documentation | 720 |
| 10.47.2.1 CRegisterPortImpl() | 720 |
| 10.47.2.2 ~CRegisterPortImpl() | 720 |
| 10.47.3 Member Function Documentation | 720 |
| 10.47.3.1 GetAccessMode() | 721 |
| 10.47.3.2 Read() | 721 |
| 10.47.3.3 ReadRegister() | 721 |

| | |
|--|-----|
| 10.47.3.4 SetPortImpl() | 721 |
| 10.47.3.5 Write() | 722 |
| 10.47.3.6 WriteRegister() | 722 |
| 10.48CSelectorSet Class Reference | 722 |
| 10.48.1 Detailed Description | 723 |
| 10.48.2 Constructor & Destructor Documentation | 723 |
| 10.48.2.1 CSelectorSet() | 723 |
| 10.48.2.2 ~CSelectorSet() | 724 |
| 10.48.3 Member Function Documentation | 724 |
| 10.48.3.1 GetSelectorList() | 724 |
| 10.48.3.2 IsEmpty() | 724 |
| 10.48.3.3 Restore() | 724 |
| 10.48.3.4 SetFirst() | 724 |
| 10.48.3.5 SetNext() | 725 |
| 10.48.3.6 ToString() | 725 |
| 10.49CTestPortStruct< CDataStruct > Class Template Reference | 725 |
| 10.49.1 Detailed Description | 726 |
| 10.49.2 Constructor & Destructor Documentation | 727 |
| 10.49.2.1 CTestPortStruct() | 727 |
| 10.49.3 Member Function Documentation | 727 |
| 10.49.3.1 GetAccessMode() | 727 |
| 10.49.3.2 GetNumReads() | 727 |
| 10.49.3.3 GetNumWrites() | 727 |
| 10.49.3.4 GetPrincipalInterfaceType() | 727 |
| 10.49.3.5 MemSet() | 728 |
| 10.49.3.6 Read() | 728 |
| 10.49.3.7 ResetStatistics() | 728 |
| 10.49.3.8 Write() | 728 |
| 10.49.4 Member Data Documentation | 728 |
| 10.49.4.1 m_BaseAddress | 728 |

| | |
|--|-----|
| 10.49.4.2 m_NumReads | 729 |
| 10.49.4.3 m_NumWrites | 729 |
| 10.50DCAM_CHECKSUM Struct Reference | 729 |
| 10.50.1 Member Data Documentation | 729 |
| 10.50.1.1 CRCChecksum | 729 |
| 10.51DCAM_CHUNK_TRAILER Struct Reference | 729 |
| 10.51.1 Member Data Documentation | 730 |
| 10.51.1.1 ChunkID | 730 |
| 10.51.1.2 ChunkLength | 730 |
| 10.51.1.3 InverseChunkLength | 730 |
| 10.52DeviceEvent Class Reference | 730 |
| 10.52.1 Detailed Description | 731 |
| 10.52.2 Constructor & Destructor Documentation | 731 |
| 10.52.2.1 DeviceEvent() | 732 |
| 10.52.2.2 ~DeviceEvent() | 732 |
| 10.52.3 Member Function Documentation | 732 |
| 10.52.3.1 GetDeviceEventId() | 732 |
| 10.52.3.2 GetDeviceEventName() | 732 |
| 10.52.3.3 OnDeviceEvent() | 732 |
| 10.52.3.4 operator=() | 733 |
| 10.53double_autovector_t Class Reference | 733 |
| 10.53.1 Detailed Description | 733 |
| 10.53.2 Constructor & Destructor Documentation | 734 |
| 10.53.2.1 double_autovector_t() [1/3] | 734 |
| 10.53.2.2 double_autovector_t() [2/3] | 734 |
| 10.53.2.3 double_autovector_t() [3/3] | 734 |
| 10.53.2.4 ~double_autovector_t() | 734 |
| 10.53.3 Member Function Documentation | 734 |
| 10.53.3.1 operator delete() | 734 |
| 10.53.3.2 operator new() | 734 |

| | |
|--|-----|
| 10.53.3.3 operator=() | 735 |
| 10.53.3.4 operator[]() [1/2] | 735 |
| 10.53.3.5 operator[]() [2/2] | 735 |
| 10.53.3.6 size() | 735 |
| 10.53.4 Member Data Documentation | 735 |
| 10.53.4.1 _pCount | 735 |
| 10.53.4.2 _pv | 735 |
| 10.54EAccessModeClass Class Reference | 736 |
| 10.54.1 Detailed Description | 736 |
| 10.54.2 Member Function Documentation | 736 |
| 10.54.2.1 FromString() | 736 |
| 10.54.2.2 ToString() [1/2] | 736 |
| 10.54.2.3 ToString() [2/2] | 736 |
| 10.55ECachingModeClass Class Reference | 737 |
| 10.55.1 Detailed Description | 737 |
| 10.55.2 Member Function Documentation | 737 |
| 10.55.2.1 FromString() | 737 |
| 10.55.2.2 ToString() [1/2] | 737 |
| 10.55.2.3 ToString() [2/2] | 737 |
| 10.56EDisplayNotationClass Class Reference | 738 |
| 10.56.1 Detailed Description | 738 |
| 10.56.2 Member Function Documentation | 738 |
| 10.56.2.1 FromString() | 738 |
| 10.56.2.2 ToString() [1/2] | 738 |
| 10.56.2.3 ToString() [2/2] | 738 |
| 10.57EEndianessClass Class Reference | 739 |
| 10.57.1 Detailed Description | 739 |
| 10.57.2 Member Function Documentation | 739 |
| 10.57.2.1 FromString() | 739 |
| 10.57.2.2 ToString() [1/2] | 739 |

| | |
|--|-----|
| 10.57.2.3 ToString() [2/2] | 739 |
| 10.58EGenApiSchemaVersionClass Class Reference | 740 |
| 10.58.1 Detailed Description | 740 |
| 10.58.2 Member Function Documentation | 740 |
| 10.58.2.1 FromString() | 740 |
| 10.58.2.2 ToString() [1/2] | 740 |
| 10.58.2.3 ToString() [2/2] | 740 |
| 10.59EInputDirectionClass Class Reference | 741 |
| 10.59.1 Detailed Description | 741 |
| 10.59.2 Member Function Documentation | 741 |
| 10.59.2.1 FromString() | 741 |
| 10.59.2.2 ToString() [1/2] | 741 |
| 10.59.2.3 ToString() [2/2] | 741 |
| 10.60ENameSpaceClass Class Reference | 742 |
| 10.60.1 Detailed Description | 742 |
| 10.60.2 Member Function Documentation | 742 |
| 10.60.2.1 FromString() | 742 |
| 10.60.2.2 ToString() [1/2] | 742 |
| 10.60.2.3 ToString() [2/2] | 742 |
| 10.61EnumEntryNode Class Reference | 743 |
| 10.61.1 Detailed Description | 744 |
| 10.61.2 Constructor & Destructor Documentation | 744 |
| 10.61.2.1 EnumEntryNode() [1/2] | 744 |
| 10.61.2.2 EnumEntryNode() [2/2] | 744 |
| 10.61.2.3 ~EnumEntryNode() | 744 |
| 10.61.3 Member Function Documentation | 744 |
| 10.61.3.1 GetNumericValue() | 745 |
| 10.61.3.2 GetSymbolic() | 745 |
| 10.61.3.3 GetValue() | 745 |
| 10.61.3.4 IsSelfClearing() | 745 |

| | |
|--|-----|
| 10.61.3.5 SetReference() | 745 |
| 10.62EnumNode Class Reference | 746 |
| 10.62.1 Detailed Description | 748 |
| 10.62.2 Constructor & Destructor Documentation | 748 |
| 10.62.2.1 EnumNode() [1/2] | 748 |
| 10.62.2.2 EnumNode() [2/2] | 748 |
| 10.62.2.3 ~EnumNode() | 748 |
| 10.62.3 Member Function Documentation | 748 |
| 10.62.3.1 GetCurrentEntry() | 748 |
| 10.62.3.2 GetEntries() | 749 |
| 10.62.3.3 GetEntry() | 749 |
| 10.62.3.4 GetEntryByName() | 749 |
| 10.62.3.5 GetIntValue() | 749 |
| 10.62.3.6 GetSymbolics() | 750 |
| 10.62.3.7 operator*() | 750 |
| 10.62.3.8 operator=() | 750 |
| 10.62.3.9 SetIntValue() | 750 |
| 10.62.3.10SetReference() | 750 |
| 10.62.4 Member Data Documentation | 751 |
| 10.62.4.1 m_pEnumeration | 751 |
| 10.63ERepresentationClass Class Reference | 751 |
| 10.63.1 Detailed Description | 751 |
| 10.63.2 Member Function Documentation | 751 |
| 10.63.2.1 FromString() | 752 |
| 10.63.2.2 ToString() [1/2] | 752 |
| 10.63.2.3 ToString() [2/2] | 752 |
| 10.64ESignClass Class Reference | 752 |
| 10.64.1 Detailed Description | 752 |
| 10.64.2 Member Function Documentation | 753 |
| 10.64.2.1 FromString() | 753 |

| | |
|--|-----|
| 10.64.2.2 ToString() [1/2] | 753 |
| 10.64.2.3 ToString() [2/2] | 753 |
| 10.65ESlopeClass Class Reference | 753 |
| 10.65.1 Detailed Description | 754 |
| 10.65.2 Member Function Documentation | 754 |
| 10.65.2.1 FromString() | 754 |
| 10.65.2.2 ToString() [1/2] | 754 |
| 10.65.2.3 ToString() [2/2] | 754 |
| 10.66EStandardNameSpaceClass Class Reference | 754 |
| 10.66.1 Detailed Description | 755 |
| 10.66.2 Member Function Documentation | 755 |
| 10.66.2.1 FromString() | 755 |
| 10.66.2.2 ToString() [1/2] | 755 |
| 10.66.2.3 ToString() [2/2] | 755 |
| 10.67Event Class Reference | 756 |
| 10.67.1 Detailed Description | 757 |
| 10.67.2 Constructor & Destructor Documentation | 757 |
| 10.67.2.1 ~Event() | 757 |
| 10.67.2.2 Event() | 757 |
| 10.67.3 Member Function Documentation | 757 |
| 10.67.3.1 GetEventPayloadData() | 757 |
| 10.67.3.2 GetEventPayloadDataSize() | 757 |
| 10.67.3.3 GetEventType() | 758 |
| 10.67.3.4 operator=() | 758 |
| 10.67.3.5 SetEventPayload() | 758 |
| 10.67.3.6 SetEventType() | 758 |
| 10.67.4 Friends And Related Function Documentation | 758 |
| 10.67.4.1 EventProcessor | 758 |
| 10.67.4.2 IDataStream | 759 |
| 10.67.4.3 Stream | 759 |

| | |
|--|-----|
| 10.67.5 Member Data Documentation | 759 |
| 10.67.5.1 m_pEventData | 759 |
| 10.68EVisibilityClass Class Reference | 759 |
| 10.68.1 Detailed Description | 759 |
| 10.68.2 Member Function Documentation | 759 |
| 10.68.2.1 FromString() | 760 |
| 10.68.2.2 ToString() [1/2] | 760 |
| 10.68.2.3 ToString() [2/2] | 760 |
| 10.69Exception Class Reference | 760 |
| 10.69.1 Detailed Description | 761 |
| 10.69.2 Constructor & Destructor Documentation | 762 |
| 10.69.2.1 Exception() [1/3] | 762 |
| 10.69.2.2 Exception() [2/3] | 762 |
| 10.69.2.3 Exception() [3/3] | 762 |
| 10.69.2.4 ~Exception() | 762 |
| 10.69.3 Member Function Documentation | 763 |
| 10.69.3.1 GetBuildDate() | 763 |
| 10.69.3.2 GetBuildTime() | 763 |
| 10.69.3.3 GetError() | 763 |
| 10.69.3.4 GetErrorMessage() | 763 |
| 10.69.3.5 GetFileName() | 763 |
| 10.69.3.6 GetFullErrorMessage() | 763 |
| 10.69.3.7 GetFunctionName() | 764 |
| 10.69.3.8 GetLineNumber() | 764 |
| 10.69.3.9 operator!=(()) | 764 |
| 10.69.3.10operator=(()) | 764 |
| 10.69.3.11operator==(()) | 764 |
| 10.69.3.12what() | 764 |
| 10.70EYesNoClass Class Reference | 765 |
| 10.70.1 Detailed Description | 765 |

| | |
|--|-----|
| 10.70.2 Member Function Documentation | 765 |
| 10.70.2.1 FromString() | 765 |
| 10.70.2.2 ToString() [1/2] | 765 |
| 10.70.2.3 ToString() [2/2] | 765 |
| 10.71 FileProtocolAdapter Class Reference | 766 |
| 10.71.1 Detailed Description | 766 |
| 10.71.2 Constructor & Destructor Documentation | 766 |
| 10.71.2.1 FileProtocolAdapter() | 766 |
| 10.71.2.2 ~FileProtocolAdapter() | 766 |
| 10.71.3 Member Function Documentation | 767 |
| 10.71.3.1 attach() | 767 |
| 10.71.3.2 closeFile() | 768 |
| 10.71.3.3 deleteFile() | 768 |
| 10.71.3.4 getBufSize() | 768 |
| 10.71.3.5 openFile() | 769 |
| 10.71.3.6 read() | 769 |
| 10.71.3.7 write() | 770 |
| 10.72 FloatNode Class Reference | 770 |
| 10.72.1 Detailed Description | 772 |
| 10.72.2 Constructor & Destructor Documentation | 773 |
| 10.72.2.1 FloatNode() [1/2] | 773 |
| 10.72.2.2 FloatNode() [2/2] | 773 |
| 10.72.2.3 ~FloatNode() | 773 |
| 10.72.3 Member Function Documentation | 773 |
| 10.72.3.1 GetDisplayNotation() | 773 |
| 10.72.3.2 GetDisplayPrecision() | 773 |
| 10.72.3.3 GetEnumAlias() | 773 |
| 10.72.3.4 GetInc() | 774 |
| 10.72.3.5 GetIncMode() | 774 |
| 10.72.3.6 GetIntAlias() | 774 |

| | |
|--|-----|
| 10.72.3.7 GetListOfValidValues() | 774 |
| 10.72.3.8 GetMax() | 774 |
| 10.72.3.9 GetMin() | 774 |
| 10.72.3.10 GetRepresentation() | 775 |
| 10.72.3.11 GetUnit() | 775 |
| 10.72.3.12 GetValue() | 775 |
| 10.72.3.13 HasInc() | 775 |
| 10.72.3.14 ImposeMax() | 775 |
| 10.72.3.15 ImposeMin() | 776 |
| 10.72.3.16 operator()() | 776 |
| 10.72.3.17 operator*() | 776 |
| 10.72.3.18 operator=() | 776 |
| 10.72.3.19 SetReference() | 776 |
| 10.72.3.20 SetValue() | 776 |
| 10.73 FloatRegNode Class Reference | 777 |
| 10.73.1 Detailed Description | 778 |
| 10.73.2 Constructor & Destructor Documentation | 778 |
| 10.73.2.1 FloatRegNode() [1/2] | 779 |
| 10.73.2.2 FloatRegNode() [2/2] | 779 |
| 10.73.2.3 ~FloatRegNode() | 779 |
| 10.73.3 Member Function Documentation | 779 |
| 10.73.3.1 SetReference() | 779 |
| 10.74 Function_NodeCallback< Function > Class Template Reference | 780 |
| 10.74.1 Detailed Description | 780 |
| 10.74.2 Constructor & Destructor Documentation | 781 |
| 10.74.2.1 Function_NodeCallback() | 781 |
| 10.74.3 Member Function Documentation | 781 |
| 10.74.3.1 Destroy() | 781 |
| 10.74.3.2 operator()() | 781 |
| 10.75 gcstring Class Reference | 782 |

| | |
|--|-----|
| 10.75.1 Constructor & Destructor Documentation | 783 |
| 10.75.1.1 gcstring() [1/5] | 783 |
| 10.75.1.2 gcstring() [2/5] | 783 |
| 10.75.1.3 gcstring() [3/5] | 783 |
| 10.75.1.4 gcstring() [4/5] | 783 |
| 10.75.1.5 gcstring() [5/5] | 784 |
| 10.75.1.6 ~gcstring() | 784 |
| 10.75.2 Member Function Documentation | 784 |
| 10.75.2.1 _npos() | 784 |
| 10.75.2.2 append() [1/2] | 784 |
| 10.75.2.3 append() [2/2] | 784 |
| 10.75.2.4 assign() [1/4] | 784 |
| 10.75.2.5 assign() [2/4] | 785 |
| 10.75.2.6 assign() [3/4] | 785 |
| 10.75.2.7 assign() [4/4] | 785 |
| 10.75.2.8 c_str() | 785 |
| 10.75.2.9 compare() | 785 |
| 10.75.2.10 empty() | 785 |
| 10.75.2.11 find() [1/5] | 786 |
| 10.75.2.12 find() [2/5] | 786 |
| 10.75.2.13 find() [3/5] | 786 |
| 10.75.2.14 find() [4/5] | 786 |
| 10.75.2.15 find() [5/5] | 786 |
| 10.75.2.16 find_first_not_of() | 786 |
| 10.75.2.17 find_first_of() | 787 |
| 10.75.2.18 length() | 787 |
| 10.75.2.19 max_size() | 787 |
| 10.75.2.20 operator const char *() | 787 |
| 10.75.2.21 operator delete() [1/2] | 787 |
| 10.75.2.22 operator delete() [2/2] | 787 |

| | | |
|------------|--|-----|
| 10.75.2.23 | operator new() [1/2] | 787 |
| 10.75.2.24 | operator new() [2/2] | 788 |
| 10.75.2.25 | operator"!=() [1/2] | 788 |
| 10.75.2.26 | operator"!=() [2/2] | 788 |
| 10.75.2.27 | operator+=() [1/5] | 788 |
| 10.75.2.28 | operator+=() [2/5] | 788 |
| 10.75.2.29 | operator+=() [3/5] | 788 |
| 10.75.2.30 | operator+=() [4/5] | 788 |
| 10.75.2.31 | operator+=() [5/5] | 789 |
| 10.75.2.32 | operator<() | 789 |
| 10.75.2.33 | operator=() | 789 |
| 10.75.2.34 | operator==() [1/2] | 789 |
| 10.75.2.35 | operator==() [2/2] | 789 |
| 10.75.2.36 | operator>() | 789 |
| 10.75.2.37 | resize() | 789 |
| 10.75.2.38 | size() | 790 |
| 10.75.2.39 | substr() | 790 |
| 10.75.2.40 | swap() | 790 |
| 10.75.3 | Friends And Related Function Documentation | 790 |
| 10.75.3.1 | operator+ [1/3] | 790 |
| 10.75.3.2 | operator+ [2/3] | 790 |
| 10.75.3.3 | operator+ [3/3] | 790 |
| 10.75.4 | Member Data Documentation | 791 |
| 10.75.4.1 | npos | 791 |
| 10.76 | GVCP_CHUNK_TRAILER Struct Reference | 791 |
| 10.76.1 | Detailed Description | 791 |
| 10.76.2 | Member Data Documentation | 791 |
| 10.76.2.1 | ChunkID | 791 |
| 10.76.2.2 | ChunkLength | 791 |
| 10.77 | GVCP_EVENT_ITEM Struct Reference | 792 |

| | |
|--|-----|
| 10.77.1 Detailed Description | 792 |
| 10.77.2 Member Data Documentation | 792 |
| 10.77.2.1 BlockId | 792 |
| 10.77.2.2 EventId | 792 |
| 10.77.2.3 ReservedOrEventSize | 792 |
| 10.77.2.4 StreamChannelId | 792 |
| 10.77.2.5 TimestampHigh | 793 |
| 10.77.2.6 TimestampLow | 793 |
| 10.78GVCP_EVENT_ITEM_BASIC Struct Reference | 793 |
| 10.78.1 Detailed Description | 793 |
| 10.78.2 Member Data Documentation | 793 |
| 10.78.2.1 EventId | 793 |
| 10.78.2.2 ReservedOrEventSize | 793 |
| 10.79GVCP_EVENT_ITEM_EXTENDED_ID Struct Reference | 794 |
| 10.79.1 Detailed Description | 794 |
| 10.79.2 Member Data Documentation | 794 |
| 10.79.2.1 BlockId | 794 |
| 10.79.2.2 BlockId64High | 794 |
| 10.79.2.3 BlockId64Low | 794 |
| 10.79.2.4 EventId | 794 |
| 10.79.2.5 ReservedOrEventSize | 795 |
| 10.79.2.6 StreamChannelId | 795 |
| 10.79.2.7 TimestampHigh | 795 |
| 10.79.2.8 TimestampLow | 795 |
| 10.80GVCP_EVENT_REQUEST Struct Reference | 795 |
| 10.80.1 Detailed Description | 796 |
| 10.80.2 Member Data Documentation | 796 |
| 10.80.2.1 Header | 796 |
| 10.80.2.2 Items | 796 |
| 10.81GVCP_EVENT_REQUEST_EXTENDED_ID Struct Reference | 796 |

| | |
|--|-----|
| 10.81.1 Detailed Description | 797 |
| 10.81.2 Member Data Documentation | 797 |
| 10.81.2.1 Header | 797 |
| 10.81.2.2 Items | 797 |
| 10.82GVCP_EVENTDATA_REQUEST Struct Reference | 797 |
| 10.82.1 Detailed Description | 798 |
| 10.82.2 Member Data Documentation | 798 |
| 10.82.2.1 Data | 798 |
| 10.82.2.2 Event | 798 |
| 10.82.2.3 Header | 798 |
| 10.83GVCP_EVENTDATA_REQUEST_EXTENDED_ID Struct Reference | 798 |
| 10.83.1 Detailed Description | 799 |
| 10.83.2 Member Data Documentation | 799 |
| 10.83.2.1 Data | 799 |
| 10.83.2.2 Event | 799 |
| 10.83.2.3 Header | 799 |
| 10.84GVCP_REQUEST_HEADER Struct Reference | 799 |
| 10.84.1 Detailed Description | 800 |
| 10.84.2 Member Data Documentation | 800 |
| 10.84.2.1 Command | 800 |
| 10.84.2.2 Flags | 800 |
| 10.84.2.3 Length | 800 |
| 10.84.2.4 Magic | 800 |
| 10.84.2.5 ReqId | 800 |
| 10.85H264Option Struct Reference | 800 |
| 10.85.1 Detailed Description | 801 |
| 10.85.2 Constructor & Destructor Documentation | 801 |
| 10.85.2.1 H264Option() | 801 |
| 10.85.3 Member Data Documentation | 801 |
| 10.85.3.1 bitrate | 801 |

| | |
|--|-----|
| 10.85.3.2 frameRate | 802 |
| 10.85.3.3 height | 802 |
| 10.85.3.4 reserved | 802 |
| 10.85.3.5 width | 802 |
| 10.86IArrivalEvent Class Reference | 803 |
| 10.86.1 Constructor & Destructor Documentation | 804 |
| 10.86.1.1 ~IArrivalEvent() | 804 |
| 10.86.1.2 IArrivalEvent() [1/2] | 804 |
| 10.86.1.3 IArrivalEvent() [2/2] | 804 |
| 10.86.2 Member Function Documentation | 804 |
| 10.86.2.1 OnDeviceArrival() | 804 |
| 10.86.2.2 operator=() | 804 |
| 10.87ICameraBase Class Reference | 805 |
| 10.87.1 Detailed Description | 806 |
| 10.87.2 Constructor & Destructor Documentation | 807 |
| 10.87.2.1 ~ICameraBase() | 807 |
| 10.87.2.2 ICameraBase() [1/2] | 807 |
| 10.87.2.3 ICameraBase() [2/2] | 807 |
| 10.87.3 Member Function Documentation | 807 |
| 10.87.3.1 BeginAcquisition() | 807 |
| 10.87.3.2 Delnit() | 807 |
| 10.87.3.3 DiscoverMaxPacketSize() | 808 |
| 10.87.3.4 EndAcquisition() | 808 |
| 10.87.3.5 ForceIP() | 808 |
| 10.87.3.6 GetAccessMode() | 808 |
| 10.87.3.7 GetBufferOwnership() | 808 |
| 10.87.3.8 GetGuiXml() | 808 |
| 10.87.3.9 GetNextImage() | 809 |
| 10.87.3.10GetNodeMap() | 809 |
| 10.87.3.11GetNumDataStreams() | 809 |

| | | |
|------------|--|-----|
| 10.87.3.12 | GetNumImagesInUse() | 809 |
| 10.87.3.13 | GetTLDeviceNodeMap() | 809 |
| 10.87.3.14 | GetTLStreamNodeMap() | 809 |
| 10.87.3.15 | GetUniqueID() | 810 |
| 10.87.3.16 | GetUserBufferCount() | 810 |
| 10.87.3.17 | GetUserBufferSize() | 810 |
| 10.87.3.18 | GetUserBufferTotalSize() | 810 |
| 10.87.3.19 | init() | 810 |
| 10.87.3.20 | IsInitialized() | 810 |
| 10.87.3.21 | IsStreaming() | 811 |
| 10.87.3.22 | IsValid() | 811 |
| 10.87.3.23 | operator=() | 811 |
| 10.87.3.24 | ReadPort() | 811 |
| 10.87.3.25 | RegisterEvent() [1/2] | 811 |
| 10.87.3.26 | RegisterEvent() [2/2] | 811 |
| 10.87.3.27 | SetBufferOwnership() | 812 |
| 10.87.3.28 | SetUserBuffers() [1/2] | 812 |
| 10.87.3.29 | SetUserBuffers() [2/2] | 812 |
| 10.87.3.30 | UnregisterEvent() | 812 |
| 10.87.3.31 | WritePort() | 812 |
| 10.87.4 | Friends And Related Function Documentation | 813 |
| 10.87.4.1 | CameraInternal | 813 |
| 10.87.4.2 | InterfaceImpl | 813 |
| 10.87.5 | Member Data Documentation | 813 |
| 10.87.5.1 | m_pCameraBaseData | 813 |
| 10.87.5.2 | TLDevice | 813 |
| 10.87.5.3 | TLStream | 813 |
| 10.88 | ICameraList Class Reference | 814 |
| 10.88.1 | Detailed Description | 814 |
| 10.88.2 | Constructor & Destructor Documentation | 815 |

| | |
|--|-----|
| 10.88.2.1 ~ICameraList() | 815 |
| 10.88.2.2 ICameraList() [1/2] | 815 |
| 10.88.2.3 ICameraList() [2/2] | 815 |
| 10.88.3 Member Function Documentation | 815 |
| 10.88.3.1 Append() | 815 |
| 10.88.3.2 Clear() | 815 |
| 10.88.3.3 GetByIndex() | 816 |
| 10.88.3.4 GetBySerial() | 816 |
| 10.88.3.5 GetSize() | 816 |
| 10.88.3.6 operator=() | 816 |
| 10.88.3.7 operator[]() | 816 |
| 10.88.3.8 RemoveByIndex() | 816 |
| 10.88.3.9 RemoveBySerial() | 817 |
| 10.88.4 Friends And Related Function Documentation | 817 |
| 10.88.4.1 CameraListImpl | 817 |
| 10.88.4.2 InterfacelImpl | 817 |
| 10.88.5 Member Data Documentation | 817 |
| 10.88.5.1 m_pCameraListData | 817 |
| 10.89 IChunkData Class Reference | 817 |
| 10.89.1 Detailed Description | 818 |
| 10.89.2 Constructor & Destructor Documentation | 818 |
| 10.89.2.1 ~IChunkData() | 819 |
| 10.89.2.2 IChunkData() | 819 |
| 10.89.3 Member Function Documentation | 819 |
| 10.89.3.1 GetBlackLevel() | 819 |
| 10.89.3.2 GetCounterValue() | 819 |
| 10.89.3.3 GetCRC() | 819 |
| 10.89.3.4 GetEncoderValue() | 819 |
| 10.89.3.5 GetExposureEndLineStatusAll() | 820 |
| 10.89.3.6 GetExposureTime() | 820 |

| | |
|--|-----|
| 10.89.3.7 GetFrameID() | 820 |
| 10.89.3.8 GetGain() | 820 |
| 10.89.3.9 GetHeight() | 820 |
| 10.89.3.10 GetImage() | 820 |
| 10.89.3.11 GetInferenceConfidence() | 821 |
| 10.89.3.12 GetInferenceResult() | 821 |
| 10.89.3.13 GetLinePitch() | 821 |
| 10.89.3.14 GetLineStatusAll() | 821 |
| 10.89.3.15 GetOffsetX() | 821 |
| 10.89.3.16 GetOffsetY() | 821 |
| 10.89.3.17 GetPartSelector() | 822 |
| 10.89.3.18 GetPixelDynamicRangeMax() | 822 |
| 10.89.3.19 GetPixelDynamicRangeMin() | 822 |
| 10.89.3.20 GetScan3dAxisMax() | 822 |
| 10.89.3.21 GetScan3dAxisMin() | 822 |
| 10.89.3.22 GetScan3dCoordinateOffset() | 822 |
| 10.89.3.23 GetScan3dCoordinateReferenceValue() | 823 |
| 10.89.3.24 GetScan3dCoordinateScale() | 823 |
| 10.89.3.25 GetScan3dInvalidDataValue() | 823 |
| 10.89.3.26 GetScan3dTransformValue() | 823 |
| 10.89.3.27 GetScanLineSelector() | 823 |
| 10.89.3.28 GetSequencerSetActive() | 823 |
| 10.89.3.29 GetSerialDataLength() | 824 |
| 10.89.3.30 GetStreamChannelID() | 824 |
| 10.89.3.31 GetTimerValue() | 824 |
| 10.89.3.32 GetTimestamp() | 824 |
| 10.89.3.33 GetTimestampLatchValue() | 824 |
| 10.89.3.34 GetTransferBlockID() | 824 |
| 10.89.3.35 GetTransferQueueCurrentBlockCount() | 825 |
| 10.89.3.36 GetWidth() | 825 |

| | |
|--|-----|
| 10.89.3.37SetChunks() | 825 |
| 10.90IDataStream Class Reference | 825 |
| 10.90.1 Constructor & Destructor Documentation | 826 |
| 10.90.1.1 ~IDataStream() | 826 |
| 10.90.1.2 IDataStream() | 826 |
| 10.90.2 Member Function Documentation | 826 |
| 10.90.2.1 AddChunks() | 826 |
| 10.90.2.2 AnnounceImage() [1/3] | 826 |
| 10.90.2.3 AnnounceImage() [2/3] | 826 |
| 10.90.2.4 AnnounceImage() [3/3] | 827 |
| 10.90.2.5 CleanupChunkAdapter() | 827 |
| 10.90.2.6 FillCRCInfo() | 827 |
| 10.90.2.7 FlushQueueAllDiscard() | 827 |
| 10.90.2.8 GetNextImage() | 827 |
| 10.90.2.9 GetNextImageInternal() | 827 |
| 10.90.2.10GetNodeMap() | 827 |
| 10.90.2.11GetNumImagesInUse() | 828 |
| 10.90.2.12GetPort() | 828 |
| 10.90.2.13InitChunkAdapter() | 828 |
| 10.90.2.14IsImageInUse() | 828 |
| 10.90.2.15IsStreaming() | 828 |
| 10.90.2.16KillBufferEvent() | 828 |
| 10.90.2.17RegisterImageEvent() | 828 |
| 10.90.2.18ReleaseImage() | 829 |
| 10.90.2.19RevokeImages() | 829 |
| 10.90.2.20StartStream() | 829 |
| 10.90.2.21StopStream() | 829 |
| 10.90.2.22TransportLayerStreamInfo() | 829 |
| 10.90.2.23UnregisterImageEvent() | 829 |
| 10.90.2.24WaitOnImageEvent() | 829 |

| | |
|---|-----|
| 10.91 IDevFileStreamBase< CharType, Traits > Class Template Reference | 830 |
| 10.91.1 Member Typedef Documentation | 831 |
| 10.91.1.1 filebuf_type | 831 |
| 10.91.1.2 ios_type | 831 |
| 10.91.1.3 istream_type | 831 |
| 10.91.2 Member Function Documentation | 831 |
| 10.91.2.1 close() | 831 |
| 10.91.2.2 is_open() | 831 |
| 10.91.2.3 open() | 831 |
| 10.91.2.4 rdbuf() | 832 |
| 10.92 IDevFileStreamBuf< CharType, Traits > Class Template Reference | 832 |
| 10.92.1 Constructor & Destructor Documentation | 833 |
| 10.92.1.1 IDevFileStreamBuf() | 833 |
| 10.92.1.2 ~IDevFileStreamBuf() | 833 |
| 10.92.2 Member Function Documentation | 833 |
| 10.92.2.1 close() | 833 |
| 10.92.2.2 is_open() | 833 |
| 10.92.2.3 open() | 834 |
| 10.92.2.4 pbackfail() | 834 |
| 10.92.2.5 underflow() | 834 |
| 10.93 IDeviceEvent Class Reference | 834 |
| 10.93.1 Constructor & Destructor Documentation | 835 |
| 10.93.1.1 ~IDeviceEvent() | 835 |
| 10.93.1.2 IDeviceEvent() [1/2] | 835 |
| 10.93.1.3 IDeviceEvent() [2/2] | 836 |
| 10.93.2 Member Function Documentation | 836 |
| 10.93.2.1 GetDeviceEventId() | 836 |
| 10.93.2.2 GetDeviceEventName() | 836 |
| 10.93.2.3 OnDeviceEvent() | 836 |
| 10.93.2.4 operator=() | 836 |

| | |
|--|-----|
| 10.94IImage Class Reference | 837 |
| 10.94.1 Detailed Description | 838 |
| 10.94.2 Constructor & Destructor Documentation | 838 |
| 10.94.2.1 ~IImage() | 838 |
| 10.94.2.2 IImage() | 839 |
| 10.94.3 Member Function Documentation | 839 |
| 10.94.3.1 CalculateStatistics() | 839 |
| 10.94.3.2 CheckCRC() | 839 |
| 10.94.3.3 Convert() | 839 |
| 10.94.3.4 DeepCopy() | 839 |
| 10.94.3.5 DEPRECATED_FUNC() [1/3] | 840 |
| 10.94.3.6 DEPRECATED_FUNC() [2/3] | 840 |
| 10.94.3.7 DEPRECATED_FUNC() [3/3] | 840 |
| 10.94.3.8 GetBitsPerPixel() | 840 |
| 10.94.3.9 GetBufferSize() | 840 |
| 10.94.3.10GetChunkData() | 840 |
| 10.94.3.11GetChunkLayoutId() | 841 |
| 10.94.3.12GetColorProcessing() | 841 |
| 10.94.3.13GetData() | 841 |
| 10.94.3.14GetFrameId() | 841 |
| 10.94.3.15GetHeight() | 841 |
| 10.94.3.16GetID() | 841 |
| 10.94.3.17GetImageSize() | 842 |
| 10.94.3.18GetImageStatus() | 842 |
| 10.94.3.19GetNumChannels() | 842 |
| 10.94.3.20GetPayloadType() | 842 |
| 10.94.3.21GetPixelFormat() | 842 |
| 10.94.3.22GetPixelFormatIntType() | 842 |
| 10.94.3.23GetPixelFormatName() | 843 |
| 10.94.3.24GetPrivateData() | 843 |

| | | |
|------------|--|-----|
| 10.94.3.25 | GetStride() | 843 |
| 10.94.3.26 | GetTimeStamp() | 843 |
| 10.94.3.27 | GetTLPayloadType() | 843 |
| 10.94.3.28 | GetTLPixelFormat() | 843 |
| 10.94.3.29 | GetTLPixelFormatNamespace() | 844 |
| 10.94.3.30 | GetValidPayloadSize() | 844 |
| 10.94.3.31 | GetWidth() | 844 |
| 10.94.3.32 | GetXOffset() | 844 |
| 10.94.3.33 | GetXPadding() | 844 |
| 10.94.3.34 | GetYOffset() | 844 |
| 10.94.3.35 | GetYPadding() | 845 |
| 10.94.3.36 | HasCRC() | 845 |
| 10.94.3.37 | IsIncomplete() | 845 |
| 10.94.3.38 | IsInUse() | 845 |
| 10.94.3.39 | Release() | 845 |
| 10.94.3.40 | ResetImage() [1/2] | 845 |
| 10.94.3.41 | ResetImage() [2/2] | 846 |
| 10.94.3.42 | Save() [1/8] | 846 |
| 10.94.3.43 | Save() [2/8] | 846 |
| 10.94.3.44 | Save() [3/8] | 846 |
| 10.94.3.45 | Save() [4/8] | 846 |
| 10.94.3.46 | Save() [5/8] | 847 |
| 10.94.3.47 | Save() [6/8] | 847 |
| 10.94.3.48 | Save() [7/8] | 847 |
| 10.94.3.49 | Save() [8/8] | 847 |
| 10.95 | ImageEvent Class Reference | 848 |
| 10.95.1 | Constructor & Destructor Documentation | 849 |
| 10.95.1.1 | ~ImageEvent() | 849 |
| 10.95.1.2 | ImageEvent() [1/2] | 849 |
| 10.95.1.3 | ImageEvent() [2/2] | 849 |

| | |
|--|-----|
| 10.95.2 Member Function Documentation | 849 |
| 10.95.2.1 OnImageEvent() | 849 |
| 10.95.2.2 operator=() | 849 |
| 10.96 IImageStatistics Class Reference | 850 |
| 10.96.1 Detailed Description | 850 |
| 10.96.2 Constructor & Destructor Documentation | 851 |
| 10.96.2.1 ~IImageStatistics() | 851 |
| 10.96.2.2 IImageStatistics() [1/2] | 851 |
| 10.96.2.3 IImageStatistics() [2/2] | 851 |
| 10.96.3 Member Function Documentation | 851 |
| 10.96.3.1 DisableAll() | 851 |
| 10.96.3.2 EnableAll() | 851 |
| 10.96.3.3 EnableGreyOnly() | 851 |
| 10.96.3.4 EnableHSLOnly() | 852 |
| 10.96.3.5 EnableRGBOnly() | 852 |
| 10.96.3.6 GetChannelStatus() | 852 |
| 10.96.3.7 GetHistogram() | 852 |
| 10.96.3.8 GetMean() | 852 |
| 10.96.3.9 GetNumPixelValues() | 853 |
| 10.96.3.10 GetPixelValueRange() | 853 |
| 10.96.3.11 GetRange() | 853 |
| 10.96.3.12 GetStatistics() | 853 |
| 10.96.3.13 SetChannelStatus() | 854 |
| 10.97 IInterface Class Reference | 854 |
| 10.97.1 Detailed Description | 855 |
| 10.97.2 Constructor & Destructor Documentation | 855 |
| 10.97.2.1 ~IInterface() | 855 |
| 10.97.2.2 IInterface() [1/2] | 856 |
| 10.97.2.3 IInterface() [2/2] | 856 |
| 10.97.3 Member Function Documentation | 856 |

| | |
|--|-----|
| 10.97.3.1 GetCameras() | 856 |
| 10.97.3.2 GetTLNodeMap() | 856 |
| 10.97.3.3 IsInUse() | 856 |
| 10.97.3.4 operator=() | 856 |
| 10.97.3.5 RegisterEvent() | 857 |
| 10.97.3.6 SendActionCommand() | 857 |
| 10.97.3.7 UnregisterEvent() | 857 |
| 10.97.3.8 UpdateCameras() | 857 |
| 10.97.4 Friends And Related Function Documentation | 857 |
| 10.97.4.1 InterfaceInternal | 857 |
| 10.97.5 Member Data Documentation | 858 |
| 10.97.5.1 m_pInterfaceData | 858 |
| 10.97.5.2 TLInterface | 858 |
| 10.98IInterfaceEvent Class Reference | 858 |
| 10.98.1 Constructor & Destructor Documentation | 859 |
| 10.98.1.1 ~IInterfaceEvent() | 859 |
| 10.98.1.2 IInterfaceEvent() [1/2] | 859 |
| 10.98.1.3 IInterfaceEvent() [2/2] | 860 |
| 10.98.2 Member Function Documentation | 860 |
| 10.98.2.1 OnDeviceArrival() | 860 |
| 10.98.2.2 OnDeviceRemoval() | 860 |
| 10.98.2.3 operator=() | 860 |
| 10.99IInterfaceList Class Reference | 861 |
| 10.99.1 Detailed Description | 861 |
| 10.99.2 Constructor & Destructor Documentation | 861 |
| 10.99.2.1 ~IInterfaceList() | 862 |
| 10.99.2.2 IInterfaceList() [1/2] | 862 |
| 10.99.2.3 IInterfaceList() [2/2] | 862 |
| 10.99.3 Member Function Documentation | 862 |
| 10.99.3.1 Clear() | 862 |

| | | |
|------------|--|-----|
| 10.99.3.2 | GetByIndex() | 862 |
| 10.99.3.3 | GetSize() | 862 |
| 10.99.3.4 | operator=() | 863 |
| 10.99.3.5 | operator[]() | 863 |
| 10.99.4 | Member Data Documentation | 863 |
| 10.99.4.1 | m_pInterfaceListData | 863 |
| 10.100 | LoggingEvent Class Reference | 863 |
| 10.100.1 | Constructor & Destructor Documentation | 864 |
| 10.100.1.1 | ~ILoggingEvent() | 864 |
| 10.100.1.2 | ILoggingEvent() [1/2] | 864 |
| 10.100.1.3 | ILoggingEvent() [2/2] | 865 |
| 10.100.2 | Member Function Documentation | 865 |
| 10.100.2.1 | OnLogEvent() | 865 |
| 10.100.2.2 | operator=() | 865 |
| 10.101 | Image Class Reference | 865 |
| 10.101.1 | Detailed Description | 869 |
| 10.101.2 | Constructor & Destructor Documentation | 869 |
| 10.101.2.1 | ~Image() | 870 |
| 10.101.2.2 | Image() [1/3] | 870 |
| 10.101.2.3 | Image() [2/3] | 870 |
| 10.101.2.4 | Image() [3/3] | 870 |
| 10.101.3 | Member Function Documentation | 870 |
| 10.101.3.1 | CalculateStatistics() | 870 |
| 10.101.3.2 | CheckCRC() | 871 |
| 10.101.3.3 | Convert() [1/2] | 871 |
| 10.101.3.4 | Convert() [2/2] | 871 |
| 10.101.3.5 | Create() [1/3] | 872 |
| 10.101.3.6 | Create() [2/3] | 872 |
| 10.101.3.7 | Create() [3/3] | 872 |
| 10.101.3.8 | CreateShared() | 872 |

| | |
|--|-----|
| 10.101.3.9DeepCopy() [1/2] | 873 |
| 10.101.3.10DeepCopy() [2/2] | 873 |
| 10.101.3.11DEPRECATED_FUNC() [1/9] | 873 |
| 10.101.3.12DEPRECATED_FUNC() [2/9] | 874 |
| 10.101.3.13DEPRECATED_FUNC() [3/9] | 874 |
| 10.101.3.14DEPRECATED_FUNC() [4/9] | 874 |
| 10.101.3.15DEPRECATED_FUNC() [5/9] | 875 |
| 10.101.3.16DEPRECATED_FUNC() [6/9] | 875 |
| 10.101.3.17DEPRECATED_FUNC() [7/9] | 876 |
| 10.101.3.18DEPRECATED_FUNC() [8/9] | 876 |
| 10.101.3.19DEPRECATED_FUNC() [9/9] | 877 |
| 10.101.3.20GetBitsPerPixel() | 877 |
| 10.101.3.21GetBufferSize() | 877 |
| 10.101.3.22GetChunkData() | 878 |
| 10.101.3.23GetChunkLayoutId() | 878 |
| 10.101.3.24GetColorProcessing() | 878 |
| 10.101.3.25SetData() | 879 |
| 10.101.3.26GetDefaultColorProcessing() | 879 |
| 10.101.3.27GetFrameId() | 879 |
| 10.101.3.28GetHeight() | 880 |
| 10.101.3.29GetId() | 880 |
| 10.101.3.30GetImageSize() | 880 |
| 10.101.3.31GetImageStatus() | 881 |
| 10.101.3.32GetImageStatusDescription() | 881 |
| 10.101.3.33GetNumChannels() | 881 |
| 10.101.3.34GetPayloadType() | 882 |
| 10.101.3.35SetPixelFormat() | 882 |
| 10.101.3.36GetPixelFormatIntType() | 882 |
| 10.101.3.37GetPixelFormatName() | 883 |
| 10.101.3.38GetPrivateData() | 883 |

| | | |
|-------------|--|-----|
| 10.101.3.39 | GetStride() | 883 |
| 10.101.3.40 | GetTimeStamp() | 884 |
| 10.101.3.41 | GetTLPayloadType() | 884 |
| 10.101.3.42 | GetTLPixelFormat() | 884 |
| 10.101.3.43 | GetTLPixelFormatNamespace() | 885 |
| 10.101.3.44 | GetValidPayloadSize() | 885 |
| 10.101.3.45 | GetWidth() | 885 |
| 10.101.3.46 | GetXOffset() | 886 |
| 10.101.3.47 | GetXPadding() | 886 |
| 10.101.3.48 | GetYOffset() | 886 |
| 10.101.3.49 | GetYPadding() | 887 |
| 10.101.3.50 | HasCRC() | 887 |
| 10.101.3.51 | IsCompressed() | 887 |
| 10.101.3.52 | IsIncomplete() | 887 |
| 10.101.3.53 | IsInUse() | 888 |
| 10.101.3.54 | Release() | 888 |
| 10.101.3.55 | ResetImage() [1/2] | 888 |
| 10.101.3.56 | ResetImage() [2/2] | 888 |
| 10.101.3.57 | Save() [1/8] | 889 |
| 10.101.3.58 | Save() [2/8] | 889 |
| 10.101.3.59 | Save() [3/8] | 890 |
| 10.101.3.60 | Save() [4/8] | 890 |
| 10.101.3.61 | Save() [5/8] | 890 |
| 10.101.3.62 | Save() [6/8] | 891 |
| 10.101.3.63 | Save() [7/8] | 891 |
| 10.101.3.64 | Save() [8/8] | 891 |
| 10.101.3.65 | SetDefaultColorProcessing() | 892 |
| 10.101.4 | Friends And Related Function Documentation | 892 |
| 10.101.4.1 | DataStream | 892 |
| 10.101.4.2 | ImageConverter | 892 |

| | | |
|------------|--|-----|
| 10.101.4.3 | ImageFiler | 892 |
| 10.101.4.4 | ImageStatsCalculator | 893 |
| 10.101.4.5 | ImageUtilityImpl | 893 |
| 10.101.4.6 | Stream | 893 |
| 10.101.5 | Member Data Documentation | 893 |
| 10.101.5.1 | m_pImageData | 893 |
| 10.102 | ImageEvent Class Reference | 893 |
| 10.102.1 | Detailed Description | 894 |
| 10.102.2 | Constructor & Destructor Documentation | 894 |
| 10.102.2.1 | ImageEvent() | 895 |
| 10.102.2.2 | ~ImageEvent() | 895 |
| 10.102.3 | Member Function Documentation | 895 |
| 10.102.3.1 | OnImageEvent() | 895 |
| 10.102.3.2 | operator=() | 895 |
| 10.103 | ImagePtr Class Reference | 896 |
| 10.103.1 | Detailed Description | 897 |
| 10.103.2 | Constructor & Destructor Documentation | 897 |
| 10.103.2.1 | ImagePtr() [1/4] | 897 |
| 10.103.2.2 | ImagePtr() [2/4] | 897 |
| 10.103.2.3 | ImagePtr() [3/4] | 897 |
| 10.103.2.4 | ImagePtr() [4/4] | 897 |
| 10.103.2.5 | ~ImagePtr() | 898 |
| 10.103.3 | Member Function Documentation | 898 |
| 10.103.3.1 | operator=() | 898 |
| 10.104 | ImageStatistics Class Reference | 898 |
| 10.104.1 | Detailed Description | 899 |
| 10.104.2 | Constructor & Destructor Documentation | 900 |
| 10.104.2.1 | ImageStatistics() [1/2] | 900 |
| 10.104.2.2 | ~ImageStatistics() | 900 |
| 10.104.2.3 | ImageStatistics() [2/2] | 900 |

| | | |
|-------------|--|-----|
| 10.104.3 | Member Function Documentation | 900 |
| 10.104.3.1 | DisableAll() | 900 |
| 10.104.3.2 | EnableAll() | 900 |
| 10.104.3.3 | EnableGreyOnly() | 901 |
| 10.104.3.4 | EnableHSLOnly() | 901 |
| 10.104.3.5 | EnableRGBOnly() | 901 |
| 10.104.3.6 | GetChannelStatus() | 901 |
| 10.104.3.7 | GetHistogram() | 902 |
| 10.104.3.8 | GetMean() | 902 |
| 10.104.3.9 | GetNumPixelValues() | 902 |
| 10.104.3.10 | GetPixelValueRange() | 903 |
| 10.104.3.11 | GetRange() | 903 |
| 10.104.3.12 | GetStatistics() | 903 |
| 10.104.3.13 | operator=() | 904 |
| 10.104.3.14 | SetChannelStatus() | 904 |
| 10.104.4 | Friends And Related Function Documentation | 905 |
| 10.104.4.1 | ImageStatsCalculator | 905 |
| 10.105 | ImageUtility Class Reference | 905 |
| 10.105.1 | Detailed Description | 906 |
| 10.105.2 | Member Enumeration Documentation | 906 |
| 10.105.2.1 | ImageScalingAlgorithm | 906 |
| 10.105.3 | Member Function Documentation | 906 |
| 10.105.3.1 | CreateNormalized() [1/5] | 906 |
| 10.105.3.2 | CreateNormalized() [2/5] | 907 |
| 10.105.3.3 | CreateNormalized() [3/5] | 907 |
| 10.105.3.4 | CreateNormalized() [4/5] | 908 |
| 10.105.3.5 | CreateNormalized() [5/5] | 908 |
| 10.105.3.6 | CreateScaled() [1/2] | 908 |
| 10.105.3.7 | CreateScaled() [2/2] | 909 |
| 10.106 | ImageUtilityHeatmap Class Reference | 909 |

| | |
|--|-----|
| 10.106.1Detailed Description | 910 |
| 10.106.2Member Enumeration Documentation | 910 |
| 10.106.2.1HeatmapColor | 910 |
| 10.106.3Member Function Documentation | 910 |
| 10.106.3.1CreateHeatmap() [1/2] | 911 |
| 10.106.3.2CreateHeatmap() [2/2] | 911 |
| 10.106.3.3GetHeatmapColorGradient() | 912 |
| 10.106.3.4GetHeatmapRange() | 912 |
| 10.106.3.5SetHeatmapColorGradient() | 912 |
| 10.106.3.6SetHeatmapRange() | 913 |
| 10.107ImageUtilityPolarization Class Reference | 913 |
| 10.107.1Detailed Description | 914 |
| 10.107.2Member Enumeration Documentation | 914 |
| 10.107.2.1PolarizationQuadrant | 914 |
| 10.107.3Member Function Documentation | 915 |
| 10.107.3.1CreateAolp() [1/2] | 915 |
| 10.107.3.2CreateAolp() [2/2] | 915 |
| 10.107.3.3CreateDolp() [1/2] | 916 |
| 10.107.3.4CreateDolp() [2/2] | 916 |
| 10.107.3.5CreateStokesS0() [1/2] | 917 |
| 10.107.3.6CreateStokesS0() [2/2] | 917 |
| 10.107.3.7CreateStokesS1() [1/2] | 917 |
| 10.107.3.8CreateStokesS1() [2/2] | 919 |
| 10.107.3.9CreateStokesS2() [1/2] | 919 |
| 10.107.3.10CreateStokesS2() [2/2] | 920 |
| 10.107.3.11ExtractPolarQuadrant() [1/2] | 920 |
| 10.107.3.12ExtractPolarQuadrant() [2/2] | 921 |
| 10.108ht64_autovector_t Class Reference | 921 |
| 10.108.1Detailed Description | 921 |
| 10.108.2Constructor & Destructor Documentation | 922 |

| | |
|--|-----|
| 10.108.2.1int64_autovector_t() [1/3] | 922 |
| 10.108.2.2int64_autovector_t() [2/3] | 922 |
| 10.108.2.3int64_autovector_t() [3/3] | 922 |
| 10.108.2.4~int64_autovector_t() | 922 |
| 10.108.3Member Function Documentation | 922 |
| 10.108.3.1operator delete() | 922 |
| 10.108.3.2operator new() | 922 |
| 10.108.3.3operator=() | 923 |
| 10.108.3.4operator[]() [1/2] | 923 |
| 10.108.3.5operator[]() [2/2] | 923 |
| 10.108.3.6size() | 923 |
| 10.108.4Member Data Documentation | 923 |
| 10.108.4.1_pCount | 923 |
| 10.108.4.2_pv | 923 |
| 10.109IntegerNode Class Reference | 924 |
| 10.109.1Detailed Description | 926 |
| 10.109.2Constructor & Destructor Documentation | 926 |
| 10.109.2.1IntegerNode() [1/2] | 926 |
| 10.109.2.2IntegerNode() [2/2] | 926 |
| 10.109.2.3~IntegerNode() | 926 |
| 10.109.3Member Function Documentation | 926 |
| 10.109.3.1GetFloatAlias() | 927 |
| 10.109.3.2GetInc() | 927 |
| 10.109.3.3GetIncMode() | 927 |
| 10.109.3.4GetListOfValidValues() | 927 |
| 10.109.3.5GetMax() | 927 |
| 10.109.3.6GetMin() | 927 |
| 10.109.3.7GetRepresentation() | 928 |
| 10.109.3.8GetUnit() | 928 |
| 10.109.3.9GetValue() | 928 |

| | | |
|-------------|--|-----|
| 10.109.3.10 | ImposeMax() | 928 |
| 10.109.3.11 | ImposeMin() | 928 |
| 10.109.3.12 | operator() | 929 |
| 10.109.3.13 | operator*() | 929 |
| 10.109.3.14 | operator=() | 929 |
| 10.109.3.15 | SetReference() | 929 |
| 10.109.3.16 | SetValue() | 929 |
| 10.110 | Interface Class Reference | 930 |
| 10.110.1 | Detailed Description | 931 |
| 10.110.2 | Constructor & Destructor Documentation | 931 |
| 10.110.2.1 | ~Interface() | 931 |
| 10.110.3 | Member Function Documentation | 931 |
| 10.110.3.1 | GetCameras() | 932 |
| 10.110.3.2 | GetTLNodeMap() | 932 |
| 10.110.3.3 | IsInUse() | 932 |
| 10.110.3.4 | RegisterEvent() | 933 |
| 10.110.3.5 | SendActionCommand() | 933 |
| 10.110.3.6 | UnregisterEvent() | 933 |
| 10.110.3.7 | UpdateCameras() | 934 |
| 10.110.4 | Friends And Related Function Documentation | 934 |
| 10.110.4.1 | InterfaceInternal | 934 |
| 10.111 | InterfaceEvent Class Reference | 935 |
| 10.111.1 | Detailed Description | 936 |
| 10.111.2 | Constructor & Destructor Documentation | 936 |
| 10.111.2.1 | InterfaceEvent() | 936 |
| 10.111.2.2 | ~InterfaceEvent() | 936 |
| 10.111.3 | Member Function Documentation | 936 |
| 10.111.3.1 | OnDeviceArrival() | 937 |
| 10.111.3.2 | OnDeviceRemoval() | 937 |
| 10.111.3.3 | operator=() | 937 |

| | | |
|------------|--|-----|
| 10.112 | InterfaceList Class Reference | 937 |
| 10.112.1 | Detailed Description | 939 |
| 10.112.2 | Constructor & Destructor Documentation | 939 |
| 10.112.2.1 | InterfaceList() [1/2] | 939 |
| 10.112.2.2 | ~InterfaceList() | 939 |
| 10.112.2.3 | InterfaceList() [2/2] | 939 |
| 10.112.3 | Member Function Documentation | 939 |
| 10.112.3.1 | Clear() | 939 |
| 10.112.3.2 | GetByIndex() | 939 |
| 10.112.3.3 | GetSize() | 940 |
| 10.112.3.4 | operator=() | 940 |
| 10.112.3.5 | operator[]() | 940 |
| 10.112.4 | Friends And Related Function Documentation | 940 |
| 10.112.4.1 | SystemImpl | 941 |
| 10.113 | InterfacePtr Class Reference | 941 |
| 10.113.1 | Detailed Description | 942 |
| 10.113.2 | Constructor & Destructor Documentation | 942 |
| 10.113.2.1 | InterfacePtr() [1/4] | 942 |
| 10.113.2.2 | InterfacePtr() [2/4] | 942 |
| 10.113.2.3 | InterfacePtr() [3/4] | 942 |
| 10.113.2.4 | InterfacePtr() [4/4] | 942 |
| 10.114 | IntRegNode Class Reference | 943 |
| 10.114.1 | Detailed Description | 944 |
| 10.114.2 | Constructor & Destructor Documentation | 944 |
| 10.114.2.1 | IntRegNode() [1/2] | 945 |
| 10.114.2.2 | IntRegNode() [2/2] | 945 |
| 10.114.2.3 | ~IntRegNode() | 945 |
| 10.114.3 | Member Function Documentation | 945 |
| 10.114.3.1 | SetReference() | 945 |
| 10.115 | RemovalEvent Class Reference | 946 |

| | | |
|-------------|--|-----|
| 10.115.1 | Constructor & Destructor Documentation | 947 |
| 10.115.1.1 | ~IRemovalEvent() | 947 |
| 10.115.1.2 | RemovalEvent() [1/2] | 947 |
| 10.115.1.3 | RemovalEvent() [2/2] | 947 |
| 10.115.2 | Member Function Documentation | 947 |
| 10.115.2.1 | OnDeviceRemoval() | 947 |
| 10.115.2.2 | operator=() | 947 |
| 10.116 | System Class Reference | 948 |
| 10.116.1 | Detailed Description | 949 |
| 10.116.2 | Constructor & Destructor Documentation | 949 |
| 10.116.2.1 | ~ISystem() | 949 |
| 10.116.2.2 | ISystem() [1/2] | 949 |
| 10.116.2.3 | ISystem() [2/2] | 949 |
| 10.116.3 | Member Function Documentation | 949 |
| 10.116.3.1 | GetCameras() | 950 |
| 10.116.3.2 | GetInterfaces() | 950 |
| 10.116.3.3 | GetLibraryVersion() | 950 |
| 10.116.3.4 | GetLoggingEventPriorityLevel() | 950 |
| 10.116.3.5 | GetTLNodeMap() | 950 |
| 10.116.3.6 | IsInUse() | 950 |
| 10.116.3.7 | operator=() | 951 |
| 10.116.3.8 | RegisterInterfaceEvent() | 951 |
| 10.116.3.9 | RegisterLoggingEvent() | 951 |
| 10.116.3.10 | ReleaseInstance() | 951 |
| 10.116.3.11 | SendActionCommand() | 951 |
| 10.116.3.12 | SetLoggingEventPriorityLevel() | 952 |
| 10.116.3.13 | UnregisterAllLoggingEvent() | 952 |
| 10.116.3.14 | UnregisterInterfaceEvent() | 952 |
| 10.116.3.15 | UnregisterLoggingEvent() | 952 |
| 10.116.3.16 | UpdateCameras() | 952 |

| | |
|--|-----|
| 10.116.3.17updateInterfaceList() | 952 |
| 10.116.4Friends And Related Function Documentation | 953 |
| 10.116.4.1SystemPtrInternal | 953 |
| 10.116.5Member Data Documentation | 953 |
| 10.116.5.1TLSystem | 953 |
| 10.117JPEGOption Struct Reference | 953 |
| 10.117.1Detailed Description | 953 |
| 10.117.2Constructor & Destructor Documentation | 954 |
| 10.117.2.1JPEGOption() | 954 |
| 10.117.3Member Data Documentation | 954 |
| 10.117.3.1progressive | 954 |
| 10.117.3.2quality | 954 |
| 10.117.3.3reserved | 954 |
| 10.118PG2Option Struct Reference | 955 |
| 10.118.1Detailed Description | 955 |
| 10.118.2Constructor & Destructor Documentation | 955 |
| 10.118.2.1JPG2Option() | 955 |
| 10.118.3Member Data Documentation | 955 |
| 10.118.3.1quality | 955 |
| 10.118.3.2reserved | 956 |
| 10.119LibraryVersion Struct Reference | 956 |
| 10.119.1Detailed Description | 956 |
| 10.119.2Member Data Documentation | 956 |
| 10.119.2.1build | 956 |
| 10.119.2.2major | 957 |
| 10.119.2.3minor | 957 |
| 10.119.2.4type | 957 |
| 10.120LockableObject< Object >::Lock Class Reference | 957 |
| 10.120.1Detailed Description | 957 |
| 10.120.2Constructor & Destructor Documentation | 958 |

| | |
|---|-----|
| 10.120.2.1Lock() | 958 |
| 10.120.2.2~Lock() | 958 |
| 10.121LockableObject< Object > Class Template Reference | 958 |
| 10.121.1Detailed Description | 959 |
| 10.121.2Member Function Documentation | 959 |
| 10.121.2.1GetLock() | 959 |
| 10.121.3Friends And Related Function Documentation | 959 |
| 10.121.3.1Lock | 959 |
| 10.121.4Member Data Documentation | 959 |
| 10.121.4.1m_Lock | 960 |
| 10.122LoggingEvent Class Reference | 960 |
| 10.122.1Detailed Description | 961 |
| 10.122.2Constructor & Destructor Documentation | 961 |
| 10.122.2.1LoggingEvent() | 961 |
| 10.122.2.2~LoggingEvent() | 961 |
| 10.122.3Member Function Documentation | 961 |
| 10.122.3.1OnLogEvent() | 961 |
| 10.122.3.2operator=() | 962 |
| 10.123LoggingEventData Class Reference | 962 |
| 10.123.1Detailed Description | 963 |
| 10.123.2Constructor & Destructor Documentation | 963 |
| 10.123.2.1~LoggingEventData() | 963 |
| 10.123.2.2LoggingEventData() | 963 |
| 10.123.3Member Function Documentation | 963 |
| 10.123.3.1GetCategoryName() | 963 |
| 10.123.3.2GetLogMessage() | 964 |
| 10.123.3.3GetNDC() | 964 |
| 10.123.3.4GetPriority() | 964 |
| 10.123.3.5GetPriorityName() | 964 |
| 10.123.3.6GetThreadName() | 965 |

| | |
|--|-----|
| 10.123.3.7GetTimestamp() | 965 |
| 10.123.4Friends And Related Function Documentation | 965 |
| 10.123.4.1SystemImpl | 965 |
| 10.124LoggingEventDataPtr Class Reference | 966 |
| 10.124.1Detailed Description | 966 |
| 10.124.2Constructor & Destructor Documentation | 967 |
| 10.124.2.1LoggingEventDataPtr() [1/4] | 967 |
| 10.124.2.2LoggingEventDataPtr() [2/4] | 967 |
| 10.124.2.3LoggingEventDataPtr() [3/4] | 967 |
| 10.124.2.4LoggingEventDataPtr() [4/4] | 967 |
| 10.125Member_NodeCallback< Client, Member > Class Template Reference | 968 |
| 10.125.1Detailed Description | 969 |
| 10.125.2Member Typedef Documentation | 969 |
| 10.125.2.1PMEMBERFUNC | 969 |
| 10.125.3Constructor & Destructor Documentation | 969 |
| 10.125.3.1Member_NodeCallback() | 969 |
| 10.125.4Member Function Documentation | 969 |
| 10.125.4.1Destroy() | 969 |
| 10.125.4.2operator>() | 970 |
| 10.126MJPGOption Struct Reference | 970 |
| 10.126.1Detailed Description | 970 |
| 10.126.2Constructor & Destructor Documentation | 970 |
| 10.126.2.1MJPGOption() | 970 |
| 10.126.3Member Data Documentation | 970 |
| 10.126.3.1frameRate | 971 |
| 10.126.3.2quality | 971 |
| 10.126.3.3reserved | 971 |
| 10.127Node Class Reference | 971 |
| 10.127.1Detailed Description | 974 |
| 10.127.2Constructor & Destructor Documentation | 974 |

| | |
|--|-----|
| 10.127.2.1Node() [1/2] | 974 |
| 10.127.2.2Node() [2/2] | 974 |
| 10.127.2.3~Node() | 974 |
| 10.127.3Member Function Documentation | 974 |
| 10.127.3.1DeregisterCallback() | 975 |
| 10.127.3.2GetAccessMode() | 975 |
| 10.127.3.3GetAlias() | 975 |
| 10.127.3.4GetCachingMode() | 975 |
| 10.127.3.5GetCastAlias() | 975 |
| 10.127.3.6GetChildren() | 975 |
| 10.127.3.7GetDescription() | 976 |
| 10.127.3.8GetDeviceName() | 976 |
| 10.127.3.9GetDisplayName() | 976 |
| 10.127.3.10GetDocuURL() | 976 |
| 10.127.3.11GetEventID() | 976 |
| 10.127.3.12GetName() | 977 |
| 10.127.3.13GetNameSpace() | 977 |
| 10.127.3.14GetNodeHandle() | 977 |
| 10.127.3.15GetNodeMap() | 977 |
| 10.127.3.16GetParents() | 977 |
| 10.127.3.17GetPollingTime() | 978 |
| 10.127.3.18GetPrincipalInterfaceType() | 978 |
| 10.127.3.19GetProperty() | 978 |
| 10.127.3.20GetPropertyNames() | 978 |
| 10.127.3.21GetSelectedFeatures() | 978 |
| 10.127.3.22GetSelectingFeatures() | 979 |
| 10.127.3.23GetToolTip() | 979 |
| 10.127.3.24GetVisibility() | 979 |
| 10.127.3.25ImposeAccessMode() | 979 |
| 10.127.3.26ImposeVisibility() | 979 |

| | | |
|-------------|--|-----|
| 10.127.3.27 | validateNode() | 979 |
| 10.127.3.28 | AccessModeCacheable() | 980 |
| 10.127.3.29 | Cachable() | 980 |
| 10.127.3.30 | Deprecated() | 980 |
| 10.127.3.31 | Feature() | 980 |
| 10.127.3.32 | Selector() | 980 |
| 10.127.3.33 | Streamable() | 980 |
| 10.127.3.34 | operator!=() | 981 |
| 10.127.3.35 | operator==() | 981 |
| 10.127.3.36 | RegisterCallback() | 981 |
| 10.127.3.37 | SetNodeHandle() | 981 |
| 10.127.3.38 | SetNodeMap() | 981 |
| 10.127.3.39 | SetReference() [1/2] | 981 |
| 10.127.3.40 | SetReference() [2/2] | 982 |
| 10.127.4 | Member Data Documentation | 982 |
| 10.127.4.1 | m_Callbacks | 982 |
| 10.127.4.2 | m_pNodeData | 982 |
| 10.127.4.3 | m_pNodeMap | 982 |
| 10.128 | NodeMap Class Reference | 982 |
| 10.128.1 | Detailed Description | 984 |
| 10.128.2 | Constructor & Destructor Documentation | 984 |
| 10.128.2.1 | NodeMap() | 985 |
| 10.128.2.2 | ~NodeMap() | 985 |
| 10.128.3 | Member Function Documentation | 985 |
| 10.128.3.1 | ClearXMLCache() | 985 |
| 10.128.3.2 | Connect() [1/2] | 985 |
| 10.128.3.3 | Connect() [2/2] | 985 |
| 10.128.3.4 | Destroy() | 986 |
| 10.128.3.5 | GetDeviceName() | 986 |
| 10.128.3.6 | GetDeviceVersion() | 986 |

| | |
|---|-----|
| 10.128.3.7GetGenApiVersion() | 986 |
| 10.128.3.8GetLock() | 986 |
| 10.128.3.9GetModelName() | 986 |
| 10.128.3.10GetNode() | 987 |
| 10.128.3.11GetNodeMapHandle() | 987 |
| 10.128.3.12GetNodes() | 987 |
| 10.128.3.13GetNumNodes() | 987 |
| 10.128.3.14GetProductGuid() | 987 |
| 10.128.3.15GetSchemaVersion() | 987 |
| 10.128.3.16GetStandardNameSpace() | 988 |
| 10.128.3.17GetSupportedSchemaVersions() | 988 |
| 10.128.3.18GetToolTip() | 988 |
| 10.128.3.19GetVendorName() | 988 |
| 10.128.3.20GetVersionGuid() | 989 |
| 10.128.3.21InvalidateNodes() | 989 |
| 10.128.3.22LoadXMLFromFile() | 989 |
| 10.128.3.23LoadXMLFromFileInject() | 989 |
| 10.128.3.24LoadXMLFromString() | 989 |
| 10.128.3.25LoadXMLFromStringInject() | 990 |
| 10.128.3.26LoadXMLFromZIPData() | 990 |
| 10.128.3.27LoadXMLFromZIPFile() | 990 |
| 10.128.3.28Roll() | 990 |
| 10.128.4Member Data Documentation | 990 |
| 10.128.4.1_Ptr | 990 |
| 10.129NodeMapFactory::NodeStatistics_t Struct Reference | 991 |
| 10.129.1Member Data Documentation | 991 |
| 10.129.1.1NumLinks | 991 |
| 10.129.1.2NumNodes | 991 |
| 10.129.1.3NumProperties | 991 |
| 10.129.1.4NumStrings | 991 |

| | | |
|------------|---|-----|
| 10.130 | O DevFileStreamBase< CharType, Traits > Class Template Reference | 992 |
| 10.130.1 | Member Typedef Documentation | 993 |
| 10.130.1.1 | filebuf_type | 993 |
| 10.130.1.2 | os_type | 993 |
| 10.130.1.3 | ostream_type | 993 |
| 10.130.2 | Member Function Documentation | 993 |
| 10.130.2.1 | close() | 993 |
| 10.130.2.2 | rs_open() | 993 |
| 10.130.2.3 | open() | 993 |
| 10.130.2.4 | rdbuf() | 994 |
| 10.130 | O DevFileStreamBuf< CharType, Traits > Class Template Reference | 994 |
| 10.131.1 | Constructor & Destructor Documentation | 995 |
| 10.131.1.1 | O DevFileStreamBuf() | 995 |
| 10.131.1.2 | ~ O DevFileStreamBuf() | 995 |
| 10.131.2 | Member Function Documentation | 995 |
| 10.131.2.1 | close() | 995 |
| 10.131.2.2 | rs_open() | 995 |
| 10.131.2.3 | open() | 996 |
| 10.131.2.4 | overflow() | 996 |
| 10.131.2.5 | sync() | 996 |
| 10.131.2.6 | &xputn() | 996 |
| 10.132 | P GMOption Struct Reference | 996 |
| 10.132.1 | Detailed Description | 997 |
| 10.132.2 | Constructor & Destructor Documentation | 997 |
| 10.132.2.1 | P GMOption() | 997 |
| 10.132.3 | Member Data Documentation | 997 |
| 10.132.3.1 | binaryFile | 997 |
| 10.132.3.2 | reserved | 997 |
| 10.133 | P NGOption Struct Reference | 997 |
| 10.133.1 | Detailed Description | 998 |

| | | |
|-------------|--|------|
| 10.133.2 | Constructor & Destructor Documentation | 998 |
| 10.133.2.1 | PNGOption() | 998 |
| 10.133.3 | Member Data Documentation | 998 |
| 10.133.3.1 | compressionLevel | 998 |
| 10.133.3.2 | Interlaced | 998 |
| 10.133.3.3 | reserved | 999 |
| 10.134 | PortNode Class Reference | 999 |
| 10.134.1 | Detailed Description | 1001 |
| 10.134.2 | Constructor & Destructor Documentation | 1001 |
| 10.134.2.1 | PortNode() [1/2] | 1001 |
| 10.134.2.2 | PortNode() [2/2] | 1001 |
| 10.134.2.3 | ~PortNode() | 1001 |
| 10.134.3 | Member Function Documentation | 1001 |
| 10.134.3.1 | CacheChunkData() | 1001 |
| 10.134.3.2 | GetChunkID() | 1002 |
| 10.134.3.3 | GetPortHandle() | 1002 |
| 10.134.3.4 | GetSwapEndianness() | 1002 |
| 10.134.3.5 | Read() | 1002 |
| 10.134.3.6 | Replay() | 1002 |
| 10.134.3.7 | SetPortImpl() | 1003 |
| 10.134.3.8 | SetReference() [1/3] | 1003 |
| 10.134.3.9 | SetReference() [2/3] | 1003 |
| 10.134.3.10 | SetReference() [3/3] | 1003 |
| 10.134.3.11 | StartRecording() | 1003 |
| 10.134.3.12 | StopRecording() | 1004 |
| 10.134.3.13 | Write() | 1004 |
| 10.135 | PortRecorder Class Reference | 1004 |
| 10.135.1 | Detailed Description | 1005 |
| 10.135.2 | Constructor & Destructor Documentation | 1005 |
| 10.135.2.1 | PortRecorder() | 1006 |

| | |
|--|------|
| 10.135.2.2~PortRecorder() | 1006 |
| 10.135.3Member Function Documentation | 1006 |
| 10.135.3.1GetAccessMode() | 1006 |
| 10.135.3.2SetReference() | 1006 |
| 10.135.3.3StartRecording() | 1006 |
| 10.135.3.4StopRecording() | 1007 |
| 10.136PortReplay Class Reference | 1007 |
| 10.136.1Detailed Description | 1008 |
| 10.136.2Constructor & Destructor Documentation | 1008 |
| 10.136.2.1PortReplay() | 1008 |
| 10.136.2.2~PortReplay() | 1009 |
| 10.136.3Member Function Documentation | 1009 |
| 10.136.3.1GetPortReplayHandle() | 1009 |
| 10.136.3.2Replay() | 1009 |
| 10.136.3.3SetReference() | 1009 |
| 10.137PPMOption Struct Reference | 1009 |
| 10.137.1Detailed Description | 1010 |
| 10.137.2Constructor & Destructor Documentation | 1010 |
| 10.137.2.1PPMOption() | 1010 |
| 10.137.3Member Data Documentation | 1010 |
| 10.137.3.1binaryFile | 1010 |
| 10.137.3.2reserved | 1010 |
| 10.138RegisterNode Class Reference | 1011 |
| 10.138.1Detailed Description | 1012 |
| 10.138.2Constructor & Destructor Documentation | 1012 |
| 10.138.2.1RegisterNode() [1/2] | 1013 |
| 10.138.2.2RegisterNode() [2/2] | 1013 |
| 10.138.2.3~RegisterNode() | 1013 |
| 10.138.3Member Function Documentation | 1013 |
| 10.138.3.1Get() | 1013 |

| | | |
|------------|--|------|
| 10.138.3.2 | GetAddress() | 1013 |
| 10.138.3.3 | GetLength() | 1014 |
| 10.138.3.4 | Set() | 1014 |
| 10.138.3.5 | SetReference() | 1014 |
| 10.139 | RemovalEvent Class Reference | 1015 |
| 10.139.1 | Detailed Description | 1016 |
| 10.139.2 | Constructor & Destructor Documentation | 1016 |
| 10.139.2.1 | RemovalEvent() | 1016 |
| 10.139.2.2 | ~RemovalEvent() | 1016 |
| 10.139.3 | Member Function Documentation | 1016 |
| 10.139.3.1 | OnDeviceRemoval() | 1016 |
| 10.139.3.2 | operator=() | 1017 |
| 10.140 | SingleChunkData_t Struct Reference | 1017 |
| 10.140.1 | Member Data Documentation | 1017 |
| 10.140.1.1 | ChunkID | 1017 |
| 10.140.1.2 | ChunkLength | 1017 |
| 10.140.1.3 | ChunkOffset | 1017 |
| 10.141 | SingleChunkDataStr_t Struct Reference | 1018 |
| 10.141.1 | Member Data Documentation | 1018 |
| 10.141.1.1 | ChunkID | 1018 |
| 10.141.1.2 | ChunkLength | 1018 |
| 10.141.1.3 | ChunkOffset | 1018 |
| 10.142 | SpinTestCamera Class Reference | 1019 |
| 10.143 | SpinVideo Class Reference | 1019 |
| 10.143.1 | Detailed Description | 1020 |
| 10.143.2 | Constructor & Destructor Documentation | 1020 |
| 10.143.2.1 | SpinVideo() | 1020 |
| 10.143.2.2 | ~SpinVideo() | 1020 |
| 10.143.3 | Member Function Documentation | 1020 |
| 10.143.3.1 | Append() | 1020 |

| | | |
|------------|--|------|
| 10.143.3.2 | Close() | 1021 |
| 10.143.3.3 | Open() [1/3] | 1021 |
| 10.143.3.4 | Open() [2/3] | 1021 |
| 10.143.3.5 | Open() [3/3] | 1023 |
| 10.143.3.6 | SetMaximumFileSize() | 1023 |
| 10.144 | StringNode Class Reference | 1024 |
| 10.144.1 | Detailed Description | 1025 |
| 10.144.2 | Constructor & Destructor Documentation | 1026 |
| 10.144.2.1 | StringNode() [1/2] | 1026 |
| 10.144.2.2 | StringNode() [2/2] | 1026 |
| 10.144.2.3 | ~StringNode() | 1026 |
| 10.144.3 | Member Function Documentation | 1026 |
| 10.144.3.1 | GetMaxLength() | 1026 |
| 10.144.3.2 | GetValue() | 1026 |
| 10.144.3.3 | operator()() | 1027 |
| 10.144.3.4 | operator*() | 1027 |
| 10.144.3.5 | operator=() | 1027 |
| 10.144.3.6 | SetReference() | 1027 |
| 10.144.3.7 | SetValue() | 1027 |
| 10.145 | StringRegNode Class Reference | 1028 |
| 10.145.1 | Detailed Description | 1029 |
| 10.145.2 | Constructor & Destructor Documentation | 1029 |
| 10.145.2.1 | StringRegNode() [1/2] | 1030 |
| 10.145.2.2 | StringRegNode() [2/2] | 1030 |
| 10.145.2.3 | ~StringRegNode() | 1030 |
| 10.145.3 | Member Function Documentation | 1030 |
| 10.145.3.1 | SetReference() | 1030 |
| 10.146 | System Class Reference | 1031 |
| 10.146.1 | Detailed Description | 1032 |
| 10.146.2 | Constructor & Destructor Documentation | 1032 |

| | |
|--|------|
| 10.146.2.1~System() | 1033 |
| 10.146.2.2System() | 1033 |
| 10.146.3Member Function Documentation | 1033 |
| 10.146.3.1GetCameras() | 1033 |
| 10.146.3.2GetInstance() | 1034 |
| 10.146.3.3GetInterfaces() | 1034 |
| 10.146.3.4GetLibraryVersion() | 1034 |
| 10.146.3.5GetLoggingEventPriorityLevel() | 1035 |
| 10.146.3.6GetTLNodeMap() | 1035 |
| 10.146.3.7IsInUse() | 1036 |
| 10.146.3.8RegisterInterfaceEvent() | 1036 |
| 10.146.3.9RegisterLoggingEvent() | 1036 |
| 10.146.3.10ReleaseInstance() | 1036 |
| 10.146.3.11SendActionCommand() | 1037 |
| 10.146.3.12SetLoggingEventPriorityLevel() | 1037 |
| 10.146.3.13UnregisterAllLoggingEvent() | 1038 |
| 10.146.3.14UnregisterInterfaceEvent() | 1038 |
| 10.146.3.15UnregisterLoggingEvent() | 1039 |
| 10.146.3.16UpdateCameras() | 1039 |
| 10.146.3.17UpdateInterfaceList() | 1039 |
| 10.147SystemPtr Class Reference | 1040 |
| 10.147.1Detailed Description | 1041 |
| 10.147.2Constructor & Destructor Documentation | 1041 |
| 10.147.2.1SystemPtr() [1/4] | 1041 |
| 10.147.2.2SystemPtr() [2/4] | 1041 |
| 10.147.2.3SystemPtr() [3/4] | 1041 |
| 10.147.2.4SystemPtr() [4/4] | 1042 |
| 10.147.2.5~SystemPtr() | 1042 |
| 10.148IFFOption Struct Reference | 1042 |
| 10.148.1Detailed Description | 1043 |

| | |
|--|------|
| 10.148.2Member Enumeration Documentation | 1043 |
| 10.148.2.1CompressionMethod | 1043 |
| 10.148.3Constructor & Destructor Documentation | 1043 |
| 10.148.3.1TIFFOption() | 1043 |
| 10.148.4Member Data Documentation | 1043 |
| 10.148.4.1compression | 1043 |
| 10.148.4.2reserved | 1044 |
| 10.149TransportLayerDevice Class Reference | 1044 |
| 10.149.1Detailed Description | 1046 |
| 10.149.2Constructor & Destructor Documentation | 1046 |
| 10.149.2.1TransportLayerDevice() [1/2] | 1046 |
| 10.149.2.2~TransportLayerDevice() | 1046 |
| 10.149.2.3TransportLayerDevice() [2/2] | 1046 |
| 10.149.3Friends And Related Function Documentation | 1046 |
| 10.149.3.1CameraBase | 1046 |
| 10.149.3.2CameraInternal | 1047 |
| 10.149.3.3CameraBase | 1047 |
| 10.149.4Member Data Documentation | 1047 |
| 10.149.4.1DeviceAccessStatus | 1047 |
| 10.149.4.2DeviceCurrentSpeed | 1047 |
| 10.149.4.3DeviceDisplayName | 1047 |
| 10.149.4.4DeviceDriverVersion | 1047 |
| 10.149.4.5DeviceEndiannessMechanism | 1048 |
| 10.149.4.6DeviceID | 1048 |
| 10.149.4.7DeviceInstanceId | 1048 |
| 10.149.4.8DevicesUpdater | 1048 |
| 10.149.4.9DeviceLinkSpeed | 1048 |
| 10.149.4.10DeviceLocation | 1048 |
| 10.149.4.11DeviceModelName | 1049 |
| 10.149.4.12DeviceMulticastMonitorMode | 1049 |

| | | |
|-------------|---|------|
| 10.149.4.13 | DeviceSerialNumber | 1049 |
| 10.149.4.14 | DeviceType | 1049 |
| 10.149.4.15 | DeviceU3VProtocol | 1049 |
| 10.149.4.16 | DeviceUserID | 1049 |
| 10.149.4.17 | DeviceVendorName | 1050 |
| 10.149.4.18 | DeviceVersion | 1050 |
| 10.149.4.19 | GenICamXMLLocation | 1050 |
| 10.149.4.20 | GenICamXMLPath | 1050 |
| 10.149.4.21 | DevCCP | 1050 |
| 10.149.4.22 | DevDeviceDiscoverMaximumPacketSize | 1050 |
| 10.149.4.23 | DevDeviceForceIP | 1051 |
| 10.149.4.24 | DevDeviceGateway | 1051 |
| 10.149.4.25 | DevDeviceIPAddress | 1051 |
| 10.149.4.26 | DevDevicesWrongSubnet | 1051 |
| 10.149.4.27 | DevDeviceMACAddress | 1051 |
| 10.149.4.28 | DevDeviceMaximumPacketSize | 1051 |
| 10.149.4.29 | DevDeviceMaximumRetryCount | 1052 |
| 10.149.4.30 | DevDeviceModelsBigEndian | 1052 |
| 10.149.4.31 | DevDevicePort | 1052 |
| 10.149.4.32 | DevDeviceReadAndWriteTimeout | 1052 |
| 10.149.4.33 | DevDeviceSubnetMask | 1052 |
| 10.149.4.34 | DevVersionMajor | 1052 |
| 10.149.4.35 | DevVersionMinor | 1053 |
| 10.149.4.36 | UIXMLLocation | 1053 |
| 10.149.4.37 | UIXMLPath | 1053 |
| 10.150 | TransportLayerInterface Class Reference | 1053 |
| 10.150.1 | Detailed Description | 1055 |
| 10.150.2 | Constructor & Destructor Documentation | 1056 |
| 10.150.2.1 | TransportLayerInterface() [1/2] | 1056 |
| 10.150.2.2 | ~TransportLayerInterface() | 1056 |

| | |
|--|------|
| 10.150.2.3TransportLayerInterface() [2/2] | 1056 |
| 10.150.3Friends And Related Function Documentation | 1056 |
| 10.150.3.1Interface | 1056 |
| 10.150.3.2Interface | 1056 |
| 10.150.3.3InterfaceInternal | 1056 |
| 10.150.4Member Data Documentation | 1056 |
| 10.150.4.1ActionCommand | 1057 |
| 10.150.4.2AutoForceIP | 1057 |
| 10.150.4.3DeviceAccessStatus | 1057 |
| 10.150.4.4DeviceCount | 1057 |
| 10.150.4.5DeviceID | 1057 |
| 10.150.4.6DeviceModelName | 1057 |
| 10.150.4.7DeviceSelector | 1058 |
| 10.150.4.8DeviceUnlock | 1058 |
| 10.150.4.9DeviceUpdateList | 1058 |
| 10.150.4.10DeviceVendorName | 1058 |
| 10.150.4.11FilterDriverStatus | 1058 |
| 10.150.4.12DevActionDeviceKey | 1058 |
| 10.150.4.13DevActionGroupKey | 1059 |
| 10.150.4.14DevActionGroupMask | 1059 |
| 10.150.4.15DevActionTime | 1059 |
| 10.150.4.16DevDeviceIPAddress | 1059 |
| 10.150.4.17DevDeviceMACAddress | 1059 |
| 10.150.4.18DevDeviceSubnetMask | 1059 |
| 10.150.4.19DevInterfaceGateway | 1060 |
| 10.150.4.20DevInterfaceIPAddress | 1060 |
| 10.150.4.21DevInterfaceMACAddress | 1060 |
| 10.150.4.22DevInterfaceMTU | 1060 |
| 10.150.4.23DevInterfaceReceiveLinkSpeed | 1060 |
| 10.150.4.24DevInterfaceSubnetMask | 1060 |

| | | |
|-------------|--|------|
| 10.150.4.25 | GevInterfaceTransmitLinkSpeed | 1061 |
| 10.150.4.26 | HostAdapterDriverVersion | 1061 |
| 10.150.4.27 | HostAdapterName | 1061 |
| 10.150.4.28 | HostAdapterVendor | 1061 |
| 10.150.4.29 | IncompatibleDeviceCount | 1061 |
| 10.150.4.30 | IncompatibleDeviceID | 1061 |
| 10.150.4.31 | IncompatibleDeviceModelName | 1062 |
| 10.150.4.32 | IncompatibleDeviceSelector | 1062 |
| 10.150.4.33 | IncompatibleDeviceVendorName | 1062 |
| 10.150.4.34 | IncompatibleGevDeviceIPAddress | 1062 |
| 10.150.4.35 | IncompatibleGevDeviceMACAddress | 1062 |
| 10.150.4.36 | IncompatibleGevDeviceSubnetMask | 1062 |
| 10.150.4.37 | InterfaceDisplayName | 1063 |
| 10.150.4.38 | InterfaceID | 1063 |
| 10.150.4.39 | InterfaceType | 1063 |
| 10.150.4.40 | IOEStatus | 1063 |
| 10.151 | TransportLayerStream Class Reference | 1063 |
| 10.151.1 | Detailed Description | 1065 |
| 10.151.2 | Constructor & Destructor Documentation | 1065 |
| 10.151.2.1 | TransportLayerStream() [1/2] | 1065 |
| 10.151.2.2 | ~TransportLayerStream() | 1065 |
| 10.151.2.3 | TransportLayerStream() [2/2] | 1065 |
| 10.151.3 | Friends And Related Function Documentation | 1065 |
| 10.151.3.1 | CameraBase | 1065 |
| 10.151.3.2 | CameraInternal | 1066 |
| 10.151.3.3 | CameraBase | 1066 |
| 10.151.4 | Member Data Documentation | 1066 |
| 10.151.4.1 | GevFailedPacketCount | 1066 |
| 10.151.4.2 | GevMaximumNumberResendBuffers | 1066 |
| 10.151.4.3 | GevMaximumNumberResendRequests | 1066 |

| | | |
|-------------|--|------|
| 10.151.4.4 | GevPacketResendMode | 1066 |
| 10.151.4.5 | GevPacketResendTimeout | 1067 |
| 10.151.4.6 | GevResendPacketCount | 1067 |
| 10.151.4.7 | GevResendRequestCount | 1067 |
| 10.151.4.8 | GevTotalPacketCount | 1067 |
| 10.151.4.9 | StreamBlockTransferSize | 1067 |
| 10.151.4.10 | StreamBufferCountManual | 1067 |
| 10.151.4.11 | StreamBufferCountMax | 1068 |
| 10.151.4.12 | StreamBufferCountMode | 1068 |
| 10.151.4.13 | StreamBufferCountResult | 1068 |
| 10.151.4.14 | StreamBufferHandlingMode | 1068 |
| 10.151.4.15 | StreamBufferUnderrunCount | 1068 |
| 10.151.4.16 | StreamCRCCheckEnable | 1068 |
| 10.151.4.17 | StreamDefaultBufferCount | 1069 |
| 10.151.4.18 | StreamDefaultBufferCountMax | 1069 |
| 10.151.4.19 | StreamDefaultBufferCountMode | 1069 |
| 10.151.4.20 | StreamFailedBufferCount | 1069 |
| 10.151.4.21 | StreamID | 1069 |
| 10.151.4.22 | StreamTotalBufferCount | 1069 |
| 10.151.4.23 | StreamType | 1070 |
| 10.152 | TransportLayerSystem Class Reference | 1070 |
| 10.152.1 | Detailed Description | 1070 |
| 10.152.2 | Constructor & Destructor Documentation | 1070 |
| 10.152.2.1 | TransportLayerSystem() [1/2] | 1071 |
| 10.152.2.2 | ~TransportLayerSystem() | 1071 |
| 10.152.2.3 | TransportLayerSystem() [2/2] | 1071 |
| 10.152.3 | Friends And Related Function Documentation | 1071 |
| 10.152.3.1 | ISystem | 1071 |
| 10.152.3.2 | System | 1071 |
| 10.152.3.3 | SystemPtrInternal | 1071 |

| | |
|--|------|
| 10.152.4Member Data Documentation | 1071 |
| 10.152.4.1EnumerateGEVInterfaces | 1072 |
| 10.1533V_CHUNK_TRAILER Struct Reference | 1072 |
| 10.153.1Detailed Description | 1072 |
| 10.153.2Member Data Documentation | 1072 |
| 10.153.2.1ChunkID | 1072 |
| 10.153.2.2ChunkLength | 1072 |
| 10.1543V_COMMAND_HEADER Struct Reference | 1073 |
| 10.154.1Detailed Description | 1073 |
| 10.154.2Member Data Documentation | 1073 |
| 10.154.2.1CommandId | 1073 |
| 10.154.2.2Flags | 1073 |
| 10.154.2.3Length | 1073 |
| 10.154.2.4Prefix | 1073 |
| 10.154.2.5ReqId | 1074 |
| 10.1553V_EVENT_DATA Struct Reference | 1074 |
| 10.155.1Detailed Description | 1074 |
| 10.155.2Member Data Documentation | 1074 |
| 10.155.2.1EventId | 1074 |
| 10.155.2.2Reserved | 1074 |
| 10.155.2.3Timestamp | 1074 |
| 10.1563V_EVENT_MESSAGE Struct Reference | 1075 |
| 10.156.1Detailed Description | 1075 |
| 10.156.2Member Data Documentation | 1075 |
| 10.156.2.1CommandHeader | 1075 |
| 10.156.2.2EventData | 1075 |
| 10.157ValueNode Class Reference | 1076 |
| 10.157.1Detailed Description | 1077 |
| 10.157.2Constructor & Destructor Documentation | 1077 |
| 10.157.2.1ValueNode() [1/2] | 1077 |

| | |
|--|-------------|
| 10.157.2.2ValueNode() [2/2] | 1077 |
| 10.157.2.3~ValueNode() | 1077 |
| 10.157.3Member Function Documentation | 1078 |
| 10.157.3.1FromString() | 1078 |
| 10.157.3.2GetNode() | 1079 |
| 10.157.3.3sValueCacheValid() | 1079 |
| 10.157.3.4SetReference() | 1079 |
| 10.157.3.5ToString() | 1079 |
| 10.158Version_t Struct Reference | 1080 |
| 10.158.1Detailed Description | 1080 |
| 10.158.2Member Data Documentation | 1080 |
| 10.158.2.1Major | 1080 |
| 10.158.2.2Minor | 1080 |
| 10.158.2.3SubMinor | 1080 |
| 11 File Documentation | 1081 |
| 11.1 doc/Doxygen/spindocs/Licensing.dox File Reference | 1081 |
| 11.2 doc/Doxygen/spindocs/MainPage.dox File Reference | 1081 |
| 11.3 include/ArrivalEvent.h File Reference | 1081 |
| 11.4 include/AVIRecorder.h File Reference | 1083 |
| 11.5 include/BasePtr.h File Reference | 1083 |
| 11.6 include/Camera.h File Reference | 1085 |
| 11.7 include/CameraBase.h File Reference | 1087 |
| 11.8 include/CameraDefs.h File Reference | 1089 |
| 11.9 include/CameraList.h File Reference | 1122 |
| 11.10include/CameraPtr.h File Reference | 1124 |
| 11.11include/ChunkData.h File Reference | 1126 |
| 11.12include/DeviceEvent.h File Reference | 1128 |
| 11.13include/Event.h File Reference | 1130 |
| 11.14include/Exception.h File Reference | 1132 |
| 11.15include/Image.h File Reference | 1133 |

| | |
|--|------|
| 11.16include/ImageEvent.h File Reference | 1135 |
| 11.17include/ImagePtr.h File Reference | 1136 |
| 11.18include/ImageStatistics.h File Reference | 1138 |
| 11.19include/ImageUtility.h File Reference | 1140 |
| 11.20include/ImageUtilityHeatmap.h File Reference | 1140 |
| 11.21include/ImageUtilityPolarization.h File Reference | 1141 |
| 11.22include/Interface.h File Reference | 1141 |
| 11.23include/Interface/IArrivalEvent.h File Reference | 1143 |
| 11.24include/Interface/ICameraBase.h File Reference | 1145 |
| 11.25include/Interface/ICameraList.h File Reference | 1147 |
| 11.26include/Interface/IChunkData.h File Reference | 1149 |
| 11.27include/Interface/IDeviceEvent.h File Reference | 1151 |
| 11.28include/Interface/IImage.h File Reference | 1153 |
| 11.29include/Interface/IImageEvent.h File Reference | 1155 |
| 11.30include/Interface/IImageStatistics.h File Reference | 1156 |
| 11.31include/Interface/IInterface.h File Reference | 1158 |
| 11.32include/Interface/IInterfaceEvent.h File Reference | 1160 |
| 11.33include/Interface/IInterfaceList.h File Reference | 1162 |
| 11.34include/Interface/ILoggingEvent.h File Reference | 1163 |
| 11.35include/Interface/IRemovalEvent.h File Reference | 1165 |
| 11.36include/Interface/IStream.h File Reference | 1167 |
| 11.37include/Interface/ISystem.h File Reference | 1167 |
| 11.38include/InterfaceEvent.h File Reference | 1169 |
| 11.39include/InterfaceList.h File Reference | 1171 |
| 11.40include/InterfacePtr.h File Reference | 1172 |
| 11.41include/LoggingEvent.h File Reference | 1174 |
| 11.42include/LoggingEventData.h File Reference | 1175 |
| 11.43include/LoggingEventDataPtr.h File Reference | 1177 |
| 11.44include/RemovalEvent.h File Reference | 1179 |
| 11.45include/SpinGenApi/Autovector.h File Reference | 1181 |

| | |
|--|------|
| 11.46include/SpinGenApi/Base.h File Reference | 1182 |
| 11.47include/SpinGenApi/BooleanNode.h File Reference | 1183 |
| 11.48include/SpinGenApi/CategoryNode.h File Reference | 1185 |
| 11.49include/SpinGenApi/ChunkAdapter.h File Reference | 1187 |
| 11.50include/SpinGenApi/ChunkAdapterDcam.h File Reference | 1189 |
| 11.51include/SpinGenApi/ChunkAdapterGeneric.h File Reference | 1191 |
| 11.52include/SpinGenApi/ChunkAdapterGEV.h File Reference | 1193 |
| 11.53include/SpinGenApi/ChunkAdapterU3V.h File Reference | 1195 |
| 11.54include/SpinGenApi/ChunkPort.h File Reference | 1197 |
| 11.55include/SpinGenApi/CommandNode.h File Reference | 1199 |
| 11.56include/SpinGenApi/Compatibility.h File Reference | 1202 |
| 11.56.1 Macro Definition Documentation | 1203 |
| 11.56.1.1 FMT_I64 | 1203 |
| 11.57include/SpinGenApi/Container.h File Reference | 1203 |
| 11.58include/SpinGenApi/Counter.h File Reference | 1203 |
| 11.59include/SpinGenApi/EnumClasses.h File Reference | 1204 |
| 11.60include/SpinGenApi/EnumEntryNode.h File Reference | 1206 |
| 11.61include/SpinGenApi/EnumNode.h File Reference | 1208 |
| 11.62include/SpinGenApi/EnumNodeT.h File Reference | 1210 |
| 11.63include/SpinGenApi/EventAdapter.h File Reference | 1212 |
| 11.64include/SpinGenApi/EventAdapter1394.h File Reference | 1214 |
| 11.65include/SpinGenApi/EventAdapterGeneric.h File Reference | 1216 |
| 11.66include/SpinGenApi/EventAdapterGEV.h File Reference | 1218 |
| 11.67include/SpinGenApi/EventAdapterU3V.h File Reference | 1220 |
| 11.68include/SpinGenApi/EventPort.h File Reference | 1222 |
| 11.69include/SpinGenApi/Filestream.h File Reference | 1224 |
| 11.70include/SpinGenApi/FloatNode.h File Reference | 1226 |
| 11.71include/SpinGenApi/FloatRegNode.h File Reference | 1228 |
| 11.72include/SpinGenApi/GCBase.h File Reference | 1230 |
| 11.73include/SpinGenApi/GCString.h File Reference | 1231 |

| | |
|---|------|
| 11.73.1 Macro Definition Documentation | 1232 |
| 11.73.1.1 GCSTRING_NPOS | 1232 |
| 11.73.2 Function Documentation | 1232 |
| 11.73.2.1 operator<<() | 1233 |
| 11.73.2.2 operator>>() | 1233 |
| 11.74include/SpinGenApi/GCStringVector.h File Reference | 1233 |
| 11.75include/SpinGenApi/GCSynch.h File Reference | 1234 |
| 11.76include/SpinGenApi/GCTypes.h File Reference | 1235 |
| 11.76.1 Macro Definition Documentation | 1236 |
| 11.76.1.1 __STDC_CONSTANT_MACROS | 1236 |
| 11.76.1.2 __STDC_LIMIT_MACROS | 1236 |
| 11.76.1.3 GC_INT32_MAX | 1236 |
| 11.76.1.4 GC_INT32_MIN | 1236 |
| 11.76.1.5 GC_INT64_MAX | 1236 |
| 11.76.1.6 GC_INT64_MIN | 1237 |
| 11.76.1.7 GC_INT8_MAX | 1237 |
| 11.76.1.8 GC_INT8_MIN | 1237 |
| 11.76.1.9 GC_UINT32_MAX | 1237 |
| 11.76.1.10GC_UINT64_MAX | 1237 |
| 11.76.1.11GC_UINT8_MAX | 1237 |
| 11.77include/SpinGenApi/GCUtilities.h File Reference | 1238 |
| 11.77.1 Macro Definition Documentation | 1241 |
| 11.77.1.1 __ERR__ | 1241 |
| 11.77.1.2 __LINE_STR__ | 1241 |
| 11.77.1.3 __LOCATION__ | 1241 |
| 11.77.1.4 __OUTPUT_FORMATER__ | 1241 |
| 11.77.1.5 __TODO__ | 1241 |
| 11.77.1.6 __WARN__ | 1241 |
| 11.77.1.7 _TO_STRING | 1242 |
| 11.77.1.8 EXPAND_TO_STRINGISE | 1242 |

| | |
|--|------|
| 11.77.1.9 GC_COUNTOF | 1242 |
| 11.77.1.10 GENICAM_DEPRECATED | 1242 |
| 11.77.1.11 GENICAM_UNUSED | 1242 |
| 11.77.1.12 USE_TEMP_CACHE_FILE [1/2] | 1242 |
| 11.77.1.13 USE_TEMP_CACHE_FILE [2/2] | 1242 |
| 11.78 include/SpinGenApi/IBoolean.h File Reference | 1243 |
| 11.79 include/SpinGenApi/ICategory.h File Reference | 1245 |
| 11.80 include/SpinGenApi/IChunkPort.h File Reference | 1247 |
| 11.81 include/SpinGenApi/ICommand.h File Reference | 1249 |
| 11.82 include/SpinGenApi/IDestroy.h File Reference | 1251 |
| 11.83 include/SpinGenApi/IDeviceInfo.h File Reference | 1252 |
| 11.84 include/SpinGenApi/IEnumEntry.h File Reference | 1254 |
| 11.85 include/SpinGenApi/IEnumeration.h File Reference | 1256 |
| 11.86 include/SpinGenApi/IEnumerationT.h File Reference | 1257 |
| 11.87 include/SpinGenApi/IFloat.h File Reference | 1259 |
| 11.88 include/SpinGenApi/IInteger.h File Reference | 1261 |
| 11.89 include/SpinGenApi/INode.h File Reference | 1263 |
| 11.90 include/SpinGenApi/INodeMap.h File Reference | 1266 |
| 11.91 include/SpinGenApi/INodeMapDyn.h File Reference | 1267 |
| 11.92 include/SpinGenApi/IntegerNode.h File Reference | 1269 |
| 11.93 include/SpinGenApi/IntRegNode.h File Reference | 1271 |
| 11.94 include/SpinGenApi/IPort.h File Reference | 1273 |
| 11.95 include/SpinGenApi/IPortConstruct.h File Reference | 1274 |
| 11.96 include/SpinGenApi/IPortRecorder.h File Reference | 1276 |
| 11.97 include/SpinGenApi/IRegister.h File Reference | 1278 |
| 11.98 include/SpinGenApi/ISelector.h File Reference | 1280 |
| 11.99 include/SpinGenApi/ISelectorDigit.h File Reference | 1281 |
| 11.100 include/SpinGenApi/IString.h File Reference | 1283 |
| 11.101 include/SpinGenApi/IValue.h File Reference | 1285 |
| 11.102 include/SpinGenApi/Node.h File Reference | 1286 |

| | | |
|------------|--|------|
| 11.103 | include/SpinGenApi/NodeCallback.h File Reference | 1288 |
| 11.104 | include/SpinGenApi/NodeCallbackImpl.h File Reference | 1290 |
| 11.105 | include/SpinGenApi/NodeMap.h File Reference | 1291 |
| 11.106 | include/SpinGenApi/NodeMapFactory.h File Reference | 1293 |
| 11.107 | include/SpinGenApi/NodeMapRef.h File Reference | 1294 |
| 11.108 | include/SpinGenApi/Persistence.h File Reference | 1295 |
| 11.109 | include/SpinGenApi/Pointer.h File Reference | 1297 |
| 11.110 | include/SpinGenApi/PortImpl.h File Reference | 1300 |
| 11.111 | include/SpinGenApi/PortNode.h File Reference | 1301 |
| 11.112 | include/SpinGenApi/PortRecorder.h File Reference | 1303 |
| 11.113 | include/SpinGenApi/PortReplay.h File Reference | 1304 |
| 11.114 | include/SpinGenApi/PortWriteList.h File Reference | 1305 |
| 11.115 | include/SpinGenApi/Reference.h File Reference | 1307 |
| 11.116 | include/SpinGenApi/RegisterNode.h File Reference | 1308 |
| 11.117 | include/SpinGenApi/RegisterPortImpl.h File Reference | 1310 |
| 11.118 | include/SpinGenApi/SelectorSet.h File Reference | 1310 |
| 11.119 | include/SpinGenApi/SpinnakerGenApi.h File Reference | 1311 |
| 11.120 | include/SpinGenApi/SpinTestCamera.h File Reference | 1313 |
| 11.121 | include/SpinGenApi/StringNode.h File Reference | 1313 |
| 11.122 | include/SpinGenApi/StringRegNode.h File Reference | 1315 |
| 11.123 | include/SpinGenApi/StructPort.h File Reference | 1317 |
| 11.124 | include/SpinGenApi/Synch.h File Reference | 1317 |
| 11.125 | include/SpinGenApi/Types.h File Reference | 1318 |
| 11.125.1 | Macro Definition Documentation | 1321 |
| 11.125.1.1 | interface | 1321 |
| 11.126 | include/SpinGenApi/ValueNode.h File Reference | 1322 |
| 11.127 | include/Spinnaker.h File Reference | 1324 |
| 11.128 | include/SpinnakerDefs.h File Reference | 1325 |
| 11.129 | include/SpinnakerPlatform.h File Reference | 1329 |
| 11.130 | include/SpinUpdate.h File Reference | 1329 |

| | | |
|--------------|--|-------------|
| 11.130.1 | Macro Definition Documentation | 1330 |
| 11.130.1.1 | SPINUPDATE_API | 1330 |
| 11.130.2 | Function Documentation | 1330 |
| 11.130.2.1 | GetErrorMessage() | 1330 |
| 11.130.2.2 | SetMessageCallback() | 1330 |
| 11.130.2.3 | SetProgressCallback() | 1330 |
| 11.130.2.4 | UpdateFirmware() | 1331 |
| 11.130.2.5 | UpdateFirmwareConsole() | 1331 |
| 11.130.3 | variable Documentation | 1331 |
| 11.130.3.1 | UpdaterMessageCallback | 1331 |
| 11.130.3.2 | UpdaterProgressCallback | 1331 |
| 11.131 | include/SpinVideo.h File Reference | 1332 |
| 11.132 | include/SpinVideoDefs.h File Reference | 1332 |
| 11.133 | include/System.h File Reference | 1333 |
| 11.133.1 | Macro Definition Documentation | 1334 |
| 11.133.1.1 | FLIR_SPINNAKER_VERSION_BUILD | 1335 |
| 11.133.1.2 | FLIR_SPINNAKER_VERSION_MAJOR | 1335 |
| 11.133.1.3 | FLIR_SPINNAKER_VERSION_MINOR | 1335 |
| 11.133.1.4 | FLIR_SPINNAKER_VERSION_TYPE | 1335 |
| 11.134 | include/SystemPtr.h File Reference | 1335 |
| 11.135 | include/TransportLayerDefs.h File Reference | 1337 |
| 11.136 | include/TransportLayerDevice.h File Reference | 1339 |
| 11.137 | include/TransportLayerInterface.h File Reference | 1341 |
| 11.138 | include/TransportLayerStream.h File Reference | 1343 |
| 11.139 | include/TransportLayerSystem.h File Reference | 1345 |
| Index | | 1347 |

Chapter 1

Introduction

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

Chapter 2

Software Licensing Information

Table 2.1 License table

| Component | License |
|--|---|
| Spinnaker | Copyright © 2017 FLIR Integrated Imaging Solutions, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. |
| GenICam | GenICam License |
| AdapterManager | The Code Project Open License (CPO-OL) |
| Make ListView.ScrollIntoView Scroll the Item into the Center of the ListView | WP:CC-BY-SA License |
| Work with Bitmaps Faster in C# | The Code Project Open License (CPO-OL) 1.02 |
| FreeImage | FreeImage public license |
| Boost | Boost Software License |
| Libusb | LGPLv2.1 License |
| Libraw1394 | LGPLv2.0 License |
| FFMPEG | LGPLv2.1 License |
| log4Net | Apache license 2.0 |
| log4Cpp | LGPL License |

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

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

| | |
|--|-----|
| Spinnaker Event Classes | 27 |
| ArrivalEvent Class | 29 |
| DeviceEvent Class | 158 |
| Event Class | 159 |
| ImageEvent Class | 162 |
| InterfaceEvent Class | 169 |
| LoggingEvent Class | 172 |
| Logging Event Class | 173 |
| LoggingEventDataPtr Class | 174 |
| RemovalEvent Class | 175 |
| Spinnaker Classes | 30 |
| AVI Recorder Class | 33 |
| BasePtr Class | 36 |
| Camera Class | 37 |
| Camera Base Class | 38 |
| CameraDefs Class | 39 |
| Camera List Class | 154 |
| CameraPtr Class | 155 |
| ChunkData Class | 157 |
| Exception Class | 160 |
| Image Class | 161 |
| ImagePtr Class | 163 |
| ImageStatistics Class | 164 |
| Image Utility Class | 165 |
| Image Utility Heatmap Class | 166 |
| Image Utility Polarization Class | 167 |
| Interface Class | 168 |
| InterfaceList Class | 170 |
| InterfacePtr Class | 171 |
| Spinnaker Video Class | 191 |
| System Class | 193 |
| SystemPtr Class | 194 |
| Camera Base Interface Class | 207 |
| IChunkData Class | 208 |
| IImage Class | 209 |

| | |
|---|-----|
| ImageStatistics Class | 210 |
| IInterface Class | 211 |
| IInterfaceList Class | 212 |
| ISystem Class | 213 |
| Spinnaker Headers | 176 |
| Spinnaker.h | 178 |
| Spinnaker Definitions | 179 |
| Spinnaker Platform | 190 |
| Spinnaker Video Definitions | 192 |
| Spinnaker QuickSpin Classes | 195 |
| TransportLayerDefs Class | 196 |
| TransportLayerDevice Class | 203 |
| TransportLayerInterface Class | 204 |
| TransportLayerStream Class | 205 |
| TransportLayerSystem Class | 206 |
| Spinnaker GenApi Classes | 214 |
| AutoVector Class | 225 |
| BooleanNode Class | 229 |
| CategoryNode Class | 230 |
| ChunkAdapter Class | 231 |
| ChunkAdapterDcam Class | 232 |
| ChunkAdapterGeneric Class | 233 |
| ChunkAdapterGEV Class | 234 |
| ChunkPort Class | 235 |
| CommandNode Class | 236 |
| Container Class | 237 |
| Counter Class | 238 |
| EnumClasses Class | 239 |
| EnumEntryNode Class | 241 |
| EnumNode Class | 242 |
| EnumNodeT Class | 243 |
| EventAdapter Class | 244 |
| EventAdapter1394 Class | 245 |
| EventAdapterGeneric Class | 246 |
| EventAdapterGEV Class | 247 |
| EventAdapterU3V Class | 248 |
| EventPort Class | 249 |
| Filestream Class | 250 |
| FloatNode Class | 251 |
| FloatRegNode Class | 252 |
| GCString Class | 253 |
| GCSynch Class | 254 |
| GCTypes Class | 255 |
| IntegerNode Class | 306 |
| IntRegNode Class | 307 |
| IString Class | 319 |
| IValue Class | 320 |
| Node Class | 322 |
| NodeCallback Class | 323 |
| NodeMap Class | 326 |
| NodeMapFactory Class | 327 |
| NodeMapRef Class | 329 |
| Persistence Class | 330 |
| Pointer Class | 331 |
| PortImpl Class | 337 |
| PortNode Class | 338 |
| PortRecorder Class | 339 |

| | |
|---------------------------------------|-----|
| PortReplay Class | 340 |
| PortWriteList Class | 341 |
| RegisterNode Class | 343 |
| RegisterPortImpl Class | 344 |
| SelectorSet Class | 345 |
| SpinTestCamera Class | 346 |
| StringNode Class | 347 |
| StringRegNode Class | 348 |
| StructPort Class | 349 |
| Synch Class | 350 |
| ValueNode Class | 362 |
| ChunkAdapterU3V Class | 363 |
| IPortRecorder Interface | 311 |
| Spinnaker GenApi Interfaces | 226 |
| IBase Interface | 228 |
| IBoolean Interface | 263 |
| ICategory Interfaces | 265 |
| IChunkPort Interface | 266 |
| ICommand Interface | 268 |
| IDestroy Interface | 270 |
| IDeviceInfo Interface | 271 |
| IEnumEntry Interface | 274 |
| IEnumeration Interface | 276 |
| IEnumerationT Interface | 279 |
| IFloat Interface | 281 |
| IInteger Interface | 285 |
| INode Interface | 287 |
| INodeMap Interface | 298 |
| INodeMapDyn Interface | 301 |
| IPort Interface | 308 |
| IPortConstruct Interface | 310 |
| IPortRecorder Interface | 311 |
| IRegister Interfaces | 313 |
| ISelector Interface | 315 |
| ISelectorDigit Interface | 316 |
| Reference Interfaces | 342 |
| Spinnaker GenApi Utilities | 256 |
| GCUilities Utility | 257 |
| Spinnaker GenApi Enums | 351 |
| Types Enums | 352 |

Chapter 4

Namespace Index

4.1 Namespace List

Here is a list of all namespaces with brief descriptions:

| | |
|------------------------------------|-----|
| Spinnaker | 365 |
| Spinnaker::GenApi | 404 |
| Spinnaker::GenICam | 421 |
| Spinnaker::Video | 423 |

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

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

| | |
|--|------|
| ActionCommandResult | 425 |
| AttachStatistics_t | 428 |
| AutoLock | 429 |
| AutoLock | 429 |
| AVIOption | 430 |
| BasePtr< T, B > | 431 |
| BasePtr< Camera, ICameraBase > | 431 |
| CameraPtr | 612 |
| BasePtr< IImage > | 431 |
| ImagePtr | 896 |
| BasePtr< IInterface > | 431 |
| InterfacePtr | 941 |
| BasePtr< ISystem > | 431 |
| SystemPtr | 1040 |
| BasePtr< LoggingEventData > | 431 |
| LoggingEventDataPtr | 966 |
| basic_istream | |
| IDevFileStreamBase< CharType, Traits > | 830 |
| basic_ostream | |
| ODevFileStreamBase< CharType, Traits > | 992 |
| basic_streambuf | |
| IDevFileStreamBuf< CharType, Traits > | 832 |
| ODevFileStreamBuf< CharType, Traits > | 994 |
| BMPOption | 435 |
| CChunkAdapter | 615 |
| CChunkAdapterDcam | 619 |
| CChunkAdapterGeneric | 621 |
| CChunkAdapterGEV | 624 |
| CChunkAdapterU3V | 626 |
| CDataStruct | |
| CTestPortStruct< CDataStruct > | 725 |
| CEventAdapter | 638 |
| CEventAdapter1394 | 640 |

| | |
|--|------|
| CEventAdapterGeneric | 642 |
| CEventAdapterGEV | 645 |
| CEventAdapterU3V | 647 |
| CGeneric_XMLLoaderParams | 659 |
| CNodeMapRefT< GenApi::CGeneric_XMLLoaderParams > | 697 |
| CNodeMapRef | 694 |
| CGlobalLock | 660 |
| CGlobalLockUnlocker | 662 |
| CLock | 676 |
| CLockEx | 681 |
| CLock | 679 |
| CLockEx | 682 |
| CNodeCallback | 683 |
| Function_NodeCallback< Function > | 780 |
| Member_NodeCallback< Client, Member > | 968 |
| CNodeMapFactory | 686 |
| Counter | 705 |
| CPointer< T, B > | 707 |
| CPointer< IFloat, IBase > | 707 |
| CFloatPtr | 657 |
| CPointer< INode, IBase > | 707 |
| DCAM_CHECKSUM | 729 |
| DCAM_CHUNK_TRAILER | 729 |
| double_autovector_t | 733 |
| EAccessModeClass | 736 |
| ECachingModeClass | 737 |
| EDisplayNotationClass | 738 |
| EEndianessClass | 739 |
| EGenApiSchemaVersionClass | 740 |
| EInputDirectionClass | 741 |
| ENamespaceClass | 742 |
| ERepresentationClass | 751 |
| ESignClass | 752 |
| ESlopeClass | 753 |
| EStandardNameSpaceClass | 754 |
| Event | 756 |
| IArrivalEvent | 803 |
| ArrivalEvent | 426 |
| IInterfaceEvent | 858 |
| InterfaceEvent | 935 |
| IDeviceEvent | 834 |
| DeviceEvent | 730 |
| IImageEvent | 848 |
| ImageEvent | 893 |
| ILoggingEvent | 863 |
| LoggingEvent | 960 |
| IRemovalEvent | 946 |
| IInterfaceEvent | 858 |
| RemovalEvent | 1015 |
| EVisibilityClass | 759 |
| exception | |
| Exception | 760 |
| EYesNoClass | 765 |
| FileProtocolAdapter | 766 |
| gcstring | 782 |
| GVCP_CHUNK_TRAILER | 791 |

| | |
|--|------|
| GVCP_EVENT_ITEM | 792 |
| GVCP_EVENT_ITEM_BASIC | 793 |
| GVCP_EVENT_ITEM_EXTENDED_ID | 794 |
| GVCP_EVENT_REQUEST | 795 |
| GVCP_EVENT_REQUEST_EXTENDED_ID | 796 |
| GVCP_EVENTDATA_REQUEST | 797 |
| GVCP_EVENTDATA_REQUEST_EXTENDED_ID | 798 |
| GVCP_REQUEST_HEADER | 799 |
| H264Option | 800 |
| IBoolean | |
| BooleanNode | 437 |
| ICameraBase | 805 |
| CameraBase | 591 |
| Camera | 440 |
| ICameraList | 814 |
| CameraList | 606 |
| ICategory | |
| CategoryNode | 613 |
| IChunkData | 817 |
| ChunkData | 664 |
| IChunkPort | |
| PortNode | 999 |
| PortReplay | 1007 |
| PortRecorder | 1004 |
| ICommand | |
| CommandNode | 702 |
| IDataStream | 825 |
| IDeviceInfo | |
| NodeMap | 982 |
| SpinTestCamera | 1019 |
| IEnumEntry | |
| EnumEntryNode | 743 |
| IEnumeration | |
| EnumNode | 746 |
| CEnumerationTRef< EnumT > | 633 |
| IEnumerationT | |
| CEnumerationTRef< EnumT > | 633 |
| IFloat | |
| FloatNode | 770 |
| FloatRegNode | 777 |
| IImage | 837 |
| Image | 865 |
| IImageStatistics | 850 |
| ImageStatistics | 898 |
| IInteger | |
| IntegerNode | 924 |
| IntRegNode | 943 |
| IInterface | 854 |
| Interface | 930 |
| IInterfaceList | 861 |
| InterfaceList | 937 |
| ImageUtility | 905 |
| ImageUtilityHeatmap | 909 |
| ImageUtilityPolarization | 913 |
| INode | |

| | |
|---|------|
| Node | 971 |
| CSelectorSet | 722 |
| PortNode | 999 |
| ValueNode | 1076 |
| BooleanNode | 437 |
| CategoryNode | 613 |
| CommandNode | 702 |
| EnumEntryNode | 743 |
| EnumNode | 746 |
| FloatNode | 770 |
| IntegerNode | 924 |
| RegisterNode | 1011 |
| FloatRegNode | 777 |
| IntRegNode | 943 |
| StringRegNode | 1028 |
| StringNode | 1024 |
| StringRegNode | 1028 |
| INodeMap | |
| NodeMap | 982 |
| int64_autovector_t | 921 |
| IPersistScript | |
| CFeatureBag | 654 |
| IPortConstruct | |
| CChunkPort | 628 |
| CEventPort | 649 |
| CPortImpl | 712 |
| CRegisterPortImpl | 719 |
| CTestPortStruct< CDataStruct > | 725 |
| PortNode | 999 |
| IPortRecorder | |
| PortNode | 999 |
| PortRecorder | 1004 |
| IPortReplay | |
| CPortImpl | 712 |
| PortReplay | 1007 |
| IPortWriteList | |
| CPortWriteList | 716 |
| IRegister | |
| RegisterNode | 1011 |
| IString | |
| StringNode | 1024 |
| ISystem | 948 |
| System | 1031 |
| IValue | |
| ValueNode | 1076 |
| JPEGOption | 953 |
| JPG2Option | 955 |
| LibraryVersion | 956 |
| LockableObject< Object >::Lock | 957 |
| LockableObject< Object > | 958 |
| LoggingEventData | 962 |
| MJPGOption | 970 |
| CNodeMapFactory::NodeStatistics_t | 991 |
| PGMOption | 996 |
| PNGOption | 997 |
| PPMOption | 1009 |
| SingleChunkData_t | 1017 |
| SingleChunkDataStr_t | 1018 |

| | |
|---|------|
| SpinVideo | 1019 |
| TIFFOption | 1042 |
| TransportLayerDevice | 1044 |
| TransportLayerInterface | 1053 |
| TransportLayerStream | 1063 |
| TransportLayerSystem | 1070 |
| U3V_CHUNK_TRAILER | 1072 |
| U3V_COMMAND_HEADER | 1073 |
| U3V_EVENT_DATA | 1074 |
| U3V_EVENT_MESSAGE | 1075 |
| Version_t | 1080 |
| TCameraParams | |
| CNodeMapRefT< TCameraParams > | 697 |

Chapter 6

Class Index

6.1 Class List

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

| | |
|---|-----|
| ActionCommandResult | |
| Action Command Result | 425 |
| ArrivalEvent | |
| An event handler for capturing the device arrival event | 426 |
| AttachStatistics_t | |
| Delivers information about the attached chunks and nodes | 428 |
| AutoLock | 429 |
| AutoLock | 429 |
| AVIOption | |
| Options for saving AVI files | 430 |
| BasePtr< T, B > | |
| The base class of the SystemPtr , CameraPtr , InterfacePtr , ImagePtr and LoggingEventDataPtr objects | 431 |
| BMPOption | |
| Options for saving Bitmap image | 435 |
| BooleanNode | |
| Interface for string properties | 437 |
| Camera | |
| The camera object class | 440 |
| CameraBase | |
| The base class for the camera object | 591 |
| CameraList | |
| Used to hold a list of camera objects | 606 |
| CameraPtr | |
| A reference tracked pointer to a camera object | 612 |
| CategoryNode | |
| Interface for string properties | 613 |
| CChunkAdapter | |
| Connects a chunked buffer to a node map | 615 |
| CChunkAdapterDcam | |
| Connects a chunked DCAM buffer to a node map | 619 |
| CChunkAdapterGeneric | 621 |
| CChunkAdapterGEV | |
| Connects a chunked DCAM buffer to a node map | 624 |
| CChunkAdapterU3V | |
| Connects a chunked U3V buffer to a node map | 626 |

| | |
|---|-----|
| CChunkPort | |
| Port attachable to a chunk in a buffer | 628 |
| CEnumerationTRef< EnumT > | |
| Interface for string properties | 633 |
| CEventAdapter | |
| Delivers Events to ports | 638 |
| CEventAdapter1394 | |
| Distribute the events to the node map | 640 |
| CEventAdapterGeneric | |
| Connects a generic event to a node map | 642 |
| CEventAdapterGEV | |
| Connects a GigE Event to a node map | 645 |
| CEventAdapterU3V | |
| Connects a U3V Event to a node map | 647 |
| CEventPort | |
| Port attachable to an event | 649 |
| CFeatureBag | |
| Bag holding streamable features of a nodetree | 654 |
| CFloatPtr | |
| SmartPointer for IFloat interface pointer | 657 |
| CGeneric_XMLLoaderParams | |
| Empty base class used by class CNodeMapRef as generic template argument | 659 |
| CGlobalLock | |
| Named global lock which can be used over process boundaries | 660 |
| CGlobalLockUnlocker | |
| Unlocks the global lock object on destruction | 662 |
| ChunkData | |
| The chunk data which contains additional information about an image | 664 |
| CLock | |
| A lock class | 676 |
| CLock | |
| A lock class | 679 |
| CLockEx | |
| This class is for testing purposes only | 681 |
| CLockEx | |
| This class is for testing purposes only | 682 |
| CNodeCallback | |
| Callback body instance for INode pointers | 683 |
| CNodeMapFactory | |
| The node map factory is used for creating node maps from camera description files | 686 |
| CNodeMapRef | |
| SmartPointer for NodeMaps with create function | 694 |
| CNodeMapRefT< TCameraParams > | |
| SmartPointer template for NodeMaps with create function | 697 |
| CommandNode | |
| Interface for string properties | 702 |
| Counter | |
| Definition of a simple Counter class | 705 |
| CPointer< T, B > | |
| Encapsulates a GenApi pointer dealing with the dynamic_cast automatically | 707 |
| CPortImpl | |
| Standard implementation for a port | 712 |
| CPortWriteList | |
| Container holding a list of port write commands | 716 |
| CRegisterPortImpl | |
| Standard implementation for a port using a register based transport layer | 719 |
| CSelectorSet | |
| The set of selectors selecting a given node | 722 |

| | |
|--|-----|
| CTestPortStruct< CDataStruct > | |
| Implements a register spaces based on a C++ struct | 725 |
| DCAM_CHECKSUM | 729 |
| DCAM_CHUNK_TRAILER | 729 |
| DeviceEvent | |
| A handler to device events | 730 |
| double_autovector_t | |
| Vector of doubles with reference counting | 733 |
| EAccessModeClass | |
| Holds conversion methods for the access mode enumeration | 736 |
| ECachingModeClass | |
| Holds conversion methods for the caching mode enumeration | 737 |
| EDisplayNotationClass | |
| Holds conversion methods for the notation type of floats | 738 |
| EEndianessClass | |
| Holds conversion methods for the endianess enumeration | 739 |
| EGenApiSchemaVersionClass | |
| Helper class converting EGenApiSchemaVersion from and to string | 740 |
| EInputDirectionClass | |
| Holds conversion methods for the notation type of floats | 741 |
| ENameSpaceClass | |
| Holds conversion methods for the namespace enumeration | 742 |
| EnumEntryNode | |
| Interface for string properties | 743 |
| EnumNode | |
| Interface for string properties | 746 |
| ERepresentationClass | |
| Holds conversion methods for the representation enumeration | 751 |
| ESignClass | |
| Holds conversion methods for the sign enumeration | 752 |
| ESlopeClass | |
| Holds conversion methods for the converter formulas | 753 |
| EStandardNameSpaceClass | |
| Holds conversion methods for the standard namespace enumeration | 754 |
| Event | |
| The base class for all event types | 756 |
| EVisibilityClass | |
| Holds conversion methods for the visibility enumeration | 759 |
| Exception | |
| The Exception object represents an error that is returned from the library | 760 |
| EYesNoClass | |
| Holds conversion methods for the standard namespace enumeration | 765 |
| FileProtocolAdapter | |
| Adapter between the std::iostreambuf and the SFNC Features representing the device file system | 766 |
| FloatNode | |
| Interface for string properties | 770 |
| FloatRegNode | |
| Interface for string properties | 777 |
| Function_NodeCallback< Function > | |
| Container for a function pointer | 780 |
| gcstring | 782 |
| GVCP_CHUNK_TRAILER | |
| Header of a GVCP request packet | 791 |
| GVCP_EVENT_ITEM | |
| Layout of a GVCP event item (Extended ID flag not set) | 792 |
| GVCP_EVENT_ITEM_BASIC | |
| Layout of a GVCP event item (common to all types) | 793 |

| | |
|--|-----|
| GVCP_EVENT_ITEM_EXTENDED_ID | |
| Layout of a GVCP event item (Extended ID flag set) | 794 |
| GVCP_EVENT_REQUEST | |
| Layout of a GVCP event request packet (Extended ID flag not set) | 795 |
| GVCP_EVENT_REQUEST_EXTENDED_ID | |
| Layout of a GVCP event request packet (Extended ID flag set) | 796 |
| GVCP_EVENTDATA_REQUEST | |
| Layout of a GVCP event data request packet (Extended ID flag not set) | 797 |
| GVCP_EVENTDATA_REQUEST_EXTENDED_ID | |
| Layout of a GVCP event data request packet (Extended ID flag set) | 798 |
| GVCP_REQUEST_HEADER | |
| Header of a GVCP request packet | 799 |
| H264Option | |
| Options for saving H264 files | 800 |
| IArrivalEvent | 803 |
| ICameraBase | |
| The interface file for base class for the camera object | 805 |
| ICameraList | |
| Used to hold a list of camera objects | 814 |
| IChunkData | |
| The Interface file for ChunkData | 817 |
| IDataStream | 825 |
| IDevFileStreamBase< CharType, Traits > | 830 |
| IDevFileStreamBuf< CharType, Traits > | 832 |
| IDeviceEvent | 834 |
| IImage | |
| The interface file for Image | 837 |
| IImageEvent | 848 |
| IImageStatistics | |
| The interface file for image statistics | 850 |
| IInterface | |
| The interface file for Interface | 854 |
| IInterfaceEvent | 858 |
| IInterfaceList | |
| The interface file for InterfaceList class | 861 |
| ILoggingEvent | 863 |
| Image | |
| The image object class | 865 |
| ImageEvent | |
| A handler for capturing image arrival events | 893 |
| ImagePtr | |
| A reference tracked pointer to an image object | 896 |
| ImageStatistics | |
| Represents image statistics for an image | 898 |
| ImageUtility | |
| Static helper functions for the image object class | 905 |
| ImageUtilityHeatmap | |
| Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16 | 909 |
| ImageUtilityPolarization | |
| Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8 | 913 |
| int64_autovector_t | |
| Vector of integers with reference counting | 921 |
| IntegerNode | |
| Interface for string properties | 924 |
| Interface | |
| An interface object which holds a list of cameras | 930 |

| | |
|--|------|
| InterfaceEvent | |
| A handler to device arrival and removal events on all interfaces | 935 |
| InterfaceList | |
| A list of the available interfaces on the system | 937 |
| InterfacePtr | |
| A reference tracked pointer to the interface object | 941 |
| IntRegNode | |
| Interface for string properties | 943 |
| IRemovalEvent | 946 |
| ISystem | |
| The interface file for System | 948 |
| JPEGOOption | |
| Options for saving JPEG image | 953 |
| JPG2Option | |
| Options for saving JPEG2000 image | 955 |
| LibraryVersion | |
| Provides easier access to the current version of Spinnaker | 956 |
| LockableObject< Object >::Lock | |
| A scopelevel Lock class | 957 |
| LockableObject< Object > | |
| Instance-Lock for an object | 958 |
| LoggingEvent | |
| An event handler for capturing the device logging event | 960 |
| LoggingEventData | |
| The LoggingEventData object | 962 |
| LoggingEventDataPtr | |
| A reference tracked pointer to the LoggingEvent object | 966 |
| Member_NodeCallback< Client, Member > | |
| Container for a member function pointer | 968 |
| MJPGOption | |
| Options for saving MJPG files | 970 |
| Node | |
| Class common to all nodes | 971 |
| NodeMap | |
| Smart pointer template for NodeMaps with create function | 982 |
| CNodeMapFactory::NodeStatistics_t | 991 |
| ODevFileStreamBase< CharType, Traits > | 992 |
| ODevFileStreamBuf< CharType, Traits > | 994 |
| PGMOption | |
| Options for saving PGM images | 996 |
| PNGOption | |
| Options for saving PNG images | 997 |
| PortNode | |
| Interface for value properties | 999 |
| PortRecorder | |
| Interface for recording write commands on a port | 1004 |
| PortReplay | |
| Interface for replaying write commands on a port | 1007 |
| PPMOption | |
| Options for saving PPM images | 1009 |
| RegisterNode | |
| Interface for string properties | 1011 |
| RemovalEvent | |
| An event handler for capturing the device removal event | 1015 |
| SingleChunkData_t | 1017 |
| SingleChunkDataStr_t | 1018 |
| SpinTestCamera | 1019 |

| | | |
|---|---|------|
| SpinVideo | Provides the functionality for the user to record images to an AVI/MP4 file | 1019 |
| StringNode | Interface for string properties | 1024 |
| StringRegNode | Interface for string properties | 1028 |
| System | The system object is used to retrieve the list of interfaces and cameras available | 1031 |
| SystemPtr | A reference tracked pointer to a system object | 1040 |
| TIFFOption | Options for saving TIFF images | 1042 |
| TransportLayerDevice | Part of the QuickSpin API to provide access to camera information without having to first initialize the camera | 1044 |
| TransportLayerInterface | Part of the QuickSpin API to provide access to camera information without having to first initialize the camera | 1053 |
| TransportLayerStream | Part of the QuickSpin API to provide access to camera information without having to first initialize the camera | 1063 |
| TransportLayerSystem | Part of the QuickSpin API to provide access to camera information without having to first initialize the camera | 1070 |
| U3V_CHUNK_TRAILER | Header of a GVCP request packet | 1072 |
| U3V_COMMAND_HEADER | U3V/GenCP command header | 1073 |
| U3V_EVENT_DATA | U3V/GenCP EVENT_CMD specific command data | 1074 |
| U3V_EVENT_MESSAGE | Entire event data message (without the variable-sized data field) | 1075 |
| ValueNode | Interface for value properties | 1076 |
| Version_t | Version | 1080 |

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

| | |
|------------------------------------|------|
| include/ArrivalEvent.h | 1081 |
| include/AVIRecorder.h | 1083 |
| include/BasePtr.h | 1083 |
| include/Camera.h | 1085 |
| include/CameraBase.h | 1087 |
| include/CameraDefs.h | 1089 |
| include/CameraList.h | 1122 |
| include/CameraPtr.h | 1124 |
| include/ChunkData.h | 1126 |
| include/DeviceEvent.h | 1128 |
| include/Event.h | 1130 |
| include/Exception.h | 1132 |
| include/Image.h | 1133 |
| include/ImageEvent.h | 1135 |
| include/ImagePtr.h | 1136 |
| include/ImageStatistics.h | 1138 |
| include/ImageUtility.h | 1140 |
| include/ImageUtilityHeatmap.h | 1140 |
| include/ImageUtilityPolarization.h | 1141 |
| include/Interface.h | 1141 |
| include/InterfaceEvent.h | 1169 |
| include/InterfaceList.h | 1171 |
| include/InterfacePtr.h | 1172 |
| include/LoggingEvent.h | 1174 |
| include/LoggingEventData.h | 1175 |
| include/LoggingEventDataPtr.h | 1177 |
| include/RemovalEvent.h | 1179 |
| include/Spinnaker.h | 1324 |
| include/SpinnakerDefs.h | 1325 |
| include/SpinnakerPlatform.h | 1329 |
| include/SpinUpdate.h | 1329 |
| include/SpinVideo.h | 1332 |
| include/SpinVideoDefs.h | 1332 |
| include/System.h | 1333 |
| include/SystemPtr.h | 1335 |

| | |
|--|------|
| include/TransportLayerDefs.h | 1337 |
| include/TransportLayerDevice.h | 1339 |
| include/TransportLayerInterface.h | 1341 |
| include/TransportLayerStream.h | 1343 |
| include/TransportLayerSystem.h | 1345 |
| include/Interface/IArrivalEvent.h | 1143 |
| include/Interface/ICameraBase.h | 1145 |
| include/Interface/ICameraList.h | 1147 |
| include/Interface/IChunkData.h | 1149 |
| include/Interface/IDeviceEvent.h | 1151 |
| include/Interface/IImage.h | 1153 |
| include/Interface/IImageEvent.h | 1155 |
| include/Interface/IImageStatistics.h | 1156 |
| include/Interface/IInterface.h | 1158 |
| include/Interface/IInterfaceEvent.h | 1160 |
| include/Interface/IInterfaceList.h | 1162 |
| include/Interface/ILoggingEvent.h | 1163 |
| include/Interface/IRemovalEvent.h | 1165 |
| include/Interface/IStream.h | 1167 |
| include/Interface/ISystem.h | 1167 |
| include/SpinGenApi/Autovector.h | 1181 |
| include/SpinGenApi/Base.h | 1182 |
| include/SpinGenApi/BooleanNode.h | 1183 |
| include/SpinGenApi/CategoryNode.h | 1185 |
| include/SpinGenApi/ChunkAdapter.h | 1187 |
| include/SpinGenApi/ChunkAdapterDcam.h | 1189 |
| include/SpinGenApi/ChunkAdapterGeneric.h | 1191 |
| include/SpinGenApi/ChunkAdapterGEV.h | 1193 |
| include/SpinGenApi/ChunkAdapterU3V.h | 1195 |
| include/SpinGenApi/ChunkPort.h | 1197 |
| include/SpinGenApi/CommandNode.h | 1199 |
| include/SpinGenApi/Compatibility.h | 1202 |
| include/SpinGenApi/Container.h | 1203 |
| include/SpinGenApi/Counter.h | 1203 |
| include/SpinGenApi/EnumClasses.h | 1204 |
| include/SpinGenApi/EnumEntryNode.h | 1206 |
| include/SpinGenApi/EnumNode.h | 1208 |
| include/SpinGenApi/EnumNodeT.h | 1210 |
| include/SpinGenApi/EventAdapter.h | 1212 |
| include/SpinGenApi/EventAdapter1394.h | 1214 |
| include/SpinGenApi/EventAdapterGeneric.h | 1216 |
| include/SpinGenApi/EventAdapterGEV.h | 1218 |
| include/SpinGenApi/EventAdapterU3V.h | 1220 |
| include/SpinGenApi/EventPort.h | 1222 |
| include/SpinGenApi/Filestream.h | 1224 |
| include/SpinGenApi/FloatNode.h | 1226 |
| include/SpinGenApi/FloatRegNode.h | 1228 |
| include/SpinGenApi/GCBase.h | 1230 |
| include/SpinGenApi/GCString.h | 1231 |
| include/SpinGenApi/GCStringVector.h | 1233 |
| include/SpinGenApi/GCSynch.h | 1234 |
| include/SpinGenApi/GCTypes.h | 1235 |
| include/SpinGenApi/GCUtilities.h | 1238 |
| include/SpinGenApi/IBoolean.h | 1243 |
| include/SpinGenApi/ICategory.h | 1245 |
| include/SpinGenApi/IChunkPort.h | 1247 |
| include/SpinGenApi/ICommand.h | 1249 |
| include/SpinGenApi/IDestroy.h | 1251 |

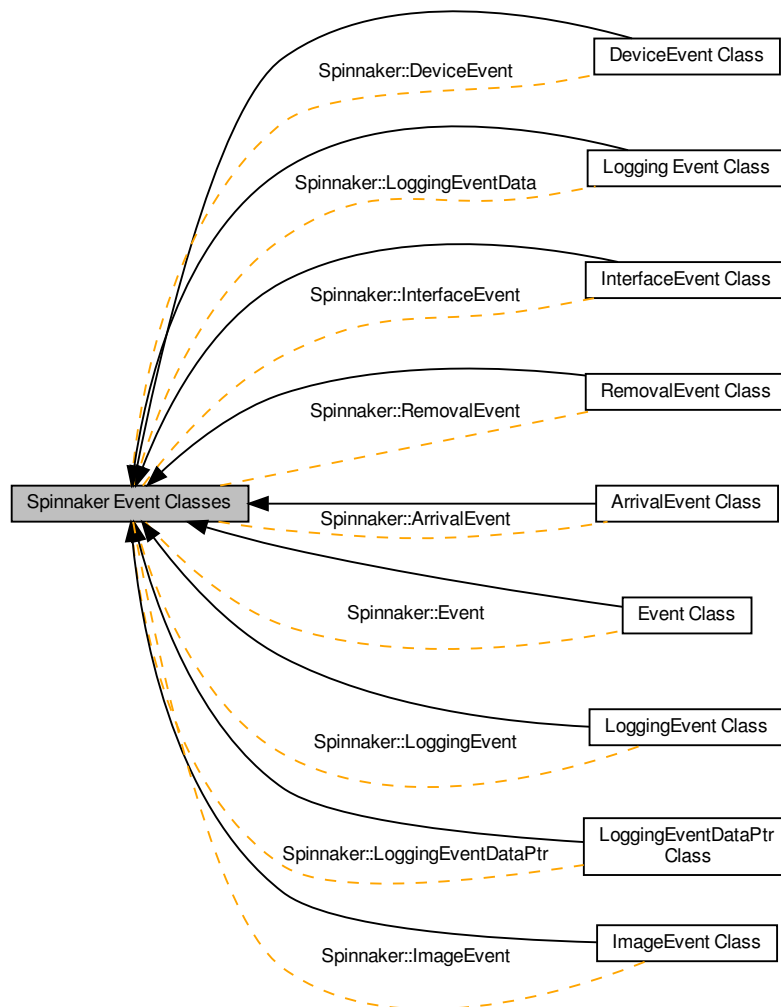
| | |
|---------------------------------------|------|
| include/SpinGenApi/IDeviceInfo.h | 1252 |
| include/SpinGenApi/IEnumEntry.h | 1254 |
| include/SpinGenApi/IEnumeration.h | 1256 |
| include/SpinGenApi/IEnumerationT.h | 1257 |
| include/SpinGenApi/IFloat.h | 1259 |
| include/SpinGenApi/IInteger.h | 1261 |
| include/SpinGenApi/INode.h | 1263 |
| include/SpinGenApi/INodeMap.h | 1266 |
| include/SpinGenApi/INodeMapDyn.h | 1267 |
| include/SpinGenApi/IntegerNode.h | 1269 |
| include/SpinGenApi/IntRegNode.h | 1271 |
| include/SpinGenApi/IPort.h | 1273 |
| include/SpinGenApi/IPortConstruct.h | 1274 |
| include/SpinGenApi/IPortRecorder.h | 1276 |
| include/SpinGenApi/IRegister.h | 1278 |
| include/SpinGenApi/ISelector.h | 1280 |
| include/SpinGenApi/ISelectorDigit.h | 1281 |
| include/SpinGenApi/IString.h | 1283 |
| include/SpinGenApi/IValue.h | 1285 |
| include/SpinGenApi/Node.h | 1286 |
| include/SpinGenApi/NodeCallback.h | 1288 |
| include/SpinGenApi/NodeCallbackImpl.h | 1290 |
| include/SpinGenApi/NodeMap.h | 1291 |
| include/SpinGenApi/NodeMapFactory.h | 1293 |
| include/SpinGenApi/NodeMapRef.h | 1294 |
| include/SpinGenApi/Persistence.h | 1295 |
| include/SpinGenApi/Pointer.h | 1297 |
| include/SpinGenApi/PortImpl.h | 1300 |
| include/SpinGenApi/PortNode.h | 1301 |
| include/SpinGenApi/PortRecorder.h | 1303 |
| include/SpinGenApi/PortReplay.h | 1304 |
| include/SpinGenApi/PortWriteList.h | 1305 |
| include/SpinGenApi/Reference.h | 1307 |
| include/SpinGenApi/RegisterNode.h | 1308 |
| include/SpinGenApi/RegisterPortImpl.h | 1310 |
| include/SpinGenApi/SelectorSet.h | 1310 |
| include/SpinGenApi/SpinnakerGenApi.h | 1311 |
| include/SpinGenApi/SpinTestCamera.h | 1313 |
| include/SpinGenApi/StringNode.h | 1313 |
| include/SpinGenApi/StringRegNode.h | 1315 |
| include/SpinGenApi/StructPort.h | 1317 |
| include/SpinGenApi/Synch.h | 1317 |
| include/SpinGenApi/Types.h | 1318 |
| include/SpinGenApi/ValueNode.h | 1322 |

Chapter 8

Module Documentation

8.1 Spinnaker Event Classes

Collaboration diagram for Spinnaker Event Classes:



Modules

- [ArrivalEvent Class](#)
- [DeviceEvent Class](#)
- [Event Class](#)
- [ImageEvent Class](#)
- [InterfaceEvent Class](#)
- [LoggingEvent Class](#)
- [Logging Event Class](#)
- [LoggingEventDataPtr Class](#)
- [RemovalEvent Class](#)

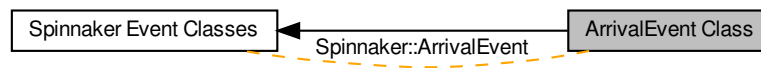
Classes

- class [ArrivalEvent](#)
An event handler for capturing the device arrival event.
- class [DeviceEvent](#)
A handler to device events.
- class [Event](#)
The base class for all event types.
- class [ImageEvent](#)
A handler for capturing image arrival events.
- class [InterfaceEvent](#)
A handler to device arrival and removal events on all interfaces.
- class [LoggingEvent](#)
An event handler for capturing the device logging event.
- class [LoggingEventData](#)
The [LoggingEventData](#) object.
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.
- class [RemovalEvent](#)
An event handler for capturing the device removal event.

8.1.1 Detailed Description

8.2 ArrivalEvent Class

Collaboration diagram for ArrivalEvent Class:



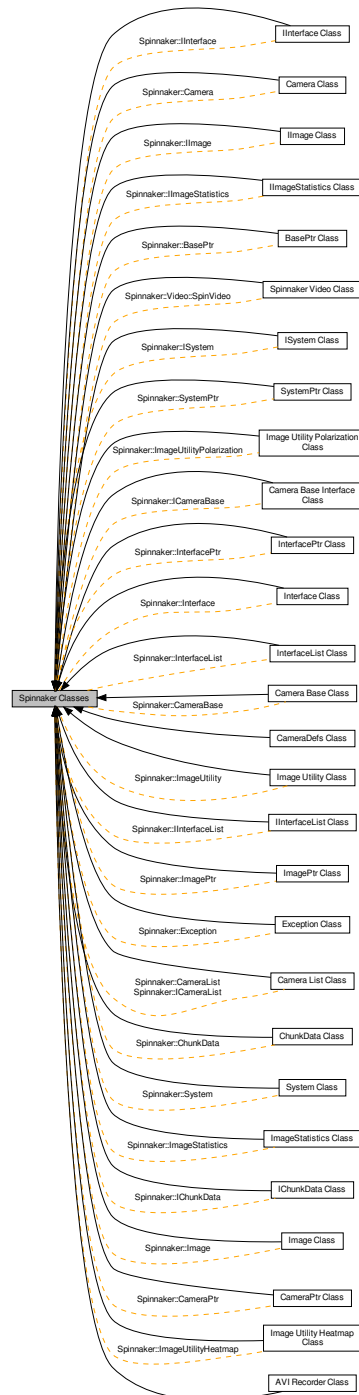
Classes

- class [ArrivalEvent](#)
An event handler for capturing the device arrival event.

8.2.1 Detailed Description

8.3 Spinnaker Classes

Collaboration diagram for Spinnaker Classes:



Modules

- [AVI Recorder Class](#)
- [BasePtr Class](#)

- [Camera Class](#)
- [Camera Base Class](#)
- [CameraDefs Class](#)
- [Camera List Class](#)
- [CameraPtr Class](#)
- [ChunkData Class](#)
- [Exception Class](#)
- [Image Class](#)
- [ImagePtr Class](#)
- [ImageStatistics Class](#)
- [Image Utility Class](#)
- [Image Utility Heatmap Class](#)
- [Image Utility Polarization Class](#)
- [Interface Class](#)
- [InterfaceList Class](#)
- [InterfacePtr Class](#)
- [Spinnaker Video Class](#)
- [System Class](#)
- [SystemPtr Class](#)
- [Camera Base Interface Class](#)
- [IChunkData Class](#)
- [IImage Class](#)
- [IImageStatistics Class](#)
- [IInterface Class](#)
- [IInterfaceList Class](#)
- [ISystem Class](#)

Classes

- class [BasePtr< T, B >](#)
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.
- 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 [Exception](#)
The [Exception](#) object represents an error that is returned from the library.
- class [Image](#)
The image object class.
- class [ImagePtr](#)
A reference tracked pointer to an image object.
- class [ImageStatistics](#)
Represents image statistics for an image.
- class [ImageUtility](#)
Static helper functions for the image object class.
- class [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.*

 - class [Interface](#)

An interface object which holds a list of cameras.
- class [InterfaceList](#)

A list of the available interfaces on the system.
- class [InterfacePtr](#)

A reference tracked pointer to the interface object.
- class [SpinVideo](#)

Provides the functionality for the user to record images to an AVI/MP4 file.
- class [System](#)

The system object is used to retrieve the list of interfaces and cameras available.
- class [SystemPtr](#)

A reference tracked pointer to a system object.
- 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 [IImage](#)

The interface file for [Image](#).
- class [IImageStatistics](#)

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

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

The interface file for [InterfaceList](#) class.
- class [ISystem](#)

The interface file for [System](#).

8.3.1 Detailed Description

8.4 AVI Recorder Class

Collaboration diagram for AVI Recorder Class:



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.

8.4.1 Detailed Description

8.4.2 Function Documentation

8.4.2.1 DEPRECATED_CLASS()

```

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

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

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

Default destructor.

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

Parameters

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

See also

AVIClose()

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

Parameters

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

See also

AVIClose()
MJPGOption

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

Parameters

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

See also

AVIClose()
H264Option

Append an image to the AVI/MP4 file.

Parameters

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

Close the AVI/MP4 file.

See also

AVIOpen()

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

Parameters

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

See also

`AVIAppend(ImagePtr pImage)`

8.5 BasePtr Class

Collaboration diagram for BasePtr Class:



Classes

- class [BasePtr< T, B >](#)

The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

8.5.1 Detailed Description

8.6 Camera Class

Collaboration diagram for Camera Class:



Classes

- class [Camera](#)
The camera object class.

8.6.1 Detailed Description

8.7 Camera Base Class

Collaboration diagram for Camera Base Class:



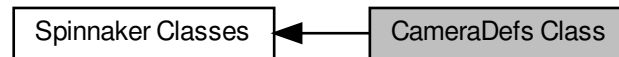
Classes

- class [CameraBase](#)
The base class for the camera object.

8.7.1 Detailed Description

8.8 CameraDefs Class

Collaboration diagram for CameraDefs Class:



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_DEVICEREGISTERSENDIANNES }

• 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_DEVICE TLTYPE }

• enum DevicePowerSupplySelectorEnums {
DevicePowerSupplySelector_External,
NUM_DEVICEPOWERSUPPLYSELECTOR }

• enum DeviceTemperatureSelectorEnums {
DeviceTemperatureSelector_Sensor,
NUM_DEVICE TEMPERATURESELECTOR }

• enum DeviceIndicatorModeEnums {
DeviceIndicatorMode_Inactive,
DeviceIndicatorMode_Active,
DeviceIndicatorMode_ErrorStatus,
NUM_DEVICE INDICATORMODE }

• 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_Deglintch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }

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

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

• enum LineSelectorEnums {
    LineSelector_Line0,

```

```

LineSelector_Line1,
LineSelector_Line2,
LineSelector_Line3,
NUM_LINESELECTOR }

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

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

• enum CounterSelectorEnums {
CounterSelector_Counter0,
CounterSelector_Counter1,
NUM_COUNTERSELECTOR }

• enum CounterStatusEnums {
CounterStatus_CounterIdle,
CounterStatus_CounterTriggerWait,
CounterStatus_CounterActive,
CounterStatus_CounterCompleted,
CounterStatus_CounterOverflow,
NUM_COUNTERSTATUS }

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

• enum CounterResetSourceEnums {
CounterResetSource_Off,
CounterResetSource_Line0,
CounterResetSource_Line1,
CounterResetSource_Line2,
CounterResetSource_Line3,
CounterResetSource_UserOutput0,
CounterResetSource_UserOutput1,
CounterResetSource_UserOutput2,
CounterResetSource_UserOutput3,
CounterResetSource_Counter0Start,

```

```

CounterResetSource_Counter1Start,
CounterResetSource_Counter0End,
CounterResetSource_Counter1End,
CounterResetSource_LogicBlock0,
CounterResetSource_LogicBlock1,
CounterResetSource_ExposureStart,
CounterResetSource_ExposureEnd,
CounterResetSource_FrameTriggerWait,
NUM_COUNTERRESETSOURCE }
• enum CounterEventSourceEnums {
CounterEventSource_Off,
CounterEventSource_MHzTick,
CounterEventSource_Line0,
CounterEventSource_Line1,
CounterEventSource_Line2,
CounterEventSource_Line3,
CounterEventSource_UserOutput0,
CounterEventSource_UserOutput1,
CounterEventSource_UserOutput2,
CounterEventSource_UserOutput3,
CounterEventSource_Counter0Start,
CounterEventSource_Counter1Start,
CounterEventSource_Counter0End,
CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
• enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
• enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
• enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }
• enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }
• enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }

```

- enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }
- enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }
- enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,
DeviceSerialPortBaudRate_Baud38400,
DeviceSerialPortBaudRate_Baud57600,
DeviceSerialPortBaudRate_Baud115200,
DeviceSerialPortBaudRate_Baud230400,
DeviceSerialPortBaudRate_Baud460800,
DeviceSerialPortBaudRate_Baud921600,
NUM_DEVICESERIALPORTBAUDRATE }
- enum SensorTapsEnums {
SensorTaps_One,
SensorTaps_Two,
SensorTaps_Three,
SensorTaps_Four,
SensorTaps_Eight,
SensorTaps_Ten,
NUM_SENSORTAPS }
- enum SensorDigitizationTapsEnums {
SensorDigitizationTaps_One,
SensorDigitizationTaps_Two,
SensorDigitizationTaps_Three,
SensorDigitizationTaps_Four,
SensorDigitizationTaps_Eight,
SensorDigitizationTaps_Ten,
NUM_SENSORDIGITIZATIONTAPS }
- enum RegionSelectorEnums {
RegionSelector_Region0,
RegionSelector_Region1,
RegionSelector_Region2,
RegionSelector_All,
NUM_REGIONSELECTOR }
- enum RegionModeEnums {
RegionMode_Off,
RegionMode_On,
NUM_REGIONMODE }
- enum RegionDestinationEnums {
RegionDestination_Stream0,

```

RegionDestination_Stream1,
RegionDestination_Stream2,
NUM_REGIONDESTINATION }

```

- `enum ImageComponentSelectorEnums {`
`ImageComponentSelector_Intensity,`
`ImageComponentSelector_Color,`
`ImageComponentSelector_Infrared,`
`ImageComponentSelector_Ultraviolet,`
`ImageComponentSelector_Range,`
`ImageComponentSelector_Disparity,`
`ImageComponentSelector_Confidence,`
`ImageComponentSelector_Scatter,`
`NUM_IMAGECOMPONENTSELECTOR }`
- `enum PixelFormatInfoSelectorEnums {`
`PixelFormatInfoSelector_Mono1p,`
`PixelFormatInfoSelector_Mono2p,`
`PixelFormatInfoSelector_Mono4p,`
`PixelFormatInfoSelector_Mono8,`
`PixelFormatInfoSelector_Mono8s,`
`PixelFormatInfoSelector_Mono10,`
`PixelFormatInfoSelector_Mono10p,`
`PixelFormatInfoSelector_Mono12,`
`PixelFormatInfoSelector_Mono12p,`
`PixelFormatInfoSelector_Mono14,`
`PixelFormatInfoSelector_Mono16,`
`PixelFormatInfoSelector_Mono16s,`
`PixelFormatInfoSelector_Mono32f,`
`PixelFormatInfoSelector_BayerBG8,`
`PixelFormatInfoSelector_BayerBG10,`
`PixelFormatInfoSelector_BayerBG10p,`
`PixelFormatInfoSelector_BayerBG12,`
`PixelFormatInfoSelector_BayerBG12p,`
`PixelFormatInfoSelector_BayerBG16,`
`PixelFormatInfoSelector_BayerGB8,`
`PixelFormatInfoSelector_BayerGB10,`
`PixelFormatInfoSelector_BayerGB10p,`
`PixelFormatInfoSelector_BayerGB12,`
`PixelFormatInfoSelector_BayerGB12p,`
`PixelFormatInfoSelector_BayerGB16,`
`PixelFormatInfoSelector_BayerGR8,`
`PixelFormatInfoSelector_BayerGR10,`
`PixelFormatInfoSelector_BayerGR10p,`
`PixelFormatInfoSelector_BayerGR12,`
`PixelFormatInfoSelector_BayerGR12p,`
`PixelFormatInfoSelector_BayerGR16,`
`PixelFormatInfoSelector_BayerRG8,`
`PixelFormatInfoSelector_BayerRG10,`
`PixelFormatInfoSelector_BayerRG10p,`
`PixelFormatInfoSelector_BayerRG12,`
`PixelFormatInfoSelector_BayerRG12p,`
`PixelFormatInfoSelector_BayerRG16,`
`PixelFormatInfoSelector_RGBa8,`
`PixelFormatInfoSelector_RGBa10,`
`PixelFormatInfoSelector_RGBa10p,`
`PixelFormatInfoSelector_RGBa12,`
`PixelFormatInfoSelector_RGBa12p,`
`PixelFormatInfoSelector_RGBa14,`
`PixelFormatInfoSelector_RGBa16,`


```
PixelFormatInfoSelector_RGB8,  
PixelFormatInfoSelector_RGB8_Planar,  
PixelFormatInfoSelector_RGB10,  
PixelFormatInfoSelector_RGB10_Planar,  
PixelFormatInfoSelector_RGB10p,  
PixelFormatInfoSelector_RGB10p32,  
PixelFormatInfoSelector_RGB12,  
PixelFormatInfoSelector_RGB12_Planar,  
PixelFormatInfoSelector_RGB12p,  
PixelFormatInfoSelector_RGB14,  
PixelFormatInfoSelector_RGB16,  
PixelFormatInfoSelector_RGB16s,  
PixelFormatInfoSelector_RGB32f,  
PixelFormatInfoSelector_RGB16_Planar,  
PixelFormatInfoSelector_RGB565p,  
PixelFormatInfoSelector_BGRa8,  
PixelFormatInfoSelector_BGRa10,  
PixelFormatInfoSelector_BGRa10p,  
PixelFormatInfoSelector_BGRa12,  
PixelFormatInfoSelector_BGRa12p,  
PixelFormatInfoSelector_BGRa14,  
PixelFormatInfoSelector_BGRa16,  
PixelFormatInfoSelector_RGBa32f,  
PixelFormatInfoSelector_BGR8,  
PixelFormatInfoSelector_BGR10,  
PixelFormatInfoSelector_BGR10p,  
PixelFormatInfoSelector_BGR12,  
PixelFormatInfoSelector_BGR12p,  
PixelFormatInfoSelector_BGR14,  
PixelFormatInfoSelector_BGR16,  
PixelFormatInfoSelector_BGR565p,  
PixelFormatInfoSelector_R8,  
PixelFormatInfoSelector_R10,  
PixelFormatInfoSelector_R12,  
PixelFormatInfoSelector_R16,  
PixelFormatInfoSelector_G8,  
PixelFormatInfoSelector_G10,  
PixelFormatInfoSelector_G12,  
PixelFormatInfoSelector_G16,  
PixelFormatInfoSelector_B8,  
PixelFormatInfoSelector_B10,  
PixelFormatInfoSelector_B12,  
PixelFormatInfoSelector_B16,  
PixelFormatInfoSelector_Coord3D_ABC8,  
PixelFormatInfoSelector_Coord3D_ABC8_Planar,  
PixelFormatInfoSelector_Coord3D_ABC10p,  
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,  
PixelFormatInfoSelector_Coord3D_ABC12p,  
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,  
PixelFormatInfoSelector_Coord3D_ABC16,  
PixelFormatInfoSelector_Coord3D_ABC16_Planar,  
PixelFormatInfoSelector_Coord3D_ABC32f,  
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,  
PixelFormatInfoSelector_Coord3D_AC8,  
PixelFormatInfoSelector_Coord3D_AC8_Planar,  
PixelFormatInfoSelector_Coord3D_AC10p,  
PixelFormatInfoSelector_Coord3D_AC10p_Planar,  
PixelFormatInfoSelector_Coord3D_AC12p,
```

[PixelFormatInfoSelector_Coord3D_AC12p_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC16](#),
[PixelFormatInfoSelector_Coord3D_AC16_Planar](#),
[PixelFormatInfoSelector_Coord3D_AC32f](#),
[PixelFormatInfoSelector_Coord3D_AC32f_Planar](#),
[PixelFormatInfoSelector_Coord3D_A8](#),
[PixelFormatInfoSelector_Coord3D_A10p](#),
[PixelFormatInfoSelector_Coord3D_A12p](#),
[PixelFormatInfoSelector_Coord3D_A16](#),
[PixelFormatInfoSelector_Coord3D_A32f](#),
[PixelFormatInfoSelector_Coord3D_B8](#),
[PixelFormatInfoSelector_Coord3D_B10p](#),
[PixelFormatInfoSelector_Coord3D_B12p](#),
[PixelFormatInfoSelector_Coord3D_B16](#),
[PixelFormatInfoSelector_Coord3D_B32f](#),
[PixelFormatInfoSelector_Coord3D_C8](#),
[PixelFormatInfoSelector_Coord3D_C10p](#),
[PixelFormatInfoSelector_Coord3D_C12p](#),
[PixelFormatInfoSelector_Coord3D_C16](#),
[PixelFormatInfoSelector_Coord3D_C32f](#),
[PixelFormatInfoSelector_Confidence1](#),
[PixelFormatInfoSelector_Confidence1p](#),
[PixelFormatInfoSelector_Confidence8](#),
[PixelFormatInfoSelector_Confidence16](#),
[PixelFormatInfoSelector_Confidence32f](#),
[PixelFormatInfoSelector_BiColorBGRG8](#),
[PixelFormatInfoSelector_BiColorBGRG10](#),
[PixelFormatInfoSelector_BiColorBGRG10p](#),
[PixelFormatInfoSelector_BiColorBGRG12](#),
[PixelFormatInfoSelector_BiColorBGRG12p](#),
[PixelFormatInfoSelector_BiColorRGBG8](#),
[PixelFormatInfoSelector_BiColorRGBG10](#),
[PixelFormatInfoSelector_BiColorRGBG10p](#),
[PixelFormatInfoSelector_BiColorRGBG12](#),
[PixelFormatInfoSelector_BiColorRGBG12p](#),
[PixelFormatInfoSelector_SCF1WBWG8](#),
[PixelFormatInfoSelector_SCF1WBWG10](#),
[PixelFormatInfoSelector_SCF1WBWG10p](#),
[PixelFormatInfoSelector_SCF1WBWG12](#),
[PixelFormatInfoSelector_SCF1WBWG12p](#),
[PixelFormatInfoSelector_SCF1WBWG14](#),
[PixelFormatInfoSelector_SCF1WBWG16](#),
[PixelFormatInfoSelector_SCF1WGWB8](#),
[PixelFormatInfoSelector_SCF1WGWB10](#),
[PixelFormatInfoSelector_SCF1WGWB10p](#),
[PixelFormatInfoSelector_SCF1WGWB12](#),
[PixelFormatInfoSelector_SCF1WGWB12p](#),
[PixelFormatInfoSelector_SCF1WGWB14](#),
[PixelFormatInfoSelector_SCF1WGWB16](#),
[PixelFormatInfoSelector_SCF1WGWR8](#),
[PixelFormatInfoSelector_SCF1WGWR10](#),
[PixelFormatInfoSelector_SCF1WGWR10p](#),
[PixelFormatInfoSelector_SCF1WGWR12](#),
[PixelFormatInfoSelector_SCF1WGWR12p](#),
[PixelFormatInfoSelector_SCF1WGWR14](#),
[PixelFormatInfoSelector_SCF1WGWR16](#),
[PixelFormatInfoSelector_SCF1WRWG8](#),
[PixelFormatInfoSelector_SCF1WRWG10](#),

PixelFormatInfoSelector_SCF1WRWG10p,
PixelFormatInfoSelector_SCF1WRWG12,
PixelFormatInfoSelector_SCF1WRWG12p,
PixelFormatInfoSelector_SCF1WRWG14,
PixelFormatInfoSelector_SCF1WRWG16,
PixelFormatInfoSelector_YCbCr8,
PixelFormatInfoSelector_YCbCr8_CbYCr,
PixelFormatInfoSelector_YCbCr10_CbYCr,
PixelFormatInfoSelector_YCbCr10p_CbYCr,
PixelFormatInfoSelector_YCbCr12_CbYCr,
PixelFormatInfoSelector_YCbCr12p_CbYCr,
PixelFormatInfoSelector_YCbCr411_8,
PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr422_8,
PixelFormatInfoSelector_YCbCr422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10,
PixelFormatInfoSelector_YCbCr422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr422_10p,
PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12,
PixelFormatInfoSelector_YCbCr422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr422_12p,
PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYYVYY,
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_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_CXPCONNECTIONTESTMODE }
• enum CxpPoCxpStatusEnums {
    CxpPoCxpStatus_Auto,
    CxpPoCxpStatus_Off,
    CxpPoCxpStatus_Tripped,
    NUM_CXPPOCXPSTATUS }

```

8.8.1 Detailed Description

8.8.2 Enumeration Type Documentation

8.8.2.1 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 | |

8.8.2.2 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_ACQUISITIONSTATUSSELECTION | |

8.8.2.3 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 | |

8.8.2.4 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 | |

8.8.2.5 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 | |

8.8.2.6 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 | |

8.8.2.7 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 | |

8.8.2.8 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 | |

8.8.2.9 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 | |

8.8.2.10 BalanceRatioSelectorEnums

enum `BalanceRatioSelectorEnums`

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

Enumerator

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

8.8.2.11 BalanceWhiteAutoEnums

```
enum BalanceWhiteAutoEnums
```

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

Enumerator

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

8.8.2.12 BalanceWhiteAutoProfileEnums

```
enum BalanceWhiteAutoProfileEnums
```

< Selects the profile used by BalanceWhiteAuto.

Enumerator

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

8.8.2.13 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 | |

8.8.2.14 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 | |

8.8.2.15 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 | |

8.8.2.16 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

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

8.8.2.17 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 <code>BlackLevel</code> . |
| <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> | |

8.8.2.18 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> | |

8.8.2.19 ChunkBlackLevelSelectorEnums

enum `ChunkBlackLevelSelectorEnums`

< Selects which black level to retrieve

Enumerator

| | |
|--|--|
| <code>ChunkBlackLevelSelector_All</code> | |
| <code>NUM_CHUNKBLACKLEVELSELECTOR</code> | |

8.8.2.20 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 | |

8.8.2.21 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 | |

8.8.2.22 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 | |

8.8.2.23 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 | |

8.8.2.24 ChunkGainSelectorEnums

enum [ChunkGainSelectorEnums](#)

< Selects which gain to retrieve

Enumerator

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

8.8.2.25 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. |

Enumerator

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

8.8.2.26 ChunkPixelFormatEnums

enum [ChunkPixelFormatEnums](#)

< Format of the pixel provided by the camera

Enumerator

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

8.8.2.27 ChunkRegionIDEnums

enum [ChunkRegionIDEnums](#)

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

Enumerator

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

8.8.2.28 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 | |

8.8.2.29 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 | |

8.8.2.30 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 | |

8.8.2.31 ChunkScan3dCoordinateSystemReferenceEnums

enum [ChunkScan3dCoordinateSystemReferenceEnums](#)

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

Enumerator

| | |
|---|---|
| ChunkScan3dCoordinateSystemReference_Anchor | Default value. Original fixed reference. The coordinate system fixed relative the camera reference point marker is used. |
| ChunkScan3dCoordinateSystemReference_↔ Transformed | Transformed reference system. The transformed coordinate system is used according to the definition in the rotation and translation matrices. |
| NUM_CHUNKSCAN3DCOORDINATESYSTEMREF↔ ERENCE | |

8.8.2.32 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 | |

8.8.2.33 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 | |

8.8.2.34 ChunkScan3dOutputModeEnums

enum [ChunkScan3dOutputModeEnums](#)

< Returns the Calibrated Mode of the payload image.

Enumerator

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

8.8.2.35 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 | |
| ChunkSelector_Gain | |
| ChunkSelector_BlackLevel | |
| ChunkSelector_PixelFormat | |
| ChunkSelector_Timestamp | |
| ChunkSelector_SequencerSetActive | |
| ChunkSelector_SerialData | |
| ChunkSelector_ExposureEndLineStatusAll | |
| NUM_CHUNKSELECTOR | |

8.8.2.36 ChunkSourceIDEnums

enum [ChunkSourceIDEnums](#)

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

Enumerator

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

8.8.2.37 ChunkTimerSelectorEnums

enum [ChunkTimerSelectorEnums](#)

< Selects which Timer to retrieve data from.

Enumerator

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

8.8.2.38 ChunkTransferStreamIDEnums

enum [ChunkTransferStreamIDEnums](#)

< Returns identifier of the stream that generated this block.

Enumerator

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

8.8.2.39 ClConfigurationEnums

enum [ClConfigurationEnums](#)

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

Enumerator

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

8.8.2.40 CTimeSlotsCountEnums

enum `CTimeSlotsCountEnums`

< 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

| | |
|------------------------------------|-------|
| <code>CTimeSlotsCount_One</code> | One |
| <code>CTimeSlotsCount_Two</code> | Two |
| <code>CTimeSlotsCount_Three</code> | Three |
| <code>NUM_CLTIMESLOTSCOUNT</code> | |

8.8.2.41 ColorTransformationSelectorEnums

enum `ColorTransformationSelectorEnums`

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

Enumerator

| | |
|---|--|
| <code>ColorTransformationSelector_RGBtoRGB</code> | |
| <code>ColorTransformationSelector_RGBtoYUV</code> | |
| <code>NUM_COLORTRANSFORMATIONSELECTOR</code> | |

8.8.2.42 ColorTransformationValueSelectorEnums

enum `ColorTransformationValueSelectorEnums`

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

Enumerator

| | |
|---|--|
| <code>ColorTransformationValueSelector_Gain00</code> | |
| <code>ColorTransformationValueSelector_Gain01</code> | |
| <code>ColorTransformationValueSelector_Gain02</code> | |
| <code>ColorTransformationValueSelector_Gain10</code> | |
| <code>ColorTransformationValueSelector_Gain11</code> | |
| <code>ColorTransformationValueSelector_Gain12</code> | |
| <code>ColorTransformationValueSelector_Gain20</code> | |
| <code>ColorTransformationValueSelector_Gain21</code> | |
| <code>ColorTransformationValueSelector_Gain22</code> | |
| <code>ColorTransformationValueSelector_Offset0</code> | |
| <code>ColorTransformationValueSelector_Offset1</code> | |
| <code>ColorTransformationValueSelector_Offset2</code> | |
| <code>NUM_COLORTRANSFORMATIONVALUESELECTOR</code> | |

8.8.2.43 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 | |

8.8.2.44 CounterEventSourceEnums

enum `CounterEventSourceEnums`

< Selects the event that will increment the counter

Enumerator

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

8.8.2.45 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 | |

8.8.2.46 CounterResetSourceEnums

enum `CounterResetSourceEnums`

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

Enumerator

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

8.8.2.47 CounterSelectorEnums

```
enum CounterSelectorEnums
```

< Selects which counter to configure

Enumerator

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

8.8.2.48 CounterStatusEnums

```
enum CounterStatusEnums
```

< Returns the current status of the Counter.

Enumerator

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

8.8.2.49 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 | |

8.8.2.50 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 | |

8.8.2.51 CxpConnectionTestModeEnums

enum [CxpConnectionTestModeEnums](#)

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

Enumerator

| | |
|-----------------------------|--------|
| CxpConnectionTestMode_Off | Off |
| CxpConnectionTestMode_Mode1 | Mode 1 |
| NUM_CXPCONNECTIONTESTMODE | |

8.8.2.52 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

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

8.8.2.53 CxpLinkConfigurationPreferredEnums

```
enum CxpLinkConfigurationPreferredEnums
```

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

Enumerator

| | |
|---------------------------------------|--|
| CxpLinkConfigurationPreferred_CXP1_X1 | 1 Connection operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X1 | 1 Connection operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X1 | 1 Connection operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X1 | 1 Connection operating at CXP-5 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X1 | 1 Connection operating at CXP-6 speed (6.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP1_X2 | 2 Connections operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X2 | 2 Connections operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X2 | 2 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X2 | 2 Connections operating at CXP-4 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X2 | 3 Connections operating at CXP-5 speed (6.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP1_X3 | 3 Connections operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X3 | 3 Connections operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X3 | 3 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X3 | 3 Connections operating at CXP-5 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X3 | 3 Connections operating at CXP-6 speed (6.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP1_X4 | 4 Connections operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X4 | 4 Connections operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X4 | 4 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X4 | 4 Connections operating at CXP-5 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X4 | 4 Connections operating at CXP-6 speed (6.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP1_X5 | 5 Connections operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X5 | 5 Connections operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X5 | 5 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X5 | 5 Connections operating at CXP-5 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X5 | 5 Connections operating at CXP-6 speed (6.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP1_X6 | 6 Connections operating at CXP-1 speed (1.25 Gbps). |
| CxpLinkConfigurationPreferred_CXP2_X6 | 6 Connections operating at CXP-2 speed (2.50 Gbps). |
| CxpLinkConfigurationPreferred_CXP3_X6 | 6 Connections operating at CXP-3 speed (3.125 Gbps). |
| CxpLinkConfigurationPreferred_CXP5_X6 | 6 Connections operating at CXP-5 speed (5.00 Gbps). |
| CxpLinkConfigurationPreferred_CXP6_X6 | 6 Connections operating at CXP-6 speed (6.25 Gbps). |
| NUM_CXPLINKCONFIGURATIONPREFERRED | |

8.8.2.54 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. |

Enumerator

| | |
|------------------------------------|--|
| 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). |
| 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 | |

8.8.2.55 CxpPoCxpStatusEnums

```
enum CxpPoCxpStatusEnums
```

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

Enumerator

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

8.8.2.56 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 | |

8.8.2.57 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 | |

8.8.2.58 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 | |

8.8.2.59 DefectCorrectionModeEnums

enum `DefectCorrectionModeEnums`

< Controls the method used for replacing defective pixels.

Enumerator

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

8.8.2.60 DeinterlacingEnums

enum `DeinterlacingEnums`

< Controls how the device performs de-interlacing.

Enumerator

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

8.8.2.61 DeviceCharacterSetEnums

enum `DeviceCharacterSetEnums`

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

Enumerator

| | |
|---------------------------------------|--|
| <code>DeviceCharacterSet_UTF8</code> | |
| <code>DeviceCharacterSet_ASCII</code> | |
| <code>NUM_DEVICECHARACTERSET</code> | |

8.8.2.62 DeviceClockSelectorEnums

enum [DeviceClockSelectorEnums](#)

< Selects the clock frequency to access from the device.

Enumerator

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

8.8.2.63 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 | |

8.8.2.64 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 | |

8.8.2.65 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 | |

8.8.2.66 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 | |

8.8.2.67 DevicePowerSupplySelectorEnums

enum `DevicePowerSupplySelectorEnums`

< Selects the power supply source to control or read.

Enumerator

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

8.8.2.68 DeviceRegistersEndiannessEnums

enum `DeviceRegistersEndiannessEnums`

< Endianness of the registers of the device.

Enumerator

| | |
|----------------------------------|--|
| DeviceRegistersEndianness_Little | |
| DeviceRegistersEndianness_Big | |
| NUM_DEVICEREGISTERSENDIANNESSE | |

8.8.2.69 DeviceScanTypeEnums

```
enum DeviceScanTypeEnums
```

< Scan type of the sensor of the device.

Enumerator

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

8.8.2.70 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_DEVICSERIALPORTBAUDRATE | |

8.8.2.71 DeviceSerialPortSelectorEnums

```
enum DeviceSerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

| | |
|-------------------------------------|---|
| DeviceSerialPortSelector_CameraLink | Serial port associated to the Camera link connection. |
| NUM_DEVICSERIALPORTSELECTOR | |

8.8.2.72 DeviceStreamChannelEndiannessEnums

enum [DeviceStreamChannelEndiannessEnums](#)

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

Enumerator

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

8.8.2.73 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 | |

8.8.2.74 DeviceTapGeometryEnums

enum [DeviceTapGeometryEnums](#)

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

Enumerator

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

Enumerator

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

8.8.2.75 DeviceTemperatureSelectorEnums

```
enum DeviceTemperatureSelectorEnums
```

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

Enumerator

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

8.8.2.76 DeviceTLTypeEnums

enum [DeviceTLTypeEnums](#)

< Transport Layer type of the device.

Enumerator

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

8.8.2.77 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 | |

8.8.2.78 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 | |

8.8.2.79 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 | |

8.8.2.80 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 | |

8.8.2.81 EncoderResetSourceEnums

enum [EncoderResetSourceEnums](#)

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

Enumerator

| | |
|---------------------------------------|---|
| EncoderResetSource_Off | Disable the Encoder Reset trigger. |
| EncoderResetSource_AcquisitionTrigger | Resets with the reception of the Acquisition Trigger. |
| EncoderResetSource_AcquisitionStart | Resets with the reception of the Acquisition Start. |
| EncoderResetSource_AcquisitionEnd | Resets with the reception of the Acquisition End. |
| EncoderResetSource_FrameTrigger | Resets with the reception of the Frame Start Trigger. |
| EncoderResetSource_FrameStart | Resets with the reception of the Frame Start. |
| EncoderResetSource_FrameEnd | Resets with the reception of the Frame End. |
| EncoderResetSource_ExposureStart | Resets with the reception of the Exposure Start. |
| EncoderResetSource_ExposureEnd | Resets with the reception of the Exposure End. |
| EncoderResetSource_Line0 | Resets by the chosen I/O Line. |
| EncoderResetSource_Line1 | Resets by the chosen I/O Line. |
| EncoderResetSource_Line2 | Resets by the chosen I/O Line. |
| EncoderResetSource_Counter0Start | Resets with the reception of the Counter Start. |
| EncoderResetSource_Counter1Start | Resets with the reception of the Counter Start. |
| EncoderResetSource_Counter2Start | Resets with the reception of the Counter Start. |
| EncoderResetSource_Counter0End | Resets with the reception of the Counter End. |
| EncoderResetSource_Counter1End | Resets with the reception of the Counter End. |
| EncoderResetSource_Counter2End | Resets with the reception of the Counter End. |
| EncoderResetSource_Timer0Start | Resets with the reception of the Timer Start. |
| EncoderResetSource_Timer1Start | Resets with the reception of the Timer Start. |
| EncoderResetSource_Timer2Start | Resets with the reception of the Timer Start. |
| EncoderResetSource_Timer0End | Resets with the reception of the Timer End. |
| EncoderResetSource_Timer1End | Resets with the reception of the Timer End. |
| EncoderResetSource_Timer2End | Resets with the reception of the Timer End. |
| EncoderResetSource_UserOutput0 | Resets by the chosen User Output bit. |
| EncoderResetSource_UserOutput1 | Resets by the chosen User Output bit. |
| EncoderResetSource_UserOutput2 | Resets by the chosen User Output bit. |
| EncoderResetSource_SoftwareSignal0 | Resets on the reception of the Software Signal. |
| EncoderResetSource_SoftwareSignal1 | Resets on the reception of the Software Signal. |
| EncoderResetSource_SoftwareSignal2 | Resets on the reception of the Software Signal. |
| EncoderResetSource_Action0 | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_Action1 | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_Action2 | Resets on assertions of the chosen action signal (Broadcasted signal on the transport layer). |
| EncoderResetSource_LinkTrigger0 | Resets on the reception of the chosen Link Trigger (received from the transport layer). |
| EncoderResetSource_LinkTrigger1 | Resets on the reception of the chosen Link Trigger (received from the transport layer). |

Enumerator

| | |
|---------------------------------|---|
| EncoderResetSource_LinkTrigger2 | Resets on the reception of the chosen Link Trigger (received from the transport layer). |
| NUM_ENCODERRESETSOURCE | |

8.8.2.82 EncoderSelectorEnums

```
enum EncoderSelectorEnums
```

< Selects which Encoder to configure.

Enumerator

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

8.8.2.83 EncoderSourceAEnums

```
enum EncoderSourceAEnums
```

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

Enumerator

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

8.8.2.84 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 | |

8.8.2.85 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 | |

8.8.2.86 EventNotificationEnums

enum [EventNotificationEnums](#)

< Enables/Disables the selected event.

Enumerator

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

8.8.2.87 EventSelectorEnums

enum [EventSelectorEnums](#)

< Selects which [Event](#) to enable or disable.

Enumerator

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

8.8.2.88 ExposureActiveModeEnums

enum [ExposureActiveModeEnums](#)

< Control sensor active exposure mode.

Enumerator

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

8.8.2.89 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 | |

8.8.2.90 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 | |

8.8.2.91 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 | |

8.8.2.92 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 | |

8.8.2.93 FileOpenModeEnums

enum `FileOpenModeEnums`

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

Enumerator

| | |
|------------------------|--|
| FileOpenMode_Read | |
| FileOpenMode_Write | |
| FileOpenMode_ReadWrite | |
| NUM_FILEOPENMODE | |

8.8.2.94 FileOperationSelectorEnums

enum `FileOperationSelectorEnums`

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

Enumerator

| | |
|------------------------------|--|
| FileOperationSelector_Open | |
| FileOperationSelector_Close | |
| FileOperationSelector_Read | |
| FileOperationSelector_Write | |
| FileOperationSelector_Delete | |
| NUM_FILEOPERATIONSELECTOR | |

8.8.2.95 FileOperationStatusEnums

enum `FileOperationStatusEnums`

< Represents the file operation execution status.

Enumerator

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

8.8.2.96 FileSelectorEnums

enum [FileSelectorEnums](#)

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

Enumerator

| | |
|-----------------------------|--|
| FileSelector_UserSetDefault | |
| FileSelector_UserSet0 | |
| FileSelector_UserSet1 | |
| FileSelector_UserFile1 | |
| FileSelector_SerialPort0 | |
| NUM_FILESELECTOR | |

8.8.2.97 GainAutoBalanceEnums

enum [GainAutoBalanceEnums](#)

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

Enumerator

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

8.8.2.98 GainAutoEnums

enum [GainAutoEnums](#)

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

Enumerator

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

8.8.2.99 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 | |

8.8.2.100 GevCCPEnums

enum [GevCCPEnums](#)

< Controls the device access privilege of an application.

Enumerator

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

8.8.2.101 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 | |

8.8.2.102 GevGVCPExtendedStatusCodesSelectorEnums

enum [GevGVCPExtendedStatusCodesSelectorEnums](#)

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

Enumerator

| | |
|---|-------------|
| GevGVCPExtendedStatusCodesSelector_Version1_1 | Version 1 1 |
| GevGVCPExtendedStatusCodesSelector_Version2_0 | Version 2 0 |
| NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR | |

8.8.2.103 GevGVSPExtendedIDModeEnums

enum [GevGVSPExtendedIDModeEnums](#)

< Enables the extended IDs mode.

Enumerator

| | |
|---------------------------|-----|
| GevGVSPExtendedIDMode_Off | Off |
| GevGVSPExtendedIDMode_On | On |
| NUM_GEVGVSPEXTENDEDIDMODE | |

8.8.2.104 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 | |

8.8.2.105 GevIEEE1588ModeEnums

enum [GevIEEE1588ModeEnums](#)

< Provides the mode of the IEEE 1588 clock.

Enumerator

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

8.8.2.106 GevIEEE1588StatusEnums

enum [GevIEEE1588StatusEnums](#)

< Provides the status of the IEEE 1588 clock.

Enumerator

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

8.8.2.107 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 | |

8.8.2.108 `GevPhysicalLinkConfigurationEnums`

enum `GevPhysicalLinkConfigurationEnums`

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

Enumerator

| | |
|--|-------------|
| <code>GevPhysicalLinkConfiguration_SingleLink</code> | Single Link |
| <code>GevPhysicalLinkConfiguration_MultiLink</code> | Multi Link |
| <code>GevPhysicalLinkConfiguration_StaticLAG</code> | Static LAG |
| <code>GevPhysicalLinkConfiguration_DynamicLAG</code> | Dynamic LAG |
| <code>NUM_GEVPHYSICALLINKCONFIGURATION</code> | |

8.8.2.109 `GevSupportedOptionSelectorEnums`

enum `GevSupportedOptionSelectorEnums`

< Selects the GEV option to interrogate for existing support.

Enumerator

| | |
|---|--|
| <code>GevSupportedOptionSelector_UserDefinedName</code> | |
| <code>GevSupportedOptionSelector_SerialNumber</code> | |
| <code>GevSupportedOptionSelector_HeartbeatDisable</code> | |
| <code>GevSupportedOptionSelector_LinkSpeed</code> | |
| <code>GevSupportedOptionSelector_CCPApplicationSocket</code> | |
| <code>GevSupportedOptionSelector_ManifestTable</code> | |
| <code>GevSupportedOptionSelector_TestData</code> | |
| <code>GevSupportedOptionSelector_DiscoveryAckDelay</code> | |
| <code>GevSupportedOptionSelector_DiscoveryAckDelayWritable</code> | |
| <code>GevSupportedOptionSelector_ExtendedStatusCodes</code> | |
| <code>GevSupportedOptionSelector_Action</code> | |
| <code>GevSupportedOptionSelector_PendingAck</code> | |
| <code>GevSupportedOptionSelector_EventData</code> | |
| <code>GevSupportedOptionSelector_Event</code> | |
| <code>GevSupportedOptionSelector_PacketResend</code> | |
| <code>GevSupportedOptionSelector_WriteMem</code> | |
| <code>GevSupportedOptionSelector_CommandsConcatenation</code> | |
| <code>GevSupportedOptionSelector_IPConfigurationLLA</code> | |
| <code>GevSupportedOptionSelector_IPConfigurationDHCP</code> | |
| <code>GevSupportedOptionSelector_IPConfigurationPersistentIP</code> | |
| <code>GevSupportedOptionSelector_StreamChannelSourceSocket</code> | |
| <code>GevSupportedOptionSelector_MessageChannelSourceSocket</code> | |
| <code>NUM_GEVSUPPORTEDOPTIONSELECTOR</code> | |

8.8.2.110 ImageComponentSelectorEnums

enum `ImageComponentSelectorEnums`

< Selects a component to activate data streaming from.

Enumerator

| | |
|---|---|
| <code>ImageComponentSelector_Intensity</code> | The acquisition of intensity of the reflected light is controlled. |
| <code>ImageComponentSelector_Color</code> | The acquisition of color of the reflected light is controlled |
| <code>ImageComponentSelector_Infrared</code> | The acquisition of non-visible infrared light is controlled. |
| <code>ImageComponentSelector_Ultraviolet</code> | The acquisition of non-visible ultraviolet light is controlled. |
| <code>ImageComponentSelector_Range</code> | 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. |
| <code>ImageComponentSelector_Disparity</code> | 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. |
| <code>ImageComponentSelector_Confidence</code> | 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. |
| <code>ImageComponentSelector_Scatter</code> | 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. |
| <code>NUM_IMAGECOMPONENTSELECTOR</code> | |

8.8.2.111 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

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

Enumerator

| | |
|--|--|
| ImageCompressionJPEGFormatOption_Progressive | Indicates this is a progressive (multi-scan) DCT-based JPEG. |
| NUM_IMAGECOMPRESSIONJPEGFORMATOPTION | |

8.8.2.112 ImageCompressionModeEnums

enum [ImageCompressionModeEnums](#)

<

Enumerator

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

8.8.2.113 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 | |

8.8.2.114 LineFormatEnums

enum [LineFormatEnums](#)

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

Enumerator

| | |
|------------------------|--|
| LineFormat_NoConnect | |
| LineFormat_TriState | |
| LineFormat_TTL | |
| LineFormat_LVDS | |
| LineFormat_RS422 | |
| LineFormat_OptoCoupled | |
| LineFormat_OpenDrain | |
| NUM_LINEFORMAT | |

8.8.2.115 LineInputFilterSelectorEnums

```
enum LineInputFilterSelectorEnums
```

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

Enumerator

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

8.8.2.116 LineModeEnums

```
enum LineModeEnums
```

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

Enumerator

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

8.8.2.117 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 | |

8.8.2.118 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 | |

8.8.2.119 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 | |

8.8.2.120 LogicBlockLUTInputSelectorEnums

```
enum LogicBlockLUTInputSelectorEnums
```

< Controls which LogicBlockLUT Input Source & Activation to access.

Enumerator

| | |
|-----------------------------------|--|
| LogicBlockLUTInputSelector_Input0 | |
| LogicBlockLUTInputSelector_Input1 | |
| LogicBlockLUTInputSelector_Input2 | |
| LogicBlockLUTInputSelector_Input3 | |
| NUM_LOGICBLOCKLUTINPUTSELECTOR | |

8.8.2.121 LogicBlockLUTInputSourceEnums

```
enum LogicBlockLUTInputSourceEnums
```

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

Enumerator

| | |
|--|---------------|
| LogicBlockLUTInputSource_Zero | Zero |
| LogicBlockLUTInputSource_Line0 | Line0 |
| LogicBlockLUTInputSource_Line1 | Line1 |
| LogicBlockLUTInputSource_Line2 | Line2 |
| LogicBlockLUTInputSource_Line3 | Line3 |
| LogicBlockLUTInputSource_UserOutput0 | UserOutput0 |
| LogicBlockLUTInputSource_UserOutput1 | UserOutput1 |
| LogicBlockLUTInputSource_UserOutput2 | UserOutput2 |
| LogicBlockLUTInputSource_UserOutput3 | UserOutput3 |
| LogicBlockLUTInputSource_Counter0Start | Counter0Start |
| LogicBlockLUTInputSource_Counter1Start | Counter1Start |
| LogicBlockLUTInputSource_Counter0End | Counter0End |

Enumerator

| | |
|--|-------------------|
| LogicBlockLUTInputSource_Counter1End | Counter1End |
| LogicBlockLUTInputSource_LogicBlock0 | LogicBlock0 |
| LogicBlockLUTInputSource_LogicBlock1 | LogicBlock1 |
| LogicBlockLUTInputSource_ExposureStart | ExposureStart |
| LogicBlockLUTInputSource_ExposureEnd | ExposureEnd |
| LogicBlockLUTInputSource_FrameTriggerWait | FrameTriggerWait |
| LogicBlockLUTInputSource_AcquisitionActive | AcquisitionActive |
| NUM_LOGICBLOCKLUTINPUTSOURCE | |

8.8.2.122 LogicBlockLUTSelectorEnums

```
enum LogicBlockLUTSelectorEnums
```

< Selects which LogicBlock LUT to configure

Enumerator

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

8.8.2.123 LogicBlockSelectorEnums

```
enum LogicBlockSelectorEnums
```

< Selects which LogicBlock to configure

Enumerator

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

8.8.2.124 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 | |

8.8.2.125 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 | |

8.8.2.126 PixelFormatEnums

```
enum PixelFormatEnums
```

< Format of the pixel provided by the camera.

Enumerator

| | |
|-----------------------------|--|
| PixelFormat_Mono8 | |
| PixelFormat_Mono16 | |
| PixelFormat_RGB8Packed | |
| PixelFormat_BayerGR8 | |
| PixelFormat_BayerRG8 | |
| PixelFormat_BayerGB8 | |
| PixelFormat_BayerBG8 | |
| PixelFormat_BayerGR16 | |
| PixelFormat_BayerRG16 | |
| PixelFormat_BayerGB16 | |
| PixelFormat_BayerBG16 | |
| PixelFormat_Mono12Packed | |
| PixelFormat_BayerGR12Packed | |
| PixelFormat_BayerRG12Packed | |
| PixelFormat_BayerGB12Packed | |
| PixelFormat_BayerBG12Packed | |

Enumerator

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

Enumerator

| | |
|-----------------------------------|--|
| PixelFormat_RGB8 | Red-Green-Blue 8-bit |
| PixelFormat_RGB8_Planar | Red-Green-Blue 8-bit planar |
| PixelFormat_RGB10 | Red-Green-Blue 10-bit unpacked |
| PixelFormat_RGB10_Planar | Red-Green-Blue 10-bit unpacked planar |
| PixelFormat_RGB10p | Red-Green-Blue 10-bit packed |
| PixelFormat_RGB10p32 | Red-Green-Blue 10-bit packed into 32-bit |
| PixelFormat_RGB12 | Red-Green-Blue 12-bit unpacked |
| PixelFormat_RGB12_Planar | Red-Green-Blue 12-bit unpacked planar |
| PixelFormat_RGB12p | Red-Green-Blue 12-bit packed |
| PixelFormat_RGB14 | Red-Green-Blue 14-bit unpacked |
| PixelFormat_RGB16 | Red-Green-Blue 16-bit |
| PixelFormat_RGB16s | Red-Green-Blue 16-bit signed |
| PixelFormat_RGB32f | Red-Green-Blue 32-bit float |
| PixelFormat_RGB16_Planar | Red-Green-Blue 16-bit planar |
| PixelFormat_RGB565p | Red-Green-Blue 5/6/5-bit packed |
| PixelFormat_BGRa10 | Blue-Green-Red-alpha 10-bit unpacked |
| PixelFormat_BGRa10p | Blue-Green-Red-alpha 10-bit packed |
| PixelFormat_BGRa12 | Blue-Green-Red-alpha 12-bit unpacked |
| PixelFormat_BGRa12p | Blue-Green-Red-alpha 12-bit packed |
| PixelFormat_BGRa14 | Blue-Green-Red-alpha 14-bit unpacked |
| PixelFormat_BGRa16 | Blue-Green-Red-alpha 16-bit |
| PixelFormat_RGBa32f | Red-Green-Blue-alpha 32-bit float |
| PixelFormat_BGR10 | Blue-Green-Red 10-bit unpacked |
| PixelFormat_BGR10p | Blue-Green-Red 10-bit packed |
| PixelFormat_BGR12 | Blue-Green-Red 12-bit unpacked |
| PixelFormat_BGR12p | Blue-Green-Red 12-bit packed |
| PixelFormat_BGR14 | Blue-Green-Red 14-bit unpacked |
| PixelFormat_BGR16 | Blue-Green-Red 16-bit |
| PixelFormat_BGR565p | Blue-Green-Red 5/6/5-bit packed |
| PixelFormat_R8 | Red 8-bit |
| PixelFormat_R10 | Red 10-bit |
| PixelFormat_R12 | Red 12-bit |
| PixelFormat_R16 | Red 16-bit |
| PixelFormat_G8 | Green 8-bit |
| PixelFormat_G10 | Green 10-bit |
| PixelFormat_G12 | Green 12-bit |
| PixelFormat_G16 | Green 16-bit |
| PixelFormat_B8 | Blue 8-bit |
| PixelFormat_B10 | Blue 10-bit |
| 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 |

Enumerator

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

Enumerator

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

Enumerator

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

Enumerator

| | |
|-----------------|--|
| NUM_PIXELFORMAT | |
|-----------------|--|

8.8.2.127 PixelFormatInfoSelectorEnums

```
enum PixelFormatInfoSelectorEnums
```

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

Enumerator

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

Enumerator

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

Enumerator

| | |
|---|--|
| PixelFormatInfoSelector_G12 | Green 12-bit |
| PixelFormatInfoSelector_G16 | Green 16-bit |
| PixelFormatInfoSelector_B8 | Blue 8-bit |
| PixelFormatInfoSelector_B10 | Blue 10-bit |
| PixelFormatInfoSelector_B12 | Blue 12-bit |
| PixelFormatInfoSelector_B16 | Blue 16-bit |
| PixelFormatInfoSelector_Coord3D_ABC8 | 3D coordinate A-B-C 8-bit |
| PixelFormatInfoSelector_Coord3D_ABC8_Planar | 3D coordinate A-B-C 8-bit planar |
| PixelFormatInfoSelector_Coord3D_ABC10p | 3D coordinate A-B-C 10-bit packed |
| PixelFormatInfoSelector_Coord3D_ABC10p_Planar | 3D coordinate A-B-C 10-bit packed planar |
| PixelFormatInfoSelector_Coord3D_ABC12p | 3D coordinate A-B-C 12-bit packed |
| PixelFormatInfoSelector_Coord3D_ABC12p_Planar | 3D coordinate A-B-C 12-bit packed planar |
| PixelFormatInfoSelector_Coord3D_ABC16 | 3D coordinate A-B-C 16-bit |
| PixelFormatInfoSelector_Coord3D_ABC16_Planar | 3D coordinate A-B-C 16-bit planar |
| PixelFormatInfoSelector_Coord3D_ABC32f | 3D coordinate A-B-C 32-bit floating point |
| PixelFormatInfoSelector_Coord3D_ABC32f_Planar | 3D coordinate A-B-C 32-bit floating point planar |
| PixelFormatInfoSelector_Coord3D_AC8 | 3D coordinate A-C 8-bit |
| PixelFormatInfoSelector_Coord3D_AC8_Planar | 3D coordinate A-C 8-bit planar |
| PixelFormatInfoSelector_Coord3D_AC10p | 3D coordinate A-C 10-bit packed |
| PixelFormatInfoSelector_Coord3D_AC10p_Planar | 3D coordinate A-C 10-bit packed planar |
| PixelFormatInfoSelector_Coord3D_AC12p | 3D coordinate A-C 12-bit packed |
| PixelFormatInfoSelector_Coord3D_AC12p_Planar | 3D coordinate A-C 12-bit packed planar |
| PixelFormatInfoSelector_Coord3D_AC16 | 3D coordinate A-C 16-bit |
| PixelFormatInfoSelector_Coord3D_AC16_Planar | 3D coordinate A-C 16-bit planar |
| PixelFormatInfoSelector_Coord3D_AC32f | 3D coordinate A-C 32-bit floating point |
| PixelFormatInfoSelector_Coord3D_AC32f_Planar | 3D coordinate A-C 32-bit floating point planar |
| PixelFormatInfoSelector_Coord3D_A8 | 3D coordinate A 8-bit |
| PixelFormatInfoSelector_Coord3D_A10p | 3D coordinate A 10-bit packed |
| PixelFormatInfoSelector_Coord3D_A12p | 3D coordinate A 12-bit packed |
| PixelFormatInfoSelector_Coord3D_A16 | 3D coordinate A 16-bit |
| PixelFormatInfoSelector_Coord3D_A32f | 3D coordinate A 32-bit floating point |
| PixelFormatInfoSelector_Coord3D_B8 | 3D coordinate B 8-bit |
| PixelFormatInfoSelector_Coord3D_B10p | 3D coordinate B 10-bit packed |
| PixelFormatInfoSelector_Coord3D_B12p | 3D coordinate B 12-bit packed |
| PixelFormatInfoSelector_Coord3D_B16 | 3D coordinate B 16-bit |
| PixelFormatInfoSelector_Coord3D_B32f | 3D coordinate B 32-bit floating point |
| 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 |

Enumerator

| | |
|--|---|
| PixelFormatInfoSelector_BiColorBGRG10 | Bi-color Blue/Green - Red/Green 10-bit unpacked |
| PixelFormatInfoSelector_BiColorBGRG10p | Bi-color Blue/Green - Red/Green 10-bit packed |
| PixelFormatInfoSelector_BiColorBGRG12 | Bi-color Blue/Green - Red/Green 12-bit unpacked |
| PixelFormatInfoSelector_BiColorBGRG12p | Bi-color Blue/Green - Red/Green 12-bit packed |
| PixelFormatInfoSelector_BiColorRGBG8 | Bi-color Red/Green - Blue/Green 8-bit |
| PixelFormatInfoSelector_BiColorRGBG10 | Bi-color Red/Green - Blue/Green 10-bit unpacked |
| PixelFormatInfoSelector_BiColorRGBG10p | Bi-color Red/Green - Blue/Green 10-bit packed |
| PixelFormatInfoSelector_BiColorRGBG12 | Bi-color Red/Green - Blue/Green 12-bit unpacked |
| PixelFormatInfoSelector_BiColorRGBG12p | Bi-color Red/Green - Blue/Green 12-bit packed |
| PixelFormatInfoSelector_SCF1WBWG8 | Sparse Color Filter #1 White-Blue-White-Green 8-bit |
| PixelFormatInfoSelector_SCF1WBWG10 | Sparse Color Filter #1 White-Blue-White-Green 10-bit unpacked |
| PixelFormatInfoSelector_SCF1WBWG10p | Sparse Color Filter #1 White-Blue-White-Green 10-bit packed |
| PixelFormatInfoSelector_SCF1WBWG12 | Sparse Color Filter #1 White-Blue-White-Green 12-bit unpacked |
| PixelFormatInfoSelector_SCF1WBWG12p | Sparse Color Filter #1 White-Blue-White-Green 12-bit packed |
| PixelFormatInfoSelector_SCF1WBWG14 | Sparse Color Filter #1 White-Blue-White-Green 14-bit unpacked |
| PixelFormatInfoSelector_SCF1WBWG16 | Sparse Color Filter #1 White-Blue-White-Green 16-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB8 | Sparse Color Filter #1 White-Green-White-Blue 8-bit |
| PixelFormatInfoSelector_SCF1WGWB10 | Sparse Color Filter #1 White-Green-White-Blue 10-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB10p | Sparse Color Filter #1 White-Green-White-Blue 10-bit packed |
| PixelFormatInfoSelector_SCF1WGWB12 | Sparse Color Filter #1 White-Green-White-Blue 12-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB12p | Sparse Color Filter #1 White-Green-White-Blue 12-bit packed |
| PixelFormatInfoSelector_SCF1WGWB14 | Sparse Color Filter #1 White-Green-White-Blue 14-bit unpacked |
| PixelFormatInfoSelector_SCF1WGWB16 | Sparse Color Filter #1 White-Green-White-Blue 16-bit |
| PixelFormatInfoSelector_SCF1WGWR8 | Sparse Color Filter #1 White-Green-White-Red 8-bit |
| PixelFormatInfoSelector_SCF1WGWR10 | Sparse Color Filter #1 White-Green-White-Red 10-bit unpacked |
| 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 |

Enumerator

| | |
|--|--|
| PixelFormatInfoSelector_SCF1WRWG10p | Sparse Color Filter #1 White-Red-White-Green 10-bit packed |
| PixelFormatInfoSelector_SCF1WRWG12 | Sparse Color Filter #1 White-Red-White-Green 12-bit unpacked |
| PixelFormatInfoSelector_SCF1WRWG12p | Sparse Color Filter #1 White-Red-White-Green 12-bit packed |
| PixelFormatInfoSelector_SCF1WRWG14 | Sparse Color Filter #1 White-Red-White-Green 14-bit unpacked |
| PixelFormatInfoSelector_SCF1WRWG16 | Sparse Color Filter #1 White-Red-White-Green 16-bit |
| PixelFormatInfoSelector_YCbCr8 | YCbCr 4:4:4 8-bit |
| PixelFormatInfoSelector_YCbCr8_CbYCr | YCbCr 4:4:4 8-bit |
| PixelFormatInfoSelector_YCbCr10_CbYCr | YCbCr 4:4:4 10-bit unpacked |
| PixelFormatInfoSelector_YCbCr10p_CbYCr | YCbCr 4:4:4 10-bit packed |
| PixelFormatInfoSelector_YCbCr12_CbYCr | YCbCr 4:4:4 12-bit unpacked |
| PixelFormatInfoSelector_YCbCr12p_CbYCr | YCbCr 4:4:4 12-bit packed |
| PixelFormatInfoSelector_YCbCr411_8 | YCbCr 4:1:1 8-bit |
| PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY | YCbCr 4:1:1 8-bit |
| PixelFormatInfoSelector_YCbCr422_8 | YCbCr 4:2:2 8-bit |
| PixelFormatInfoSelector_YCbCr422_8_CbYCrY | YCbCr 4:2:2 8-bit |
| PixelFormatInfoSelector_YCbCr422_10 | YCbCr 4:2:2 10-bit unpacked |
| PixelFormatInfoSelector_YCbCr422_10_CbYCrY | YCbCr 4:2:2 10-bit unpacked |
| PixelFormatInfoSelector_YCbCr422_10p | YCbCr 4:2:2 10-bit packed |
| PixelFormatInfoSelector_YCbCr422_10p_CbYCrY | YCbCr 4:2:2 10-bit packed |
| PixelFormatInfoSelector_YCbCr422_12 | YCbCr 4:2:2 12-bit unpacked |
| PixelFormatInfoSelector_YCbCr422_12_CbYCrY | YCbCr 4:2:2 12-bit unpacked |
| PixelFormatInfoSelector_YCbCr422_12p | YCbCr 4:2:2 12-bit packed |
| PixelFormatInfoSelector_YCbCr422_12p_CbYCrY | YCbCr 4:2:2 12-bit packed |
| PixelFormatInfoSelector_YCbCr601_8_CbYCr | YCbCr 4:4:4 8-bit BT.601 |
| PixelFormatInfoSelector_YCbCr601_10_CbYCr | YCbCr 4:4:4 10-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_10p_CbYCr | YCbCr 4:4:4 10-bit packed BT.601 |
| PixelFormatInfoSelector_YCbCr601_12_CbYCr | YCbCr 4:4:4 12-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_12p_CbYCr | YCbCr 4:4:4 12-bit packed BT.601 |
| PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY | YCbCr 4:1:1 8-bit BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_8 | YCbCr 4:2:2 8-bit BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY | YCbCr 4:2:2 8-bit BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_10 | YCbCr 4:2:2 10-bit unpacked BT.601 |
| PixelFormatInfoSelector_YCbCr601_422_10_CbYYCrY | 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_CbYYCrY | 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_CbYYCrY | 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_CbYYCrY | YCbCr 4:2:2 12-bit packed BT.601 |
| PixelFormatInfoSelector_YCbCr709_8_CbYCr | YCbCr 4:4:4 8-bit BT.709 |

Enumerator

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

8.8.2.128 PixelSizeEnums

```
enum PixelSizeEnums
```

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

Enumerator

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

8.8.2.129 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 | |

8.8.2.130 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 | |

8.8.2.131 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

| | |
|-------------------------------------|--|
| <code>RegionSelector_Region0</code> | Selected feature will control the region 0. |
| <code>RegionSelector_Region1</code> | Selected feature will control the region 1. |
| <code>RegionSelector_Region2</code> | Selected feature will control the region 2. |
| <code>RegionSelector_All</code> | Selected features will control all the regions at the same time. |
| <code>NUM_REGIONSELECTOR</code> | |

8.8.2.132 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

| | |
|---|--|
| <code>RgbTransformLightSource_General</code> | Uses a matrix calibrated for a wide range of light sources. |
| <code>RgbTransformLightSource_Tungsten2800K</code> | Uses a matrix optimized for tungsten/incandescent light with color temperature 2800K. |
| <code>RgbTransformLightSource_WarmFluorescent3000K</code> | Uses a matrix optimized for a typical warm fluorescent light with color temperature 3000K. |
| <code>RgbTransformLightSource_CoolFluorescent4000K</code> | Uses a matrix optimized for a typical cool fluorescent light with color temperature 4000K. |
| <code>RgbTransformLightSource_Daylight5000K</code> | Uses a matrix optimized for noon Daylight with color temperature 5000K. |
| <code>RgbTransformLightSource_Cloudy6500K</code> | Uses a matrix optimized for a cloudy sky with color temperature 6500K. |
| <code>RgbTransformLightSource_Shade8000K</code> | Uses a matrix optimized for shade with color temperature 8000K. |
| <code>RgbTransformLightSource_Custom</code> | Uses a custom matrix set by the user through the <code>ColorTransformationValueSelector</code> and <code>ColorTransformationValue</code> controls. |
| <code>NUM_RGBTRANSFORMLIGHTSOURCE</code> | |

8.8.2.133 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 | |

8.8.2.134 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 | |

8.8.2.135 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 | |

8.8.2.136 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 | |

8.8.2.137 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 | |

8.8.2.138 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 | |

8.8.2.139 Scan3dOutputModeEnums

enum [Scan3dOutputModeEnums](#)

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

Enumerator

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

8.8.2.140 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 | |

8.8.2.141 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 | |

8.8.2.142 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 | |

8.8.2.143 SequencerConfigurationModeEnums

enum [SequencerConfigurationModeEnums](#)

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

Enumerator

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

8.8.2.144 SequencerConfigurationValidEnums

enum [SequencerConfigurationValidEnums](#)

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

Enumerator

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

8.8.2.145 SequencerModeEnums

enum [SequencerModeEnums](#)

< Controls whether or not a sequencer is active.

Enumerator

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

8.8.2.146 SequencerSetValidEnums

enum [SequencerSetValidEnums](#)

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

Enumerator

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

8.8.2.147 SequencerTriggerActivationEnums

enum [SequencerTriggerActivationEnums](#)

< Specifies the activation mode of the sequencer trigger.

Enumerator

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

8.8.2.148 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 | |

8.8.2.149 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 | |

8.8.2.150 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 | |

8.8.2.151 SerialPortSelectorEnums

```
enum SerialPortSelectorEnums
```

< Selects which serial port of the device to control.

Enumerator

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

8.8.2.152 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 | |

8.8.2.153 SerialPortStopBitsEnums

enum [SerialPortStopBitsEnums](#)

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

Enumerator

| | |
|----------------------------------|--|
| SerialPortStopBits_Bits1 | |
| SerialPortStopBits_Bits1AndAHalf | |
| SerialPortStopBits_Bits2 | |
| NUM_SERIALPORTSTOPBITS | |

8.8.2.154 SoftwareSignalSelectorEnums

enum [SoftwareSignalSelectorEnums](#)

< Selects which Software Signal features to control.

Enumerator

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

8.8.2.155 SourceSelectorEnums

```
enum SourceSelectorEnums
```

< Selects the source to control.

Enumerator

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

8.8.2.156 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 | |

8.8.2.157 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 | |

8.8.2.158 TimerSelectorEnums

enum `TimerSelectorEnums`

< Selects which Timer to configure.

Enumerator

| | |
|-----------------------------------|----------------------|
| <code>TimerSelector_Timer0</code> | Selects the Timer 0. |
| <code>TimerSelector_Timer1</code> | Selects the Timer 1. |
| <code>TimerSelector_Timer2</code> | Selects the Timer 2. |
| <code>NUM_TIMERSELECTOR</code> | |

8.8.2.159 TimerStatusEnums

enum `TimerStatusEnums`

< Returns the current status of the Timer.

Enumerator

| | |
|---|---|
| <code>TimerStatus_TimerIdle</code> | The Timer is idle. |
| <code>TimerStatus_TimerTriggerWait</code> | The Timer is waiting for a start trigger. |
| <code>TimerStatus_TimerActive</code> | The Timer is counting for the specified duration. |
| <code>TimerStatus_TimerCompleted</code> | The Timer reached the <code>TimerDuration</code> count. |
| <code>NUM_TIMERSTATUS</code> | |

8.8.2.160 TimerTriggerActivationEnums

enum `TimerTriggerActivationEnums`

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

Enumerator

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

8.8.2.161 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. |

Enumerator

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

8.8.2.162 TransferComponentSelectorEnums

enum [TransferComponentSelectorEnums](#)

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

Enumerator

| | |
|---------------------------------|--|
| TransferComponentSelector_Red | The TransferStreamChannel feature controls the index of the stream channel for the streaming of the red plane of the planar pixel formats. |
| TransferComponentSelector_Green | The TransferStreamChannel feature controls the index of the stream channel for the streaming of the green plane of the planar pixel formats. |
| TransferComponentSelector_Blue | The TransferStreamChannel feature controls the index of the stream channel for the streaming of blue plane of the planar pixel formats. |
| TransferComponentSelector_All | The TransferStreamChannel feature controls the index of the stream channel for the streaming of all the planes of the planar pixel formats simultaneously or non planar pixel formats. |
| NUM_TRANSFERCOMPONENTSELECTOR | |

8.8.2.163 TransferControlModeEnums

enum [TransferControlModeEnums](#)

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

Enumerator

| | |
|------------------------------------|-----------------|
| TransferControlMode_Basic | Basic |
| TransferControlMode_Automatic | Automatic |
| TransferControlMode_UserControlled | User Controlled |
| NUM_TRANSFERCONTROLMODE | |

8.8.2.164 TransferOperationModeEnums

enum [TransferOperationModeEnums](#)

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

Enumerator

| | |
|----------------------------------|-------------|
| TransferOperationMode_Continuous | Continuous |
| TransferOperationMode_MultiBlock | Multi Block |
| NUM_TRANSFEROPERATIONMODE | |

8.8.2.165 TransferQueueModeEnums

enum [TransferQueueModeEnums](#)

< Specifies the operation mode of the transfer queue.

Enumerator

| | |
|-----------------------------------|--|
| TransferQueueMode_FirstInFirstOut | Blocks first In are transferred Out first. |
| NUM_TRANSFERQUEUEMODE | |

8.8.2.166 TransferSelectorEnums

enum [TransferSelectorEnums](#)

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

Enumerator

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

8.8.2.167 TransferStatusSelectorEnums

enum `TransferStatusSelectorEnums`

< Selects which status of the transfer module to read.

Enumerator

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

8.8.2.168 TransferTriggerActivationEnums

enum `TransferTriggerActivationEnums`

< Specifies the activation mode of the transfer control trigger.

Enumerator

| | |
|--|---|
| <code>TransferTriggerActivation_RisingEdge</code> | Specifies that the trigger is considered valid on the rising edge of the source signal. |
| <code>TransferTriggerActivation_FallingEdge</code> | Specifies that the trigger is considered valid on the falling edge of the source signal. |
| <code>TransferTriggerActivation_AnyEdge</code> | Specifies that the trigger is considered valid on the falling or rising edge of the source signal. |
| <code>TransferTriggerActivation_LevelHigh</code> | 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. |
| <code>TransferTriggerActivation_LevelLow</code> | 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. |
| <code>NUM_TRANSFERTRIGGERACTIVATION</code> | |

8.8.2.169 TransferTriggerModeEnums

enum `TransferTriggerModeEnums`

< Controls if the selected trigger is active.

Enumerator

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

8.8.2.170 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 | |

8.8.2.171 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. |

Enumerator

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

8.8.2.172 TriggerActivationEnums

```
enum TriggerActivationEnums
```

< Specifies the activation mode of the trigger.

Enumerator

| | |
|-------------------------------|--|
| TriggerActivation_LevelLow | |
| TriggerActivation_LevelHigh | |
| TriggerActivation_FallingEdge | |
| TriggerActivation_RisingEdge | |
| TriggerActivation_AnyEdge | |
| NUM_TRIGGERACTIVATION | |

8.8.2.173 TriggerModeEnums

enum [TriggerModeEnums](#)

< Controls whether or not trigger is active.

Enumerator

| | |
|-----------------|--|
| TriggerMode_Off | |
| TriggerMode_On | |
| NUM_TRIGGERMODE | |

8.8.2.174 TriggerOverlapEnums

enum [TriggerOverlapEnums](#)

< Specifies the overlap mode of the trigger.

Enumerator

| | |
|------------------------------|--|
| TriggerOverlap_Off | |
| TriggerOverlap_ReadOut | |
| TriggerOverlap_PreviousFrame | |
| NUM_TRIGGEROVERLAP | |

8.8.2.175 TriggerSelectorEnums

enum [TriggerSelectorEnums](#)

< Selects the type of trigger to configure.

Enumerator

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

8.8.2.176 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 | |

8.8.2.177 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 | |

8.8.2.178 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 | |

8.8.2.179 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 | |

8.8.2.180 WhiteClipSelectorEnums

```
enum WhiteClipSelectorEnums
```

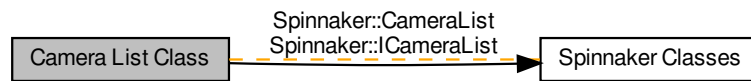
< Selects which White Clip to control.

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 | |

8.9 Camera List Class

Collaboration diagram for Camera List Class:



Classes

- class [CameraList](#)
Used to hold a list of camera objects.
- class [ICameraList](#)
Used to hold a list of camera objects.

8.9.1 Detailed Description

8.10 CameraPtr Class

Collaboration diagram for CameraPtr Class:



Classes

- class `CameraPtr`
A reference tracked pointer to a camera object.

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 ()`

8.10.1 Detailed Description

8.10.2 Function Documentation

8.10.2.1 `CameraPtr()` [1/4]

```
CameraPtr ( ) throw ( ) [inline]
```

Default constructor.

8.10.2.2 `CameraPtr()` [2/4]

```
CameraPtr (
    const int ) throw ( ) [inline]
```

Default constructor.

8.10.2.3 CameraPtr() [3/4]

```
CameraPtr (
    const long ) throw )    [inline]
```

Default constructor with argument.

8.10.2.4 CameraPtr() [4/4]

```
CameraPtr (
    const std::nullptr_t ) throw )    [inline]
```

8.11 ChunkData Class

Collaboration diagram for ChunkData Class:



Classes

- class [ChunkData](#)

The chunk data which contains additional information about an image.

8.11.1 Detailed Description

8.12 DeviceEvent Class

Collaboration diagram for DeviceEvent Class:



Classes

- class [DeviceEvent](#)
A handler to device events.

8.12.1 Detailed Description

8.13 Event Class

Collaboration diagram for Event Class:



Classes

- class [Event](#)

The base class for all event types.

8.13.1 Detailed Description

8.14 Exception Class

Collaboration diagram for Exception Class:



Classes

- class [Exception](#)

The [Exception](#) object represents an error that is returned from the library.

8.14.1 Detailed Description

8.15 Image Class

Collaboration diagram for Image Class:



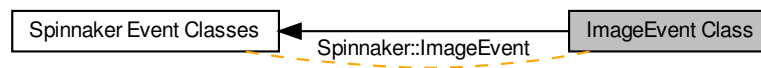
Classes

- class [Image](#)
The image object class.

8.15.1 Detailed Description

8.16 ImageEvent Class

Collaboration diagram for ImageEvent Class:



Classes

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

8.16.1 Detailed Description

8.17 ImagePtr Class

Collaboration diagram for ImagePtr Class:



Classes

- class [ImagePtr](#)
A reference tracked pointer to an image object.

8.17.1 Detailed Description

8.18 ImageStatistics Class

Collaboration diagram for ImageStatistics Class:



Classes

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

8.18.1 Detailed Description

8.19 Image Utility Class

Collaboration diagram for Image Utility Class:



Classes

- class [ImageUtility](#)
Static helper functions for the image object class.

8.19.1 Detailed Description

8.20 Image Utility Heatmap Class

Collaboration diagram for Image Utility Heatmap Class:



Classes

- class [ImageUtilityHeatmap](#)

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

8.20.1 Detailed Description

8.21 Image Utility Polarization Class

Collaboration diagram for Image Utility Polarization Class:



Classes

- class [ImageUtilityPolarization](#)

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

8.21.1 Detailed Description

8.22 Interface Class

Collaboration diagram for Interface Class:



Classes

- class [Interface](#)

An interface object which holds a list of cameras.

8.22.1 Detailed Description

8.23 InterfaceEvent Class

Collaboration diagram for InterfaceEvent Class:



Classes

- class [InterfaceEvent](#)
A handler to device arrival and removal events on all interfaces.

8.23.1 Detailed Description

8.24 InterfaceList Class

Collaboration diagram for InterfaceList Class:



Classes

- class [InterfaceList](#)

A list of the available interfaces on the system.

8.24.1 Detailed Description

8.25 InterfacePtr Class

Collaboration diagram for InterfacePtr Class:



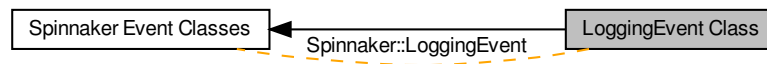
Classes

- class [InterfacePtr](#)
A reference tracked pointer to the interface object.

8.25.1 Detailed Description

8.26 LoggingEvent Class

Collaboration diagram for LoggingEvent Class:



Classes

- class [LoggingEvent](#)
An event handler for capturing the device logging event.

8.26.1 Detailed Description

8.27 Logging Event Class

Collaboration diagram for Logging Event Class:



Classes

- class [LoggingEventData](#)
The [LoggingEventData](#) object.

8.27.1 Detailed Description

8.28 LoggingEventDataPtr Class

Collaboration diagram for LoggingEventDataPtr Class:



Classes

- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.

8.28.1 Detailed Description

8.29 RemovalEvent Class

Collaboration diagram for RemovalEvent Class:



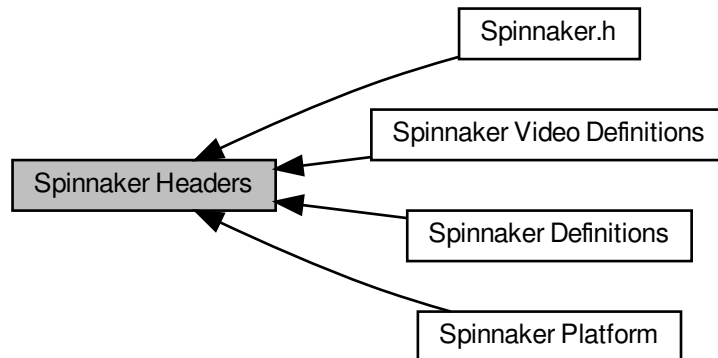
Classes

- class [RemovalEvent](#)
An event handler for capturing the device removal event.

8.29.1 Detailed Description

8.30 Spinnaker Headers

Collaboration diagram for Spinnaker Headers:



Modules

- [Spinnaker.h](#)
Global header file for [Spinnaker](#).
- [Spinnaker Definitions](#)
Definitions file for [Spinnaker](#).
- [Spinnaker Platform](#)
Platform-specific header file for [Spinnaker](#).
- [Spinnaker Video Definitions](#)
Definitions file for [Spinnaker](#) video recorder.

Classes

- struct [MJPGOption](#)
Options for saving MJPG files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [AVIOption](#)
Options for saving AVI files.

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

8.30.1 Detailed Description

8.30.2 Variable Documentation

8.30.2.1 EVENT_TIMEOUT_INFINITE

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

8.30.2.2 EVENT_TIMEOUT_NONE

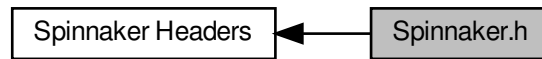
```
const uint64_t EVENT_TIMEOUT_NONE = 0
```

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

8.31 Spinnaker.h

Global header file for [Spinnaker](#).

Collaboration diagram for Spinnaker.h:



Global header file for [Spinnaker](#).

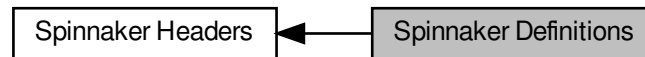
By including this file, all required header files for full [Spinnaker](#) operation will be included automatically. It is recommended that this file be used instead of manually including individual header files.

We welcome your bug reports, suggestions, and comments: <http://www.ptgrey.com/support/contact-us>

8.32 Spinnaker Definitions

Definitions file for [Spinnaker](#).

Collaboration diagram for Spinnaker Definitions:



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.

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 }

```

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 = 0x7FFFFFFF }

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` = 0x8013,  
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,  
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,  
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }  
*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` }

## Functions

- enum `DEPRECATED_CLASS` ("This enum has been deprecated. Polarization images are now created through specific functions the `ImageUtilityPolarization` class.") `PolarizationAlgorithm`
- enum `DEPRECATED_CLASS` ("This enum has been deprecated. `Image` scaling can now be applied through specific functions defined in the `ImageUtility` class.") `PolarizationResolution`
- enum `DEPRECATED_CLASS` ("This enum has been deprecated. Please use `HeatmapColor` in the `ImageUtilityHeatmap` class.") `HeatMapColor`

### 8.32.1 Detailed Description

Definitions file for `Spinnaker`.

### 8.32.2 Enumeration Type Documentation

## 8.32.2.1 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       |  |

## 8.32.2.2 BufferOwnership

enum `BufferOwnership`

## Enumerator

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

## 8.32.2.3 ColorProcessingAlgorithm

enum `ColorProcessingAlgorithm`

Color processing algorithms.

Please refer to our knowledge base at article at <http://www.ptgrey.com/support/kb/index.asp?a=4&q=33> 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.                                                                      |

## 8.32.2.4 Error

enum [Error](#)

[Spinnaker](#) enum definitions.

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

## Enumerator

|                                  |  |
|----------------------------------|--|
| SPINNAKER_ERR_SUCCESS            |  |
| SPINNAKER_ERR_ERROR              |  |
| SPINNAKER_ERR_NOT_INITIALIZED    |  |
| SPINNAKER_ERR_NOT_IMPLEMENTED    |  |
| SPINNAKER_ERR_RESOURCE_IN_USE    |  |
| SPINNAKER_ERR_ACCESS_DENIED      |  |
| SPINNAKER_ERR_INVALID_HANDLE     |  |
| SPINNAKER_ERR_INVALID_ID         |  |
| SPINNAKER_ERR_NO_DATA            |  |
| SPINNAKER_ERR_INVALID_PARAMETER  |  |
| SPINNAKER_ERR_IO                 |  |
| SPINNAKER_ERR_TIMEOUT            |  |
| SPINNAKER_ERR_ABORT              |  |
| SPINNAKER_ERR_INVALID_BUFFER     |  |
| SPINNAKER_ERR_NOT_AVAILABLE      |  |
| SPINNAKER_ERR_INVALID_ADDRESS    |  |
| SPINNAKER_ERR_BUFFER_TOO_SMALL   |  |
| SPINNAKER_ERR_INVALID_INDEX      |  |
| SPINNAKER_ERR_PARSING_CHUNK_DATA |  |
| SPINNAKER_ERR_INVALID_VALUE      |  |
| SPINNAKER_ERR_RESOURCE_EXHAUSTED |  |
| SPINNAKER_ERR_OUT_OF_MEMORY      |  |
| SPINNAKER_ERR_BUSY               |  |
| 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          |  |



## Enumerator

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

## 8.32.2.5 EventType

enum [EventType](#)

[Event](#) types in [Spinnaker](#).

See also

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

## Enumerator

|                                 |  |
|---------------------------------|--|
| SPINNAKER_EVENT_ARRIVAL_REMOVAL |  |
| SPINNAKER_EVENT_DEVICE          |  |
| SPINNAKER_EVENT_DEVICE_SPECIFIC |  |
| SPINNAKER_EVENT_NEW_BUFFER      |  |
| SPINNAKER_EVENT_LOGGING_EVENT   |  |
| SPINNAKER_EVENT_UNKNOWN         |  |

## 8.32.2.6 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.                                 |
| TIFF          | Tagged image file format.                  |

## Enumerator

|                                |                              |
|--------------------------------|------------------------------|
| PNG                            | Portable network graphics.   |
| RAW                            | Raw data.                    |
| JPEG12_C                       | 12 bit compressed JPEG data. |
| IMAGE_FILE_FORMAT_FORCE_32BITS |                              |

## 8.32.2.7 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 <code>GetNextImage()</code> 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.                                                  |
| IMAGE_LEADER_BUFFER_SIZE_INCONSISTENT  | <a href="#">Image</a> leader is incomplete.                                                 |
| IMAGE_TRAILER_BUFFER_SIZE_INCONSISTENT | <a href="#">Image</a> trailer is incomplete.                                                |
| IMAGE_PACKETID_INCONSISTENT            | <a href="#">Image</a> has an inconsistent packet id.                                        |
| IMAGE_MISSING_LEADER                   | <a href="#">Image</a> leader is missing.                                                    |
| IMAGE_MISSING_TRAILER                  | <a href="#">Image</a> trailer is missing.                                                   |
| IMAGE_DATA_INCOMPLETE                  | <a href="#">Image</a> data is incomplete.                                                   |
| IMAGE_INFO_INCONSISTENT                | <a href="#">Image</a> info is corrupted.                                                    |
| 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.                  |

## 8.32.2.8 PayloadTypeInfoIDs

enum [PayloadTypeInfoIDs](#)

## Enumerator

|                         |  |
|-------------------------|--|
| PAYLOAD_TYPE_UNKNOWN    |  |
| PAYLOAD_TYPE_IMAGE      |  |
| PAYLOAD_TYPE_RAW_DATA   |  |
| PAYLOAD_TYPE_FILE       |  |
| PAYLOAD_TYPE_CHUNK_DATA |  |
| PAYLOAD_TYPE_JPEG       |  |

## Enumerator

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

## 8.32.2.9 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 |  |

## 8.32.2.10 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        |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_IIDC       |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_16BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_PFNC_32BIT |  |
| SPINNAKER_PIXELFORMAT_NAMESPACE_CUSTOM_ID  |  |

## 8.32.2.11 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 |  |

## 8.32.2.12 StatisticsChannel

enum [StatisticsChannel](#)

Channels that allow statistics to be calculated.

## Enumerator

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

### 8.32.3 Function Documentation

#### 8.32.3.1 DEPRECATED\_CLASS() [1/3]

```
enum Spinnaker::DEPRECATED_CLASS (
 "This enum has been deprecated. Polarization images are now created through
 specific functions the ImageUtilityPolarization class.")
```

No polarization.

Extracts a Mono8 pixel format image of the 0 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 45 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 90 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format image of the 135 degree of polarization. Polarization value pointer will be null.

Extracts a Mono8 pixel format Stokes' parameter image S0.

Extracts a BGra8 pixel format Heatmap representation of the Stokes' parameter image S0.

Extracts a Mono8 pixel format Stokes' parameter image S1.

Extracts a BGra8 pixel format Heatmap representation of the Stokes' parameter image S1.

Extracts a Mono8 pixel format Stokes' parameter image S2.

Extracts a BGra8 pixel format Heatmap representation of the Stokes' parameter image S2.

Extracts a Mono8 pixel format image representation of the DoLP (Degree of Linear Polarization).

Extracts a BGra8 pixel format Heatmap representation of the DoLP (Degree of Linear Polarization). Resulting polarization values are normalized between 0 and 1.

Extracts a Mono8 pixel format image representation of the AoP (Angle of Polarization).

Extracts a BGra8 pixel format Heatmap representation of the AoP (Angle of Polarization). Resulting polarization values are normalized between 0 and 1.

#### 8.32.3.2 DEPRECATED\_CLASS() [2/3]

```
enum Spinnaker::DEPRECATED_CLASS (
 "This enum has been deprecated. Please use HeatmapColor in the ImageUtility↔Heatmap class.")
```

#### 8.32.3.3 DEPRECATED\_CLASS() [3/3]

```
enum Spinnaker::DEPRECATED_CLASS (
 "This enum has been deprecated. Image scaling can now be applied through specific
 functions defined in the ImageUtility class.")
```

Quarter Resolution.

Full Resolution.

## 8.33 Spinnaker Platform

Platform-specific header file for [Spinnaker](#).

Collaboration diagram for Spinnaker Platform:



### Macros

- `#define SPINNAKER_API_ABSTRACT /*nothing*/`
- `#define SPINNAKER_API __attribute__((visibility ("default")))`
- `#define SPINNAKER_LOCAL __attribute__((visibility ("hidden")))`

#### 8.33.1 Detailed Description

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.

#### 8.33.2 Macro Definition Documentation

##### 8.33.2.1 SPINNAKER\_API

```
#define SPINNAKER_API __attribute__((visibility ("default")))
```

##### 8.33.2.2 SPINNAKER\_API\_ABSTRACT

```
#define SPINNAKER_API_ABSTRACT /*nothing*/
```

##### 8.33.2.3 SPINNAKER\_LOCAL

```
#define SPINNAKER_LOCAL __attribute__((visibility ("hidden")))
```

## 8.34 Spinnaker Video Class

Collaboration diagram for Spinnaker Video Class:



### Classes

- class [SpinVideo](#)

*Provides the functionality for the user to record images to an AVI/MP4 file.*

#### 8.34.1 Detailed Description

## 8.35 Spinnaker Video Definitions

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

Collaboration diagram for Spinnaker Video Definitions:



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



## 8.36 System Class

Collaboration diagram for System Class:



### Classes

- class [System](#)

*The system object is used to retrieve the list of interfaces and cameras available.*

### 8.36.1 Detailed Description

## 8.37 SystemPtr Class

Collaboration diagram for SystemPtr Class:



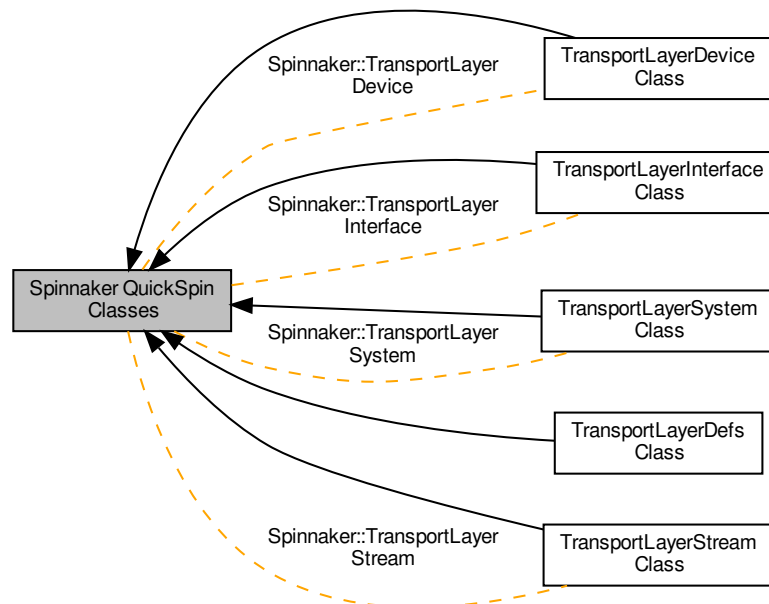
### Classes

- class [SystemPtr](#)  
*A reference tracked pointer to a system object.*

#### 8.37.1 Detailed Description

## 8.38 Spinnaker QuickSpin Classes

Collaboration diagram for Spinnaker QuickSpin Classes:



### Modules

- [TransportLayerDefs Class](#)
- [TransportLayerDevice Class](#)
- [TransportLayerInterface Class](#)
- [TransportLayerStream Class](#)
- [TransportLayerSystem Class](#)

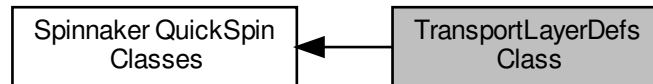
### Classes

- 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.*

### 8.38.1 Detailed Description

## 8.39 TransportLayerDefs Class

Collaboration diagram for TransportLayerDefs Class:



### Enumerations

- enum `StreamTypeEnum` {  
`StreamType_Mixed`,  
`StreamType_Custom`,  
`StreamType_GEV`,  
`StreamType_CL`,  
`StreamType_IIDC`,  
`StreamType_UVC`,  
`StreamType_CXP`,  
`StreamType_CLHS`,  
`StreamType_U3V`,  
`StreamType_ETHERNET`,  
`StreamType_PCI`,  
`NUMSTREAMTYPE` }

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

- enum `StreamDefaultBufferCountModeEnum` {  
`StreamDefaultBufferCountMode_Manual`,  
`StreamDefaultBufferCountMode_Auto`,  
`NUMSTREAMDEFAULTBUFFERCOUNTMODE` }
- enum `StreamBufferCountModeEnum` {  
`StreamBufferCountMode_Manual`,  
`StreamBufferCountMode_Auto`,  
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {  
`StreamBufferHandlingMode_OldestFirst`,  
`StreamBufferHandlingMode_OldestFirstOverwrite`,  
`StreamBufferHandlingMode_NewestFirst`,  
`StreamBufferHandlingMode_NewestFirstOverwrite`,  
`StreamBufferHandlingMode_NewestOnly`,  
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {  
`DeviceType_Mixed`,  
`DeviceType_Custom`,  
`DeviceType_GEV`,  
`DeviceType_CL`,  
`DeviceType_IIDC`,  
`DeviceType_UVC`,  
`DeviceType_CXP`,  
`DeviceType_CLHS`,

```

DeviceType_U3V,
DeviceType_ETHERNET,
DeviceType_PCI,
NUMDEVICETYPE }
• enum DeviceAccessStatusEnum {
DeviceAccessStatus_Unknown,
DeviceAccessStatus_ReadWrite,
DeviceAccessStatus_ReadOnly,
DeviceAccessStatus_NoAccess,
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 DeviceEndiannessMechanismEnum {
DeviceEndiannessMechanism_Legacy,
DeviceEndiannessMechanism_Standard,
NUMDEVICEENDIANESSMECHANISM }
• enum DeviceCurrentSpeedEnum {
DeviceCurrentSpeed_UnknownSpeed,
DeviceCurrentSpeed_LowSpeed,
DeviceCurrentSpeed_FullSpeed,
DeviceCurrentSpeed_HighSpeed,
DeviceCurrentSpeed_SuperSpeed,
NUMDEVICECURRENTSPEED }
• enum POEStatusEnum {
POEStatus_NotSupported,
POEStatus_PowerOff,
POEStatus_PowerOn,
NUMPOESTATUS }
• enum FilterDriverStatusEnum {
FilterDriverStatus_NotSupported,
FilterDriverStatus_Disabled,
FilterDriverStatus_Enabled,
NUMFILTERDRIVERSTATUS }

```

### 8.39.1 Detailed Description

### 8.39.2 Enumeration Type Documentation

#### 8.39.2.1 DeviceAccessStatusEnum

```
enum DeviceAccessStatusEnum
```

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

**Enumerator**

|                              |                       |
|------------------------------|-----------------------|
| DeviceAccessStatus_Unknown   | Unknown status        |
| DeviceAccessStatus_ReadWrite | Full access           |
| DeviceAccessStatus_ReadOnly  | Read-only access      |
| DeviceAccessStatus_NoAccess  | Non-available devices |
| NUMDEVICEACCESSSTATUS        |                       |

**8.39.2.2 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           |                |

**8.39.2.3 DeviceEndiannessMechanismEnum**

enum [DeviceEndiannessMechanismEnum](#)

< Identifies the endianness handling mode.

**Enumerator**

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

**8.39.2.4 DeviceTypeEnum**

enum [DeviceTypeEnum](#)

< Transport layer type of the device.

## Enumerator

|                     |               |
|---------------------|---------------|
| DeviceType_Mixed    | TL - Mixed    |
| DeviceType_Custom   | TL - Custom   |
| DeviceType_GEV      | TL - GEV      |
| DeviceType_CL       | TL - CL       |
| DeviceType_IIDC     | TL - IIDC     |
| DeviceType_UVC      | TL - UVC      |
| DeviceType_CXP      | TL - CXP      |
| DeviceType_CLHS     | TL - CLHS     |
| DeviceType_U3V      | TL - U3V      |
| DeviceType_ETHERNET | TL - ETHERNET |
| DeviceType_PCI      | TL - PCI      |
| NUMDEVICETYPE       |               |

## 8.39.2.5 FilterDriverStatusEnum

enum [FilterDriverStatusEnum](#)

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

## Enumerator

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

## 8.39.2.6 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     |                                              |

### 8.39.2.7 `GevCCPEnum`

enum `GevCCPEnum`

< Controls the device access privilege of an application.

#### Enumerator

|                                                      |                             |
|------------------------------------------------------|-----------------------------|
| <code>GevCCP_EnumEntry_GevCCP_OpenAccess</code>      | Open access privilege.      |
| <code>GevCCP_EnumEntry_GevCCP_ExclusiveAccess</code> | Exclusive access privilege. |
| <code>GevCCP_EnumEntry_GevCCP_ControlAccess</code>   | Control access privilege.   |
| <code>NUMGEVCCP</code>                               |                             |

### 8.39.2.8 `GUIXMLLocationEnum`

enum `GUIXMLLocationEnum`

< Sets the location to load GUI XML.

#### Enumerator

|                                    |                      |
|------------------------------------|----------------------|
| <code>GUIXMLLocation_Device</code> | Load XML from device |
| <code>GUIXMLLocation_Host</code>   | Load XML from host   |
| <code>NUMGUIXMLLOCATION</code>     |                      |

### 8.39.2.9 `POEStatusEnum`

enum `POEStatusEnum`

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

#### Enumerator

|                                     |               |
|-------------------------------------|---------------|
| <code>POEStatus_NotSupported</code> | Not Supported |
| <code>POEStatus_PowerOff</code>     | Power is Off  |
| <code>POEStatus_PowerOn</code>      | Power is On   |
| <code>NUMPOESTATUS</code>           |               |

### 8.39.2.10 `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.                                        |
| StreamBufferCountMode_Auto   | The number of buffers used for the stream is automatically calculated based on the device frame rate. |
| NUMSTREAMBUFFERCOUNTMODE     |                                                                                                       |

#### 8.39.2.11 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_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.                                                                                                                            |
| StreamBufferHandlingMode_NewestFirstOverwrite | DEPRECATED. This is replaced by NewestOnly.                                                                                                                                                                                                                                                                                                                    |
| 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.                           |
| NUMSTREAMBUFFERHANDLINGMODE                   |                                                                                                                                                                                                                                                                                                                                                                |

#### 8.39.2.12 StreamDefaultBufferCountModeEnum

enum [StreamDefaultBufferCountModeEnum](#)

< DEPRECATED; Replaced by StreamBufferCountMode. 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

|                                     |                                                                                    |
|-------------------------------------|------------------------------------------------------------------------------------|
| StreamDefaultBufferCountMode_Manual | DEPRECATED. The number of buffers used for the stream are set by the user.         |
| StreamDefaultBufferCountMode_Auto   | DEPRECATED. The number of buffers used for the stream is automatically calculated. |
| NUMSTREAMDEFAULTBUFFERCOUNTMODE     |                                                                                    |

#### 8.39.2.13 StreamTypeEnum

enum [StreamTypeEnum](#)

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

< Stream type of the device.

#### Enumerator

|                     |                        |
|---------------------|------------------------|
| StreamType_Mixed    | Stream Type - Mixed    |
| StreamType_Custom   | Stream Type - Custom   |
| StreamType_GEV      | Stream Type - GEV      |
| StreamType_CL       | Stream Type - CL       |
| StreamType_IIDC     | Stream Type - IIDC     |
| StreamType_UVC      | Stream Type - UVC      |
| StreamType_CXP      | Stream Type - CXP      |
| StreamType_CLHS     | Stream Type - CLHS     |
| StreamType_U3V      | Stream Type - U3V      |
| StreamType_ETHERNET | Stream Type - ETHERNET |
| StreamType_PCI      | Stream Type - PCI      |
| NUMSTREAMTYPE       |                        |

## 8.40 TransportLayerDevice Class

Collaboration diagram for TransportLayerDevice Class:



### Classes

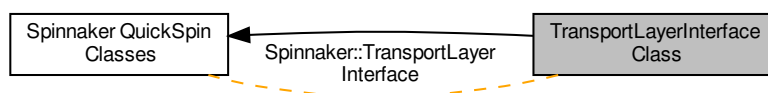
- class [TransportLayerDevice](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.40.1 Detailed Description

## 8.41 TransportLayerInterface Class

Collaboration diagram for TransportLayerInterface Class:



### Classes

- class [TransportLayerInterface](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.41.1 Detailed Description

## 8.42 TransportLayerStream Class

Collaboration diagram for TransportLayerStream Class:



### Classes

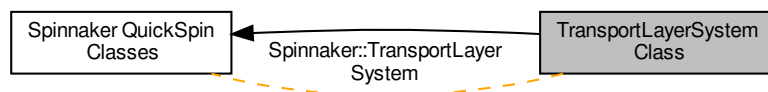
- class [TransportLayerStream](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.42.1 Detailed Description

## 8.43 TransportLayerSystem Class

Collaboration diagram for TransportLayerSystem Class:



### Classes

- class [TransportLayerSystem](#)

*Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.*

### 8.43.1 Detailed Description

## 8.44 Camera Base Interface Class

Collaboration diagram for Camera Base Interface Class:



### Classes

- class [ICameraBase](#)

*The interface file for base class for the camera object.*

### 8.44.1 Detailed Description

## 8.45 IChunkData Class

Collaboration diagram for IChunkData Class:



### Classes

- class [IChunkData](#)  
The [Interface](#) file for [ChunkData](#).

### 8.45.1 Detailed Description



## 8.46 IImage Class

Collaboration diagram for IImage Class:



### Classes

- class [IImage](#)  
*The interface file for [Image](#).*

### 8.46.1 Detailed Description

## 8.47 IImageStatistics Class

Collaboration diagram for IImageStatistics Class:



### Classes

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

### 8.47.1 Detailed Description

## 8.48 IInterface Class

Collaboration diagram for IInterface Class:



### Classes

- class [IInterface](#)

*The interface file for [IInterface](#).*

### 8.48.1 Detailed Description

## 8.49 IInterfaceList Class

Collaboration diagram for IInterfaceList Class:



### Classes

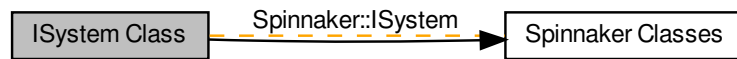
- class [IInterfaceList](#)

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

### 8.49.1 Detailed Description

## 8.50 ISystem Class

Collaboration diagram for ISystem Class:



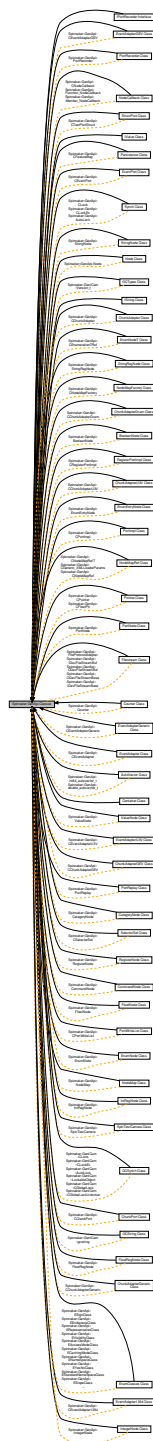
### Classes

- class [ISystem](#)  
*The interface file for [System](#).*

### 8.50.1 Detailed Description

## 8.51 Spinnaker GenApi Classes

Collaboration diagram for Spinnaker GenApi Classes:



### Modules

- [AutoVector Class](#)
- [BooleanNode Class](#)

- [CategoryNode Class](#)
- [ChunkAdapter Class](#)
- [ChunkAdapterDcam Class](#)
- [ChunkAdapterGeneric Class](#)
- [ChunkAdapterGEV Class](#)
- [ChunkPort Class](#)
- [CommandNode Class](#)
- [Container Class](#)
- [Counter Class](#)
- [EnumClasses Class](#)
- [EnumEntryNode Class](#)
- [EnumNode Class](#)
- [EnumNodeT Class](#)
- [EventAdapter Class](#)
- [EventAdapter1394 Class](#)
- [EventAdapterGeneric Class](#)
- [EventAdapterGEV Class](#)
- [EventAdapterU3V Class](#)
- [EventPort Class](#)
- [Filestream Class](#)
- [FloatNode Class](#)
- [FloatRegNode Class](#)
- [GCString Class](#)
- [GCSynch Class](#)
- [GCTypes Class](#)
- [IntegerNode Class](#)
- [IntRegNode Class](#)
- [IString Class](#)
- [IValue Class](#)
- [Node Class](#)
- [NodeCallback Class](#)
- [NodeMap Class](#)
- [NodeMapFactory Class](#)
- [NodeMapRef Class](#)
- [Persistence Class](#)
- [Pointer Class](#)
- [PortImpl Class](#)
- [PortNode Class](#)
- [PortRecorder Class](#)
- [PortReplay Class](#)
- [PortWriteList Class](#)
- [RegisterNode Class](#)
- [RegisterPortImpl Class](#)
- [SelectorSet Class](#)
- [SpinTestCamera Class](#)
- [StringNode Class](#)
- [StringRegNode Class](#)
- [StructPort Class](#)
- [Synch Class](#)
- [ValueNode Class](#)
- [ChunkAdapterU3V Class](#)
- [IPortRecorder Interface](#)

## Classes

- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*
- 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 [CommandNode](#)  
*Interface for string properties.*
- class [Counter](#)  
*Definition of a simple [Counter](#) class.*
- class [ESignClass](#)  
*Holds conversion methods for the sign enumeration.*
- class [EEndianessClass](#)  
*Holds conversion methods for the endianness enumeration.*
- class [ERepresentationClass](#)  
*Holds conversion methods for the representation enumeration.*
- class [EVisibilityClass](#)  
*Holds conversion methods for the visibility enumeration.*
- class [EAccessModeClass](#)  
*Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)  
*Holds conversion methods for the caching mode enumeration.*
- class [ENamespaceClass](#)  
*Holds conversion methods for the namespace enumeration.*
- class [EYesNoClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [EStandardNameSpaceClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [ESlopeClass](#)  
*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*



- class [EnumEntryNode](#)  
*Interface for string properties.*
- class [EnumNode](#)  
*Interface for string properties.*
- class [CEnumerationTRef< EnumT >](#)  
*Interface for string properties.*
- class [CEventAdapter](#)  
*Delivers Events to ports.*
- class [CEventAdapter1394](#)  
*Distribute the events to the node map.*
- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*
- class [CEventAdapterGEV](#)  
*Connects a GigE [Event](#) to a node map.*
- class [CEventAdapterU3V](#)  
*Connects a U3V [Event](#) to a node map.*
- class [CEventPort](#)  
*Port attachable to an event.*
- class [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 >](#)
- class [FloatNode](#)  
*Interface for string properties.*
- class [FloatRegNode](#)  
*Interface for string properties.*
- class [gcstring](#)
- 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 [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*
- struct [Version\\_t](#)  
*Version.*
- class [IntegerNode](#)  
*Interface for string properties.*
- class [IntRegNode](#)  
*Interface for string properties.*
- class [Node](#)  
*class common to all nodes*
- 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.*

  - class [NodeMap](#)
- Smart pointer template for NodeMaps with create function.*

  - class [CNodeMapFactory](#)
- The node map factory is used for creating node maps from camera description files.*

  - 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.*

  - class [CFeatureBag](#)
- Bag holding streamable features of a nodetree.*

  - class [CPointer](#)< T, B >
- Encapsulates a [GenApi](#) pointer dealing with the dynamic\_cast automatically.*

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

  - class [CPortImpl](#)
- Standard implementation for a port.*

  - 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 [CPortWriteList](#)
- Container holding a list of port write commands.*

  - class [RegisterNode](#)
- [Interface](#) for string properties.*

  - class [CRegisterPortImpl](#)
- Standard implementation for a port using a register based transport layer.*

  - class [CSelectorSet](#)
- The set of selectors selecting a given node.*

  - class [SpinTestCamera](#)
- [Interface](#) for string properties.*

  - class [StringNode](#)
- [Interface](#) for string properties.*

  - class [StringRegNode](#)
- [Interface](#) for string properties.*

  - class [CTestPortStruct](#)< CDataStruct >
- Implements a register spaces based on a C++ struct.*

  - class [CLock](#)
- A lock class.*

  - class [CLockEx](#)
- This class is for testing purposes only.*

  - class [AutoLock](#)
- [Interface](#) for value properties.*

  - class [ValueNode](#)

## Typedefs

- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)

## 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](#) ()
- [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*
- [CNodeMapRefT](#) (const [GenICam::gcstring](#) &DeviceName="Device")  
*Constructor.*
- [CNodeMapRefT](#) ([INodeMap](#) \*pNodeMap, const [GenICam::gcstring](#) &DeviceName="Device")  
*Constructor.*
- [CNodeMapRefT](#) (const [CNodeMapRefT](#) &Them)  
*Copy constructor.*

- `CNodeMapRefT` & `operator=` (`INodeMap` \*pNodeMap)  
*Assignment of an INodeMap\*.*
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)  
*Assignment.*
- virtual `~CNodeMapRefT` ()  
*Destructor.*
- void `_Destroy` ()  
*Destroys the node map.*

### 8.51.1 Detailed Description

### 8.51.2 Typedef Documentation

#### 8.51.2.1 CNodeMapRef

```
typedef NodeMap CNodeMapRef
```

#### 8.51.2.2 CNodeRef

```
typedef Node CNodeRef
```

#### 8.51.2.3 CSelectorRef

```
typedef Node CSelectorRef
```

### 8.51.3 Function Documentation

#### 8.51.3.1 \_ClearXMLCache()

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

**8.51.3.2 \_Connect()** [1/2]

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

**8.51.3.3 \_Connect()** [2/2]

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

**8.51.3.4 \_Destroy()**

```
void _Destroy () [inline]
```

Destroys the node map.

**8.51.3.5 \_GetDeviceName()**

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

**8.51.3.6 \_GetNode()**

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

**8.51.3.7 \_GetNodes()**

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

**8.51.3.8 \_GetSupportedSchemaVersions()**

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

#### 8.51.3.9 \_InvalidateNodes()

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

#### 8.51.3.10 \_LoadXMLFromFile()

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

#### 8.51.3.11 \_LoadXMLFromFileInject()

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

#### 8.51.3.12 \_LoadXMLFromString()

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

#### 8.51.3.13 \_LoadXMLFromStringInject()

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

#### 8.51.3.14 \_LoadXMLFromZIPData()

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

#### 8.51.3.15 \_LoadXMLFromZIPFile()

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

**8.51.3.16 \_Poll()**

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

**8.51.3.17 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)

**8.51.3.18 CNodeMapRefT()** [1/3]

```
CNodeMapRefT (
 const GenICam::gcstring & DeviceName = "Device") [inline]
```

Constructor.

**8.51.3.19 CNodeMapRefT()** [2/3]

```
CNodeMapRefT (
 INodeMap * pNodeMap,
 const GenICam::gcstring & DeviceName = "Device") [inline]
```

Constructor.

**8.51.3.20 CNodeMapRefT()** [3/3]

```
CNodeMapRefT (
 const CNodeMapRefT< TCameraParams > & Them)
```

Copy constructor.

**8.51.3.21 EatComments()**

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

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

**8.51.3.22 operator<<()**

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

writes out persistent data to a stream

**8.51.3.23 operator=()** [1/2]

```
CNodeMapRefT< TCameraParams > & operator= (
 const CNodeMapRefT< TCameraParams > & Them)
```

Assignment.

**8.51.3.24 operator=()** [2/2]

```
CNodeMapRefT< TCameraParams > & operator= (
 INodeMap * pNodeMap)
```

Assignment of an INodeMap\*.

**8.51.3.25 operator>>()**

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

Reads in persistent data from a stream.

**8.51.3.26 ~CNodeMapRefT()**

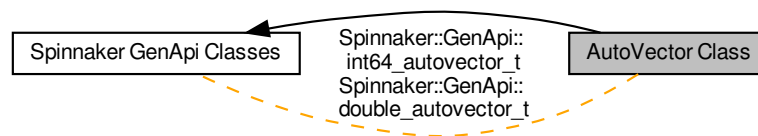
```
~CNodeMapRefT () [inline], [virtual]
```

Destructor.



## 8.52 AutoVector Class

Collaboration diagram for AutoVector Class:



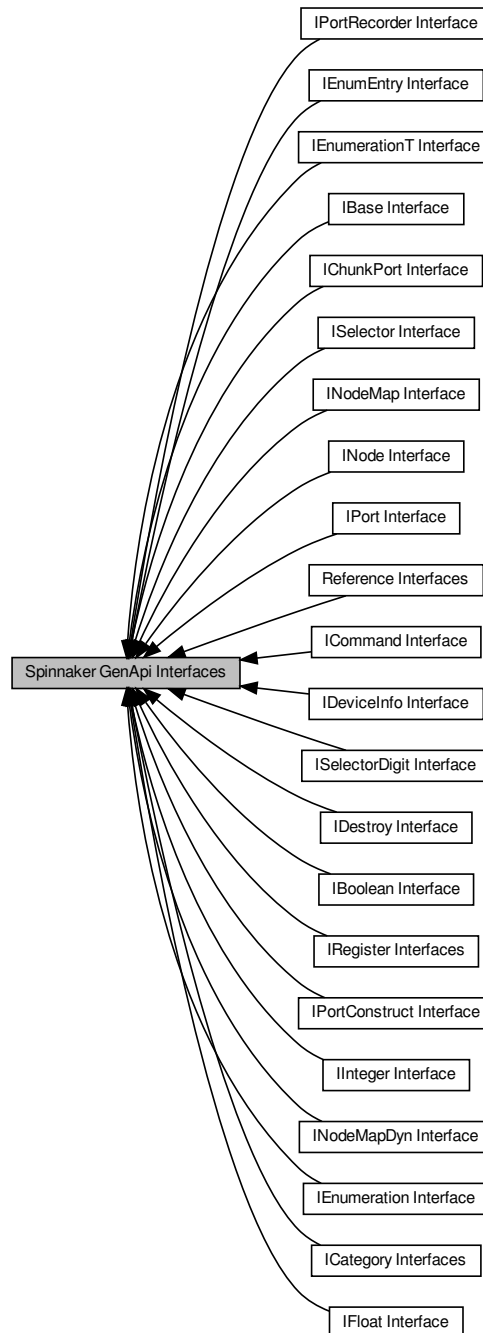
### Classes

- class [int64\\_autovector\\_t](#)  
*Vector of integers with reference counting.*
- class [double\\_autovector\\_t](#)  
*Vector of doubles with reference counting.*

### 8.52.1 Detailed Description

## 8.53 Spinnaker GenApi Interfaces

Collaboration diagram for Spinnaker GenApi Interfaces:



### Modules

- [IBase Interface](#)
- [IBoolean Interface](#)

- [ICategory Interfaces](#)
- [IChunkPort Interface](#)
- [ICommand Interface](#)
- [IDestroy Interface](#)
- [IDeviceInfo Interface](#)
- [IEnumEntry Interface](#)
- [IEnumeration Interface](#)
- [IEnumerationT Interface](#)
- [IFloat Interface](#)
- [IInteger Interface](#)
- [INode Interface](#)
- [INodeMap Interface](#)
- [INodeMapDyn Interface](#)
- [IPort Interface](#)
- [IPortConstruct Interface](#)
- [IPortRecorder Interface](#)
- [IRegister Interfaces](#)
- [ISelector Interface](#)
- [ISelectorDigit Interface](#)
- [Reference Interfaces](#)

## Typedefs

- typedef node\_vector [NodeList\\_t](#)  
*a list of node references*
- typedef intptr\_t [CallbackHandleType](#)  
*the callback handle for nodes*

### 8.53.1 Detailed Description

### 8.53.2 Typedef Documentation

#### 8.53.2.1 CallbackHandleType

```
typedef intptr_t CallbackHandleType
```

the callback handle for nodes

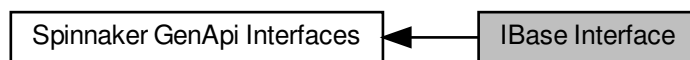
#### 8.53.2.2 NodeList\_t

```
typedef node_vector NodeList_t
```

a list of node references

## 8.54 IBase Interface

Collaboration diagram for IBase Interface:



### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IBase](#)  
*Base interface common to all nodes.*

### 8.54.1 Detailed Description

### 8.54.2 Variable Documentation

#### 8.54.2.1 IBase

```
interface SPINNAKER_API_ABSTRACT IBase
```

#### Initial value:

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

Base interface common to all nodes.

## 8.55 BooleanNode Class

Collaboration diagram for BooleanNode Class:



### Classes

- class [BooleanNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)

#### 8.55.1 Detailed Description

#### 8.55.2 Typedef Documentation

##### 8.55.2.1 CBooleanRef

```
typedef BooleanNode CBooleanRef
```

## 8.56 CategoryNode Class

Collaboration diagram for CategoryNode Class:



### Classes

- class [CategoryNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [CategoryNode](#) [CCategoryRef](#)

#### 8.56.1 Detailed Description

#### 8.56.2 Typedef Documentation

##### 8.56.2.1 CCategoryRef

typedef [CategoryNode](#) [CCategoryRef](#)

## 8.57 ChunkAdapter Class

Collaboration diagram for ChunkAdapter Class:



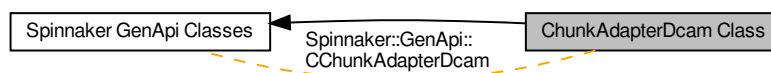
### Classes

- class [CChunkAdapter](#)  
*Connects a chunked buffer to a node map.*

#### 8.57.1 Detailed Description

## 8.58 ChunkAdapterDcam Class

Collaboration diagram for ChunkAdapterDcam Class:



### Classes

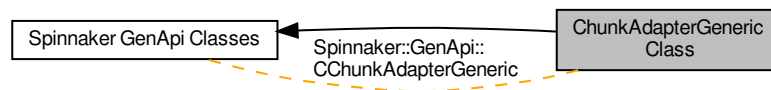
- class [CChunkAdapterDcam](#)  
*Connects a chunked DCAM buffer to a node map.*

#### 8.58.1 Detailed Description



## 8.59 ChunkAdapterGeneric Class

Collaboration diagram for ChunkAdapterGeneric Class:



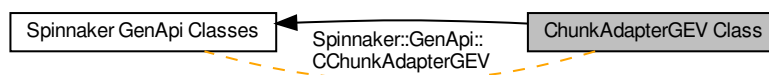
### Classes

- class [CChunkAdapterGeneric](#)

#### 8.59.1 Detailed Description

## 8.60 ChunkAdapterGEV Class

Collaboration diagram for ChunkAdapterGEV Class:



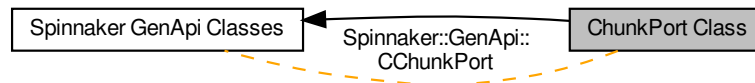
### Classes

- class [CChunkAdapterGEV](#)  
*Connects a chunked DCAM buffer to a node map.*

#### 8.60.1 Detailed Description

## 8.61 ChunkPort Class

Collaboration diagram for ChunkPort Class:



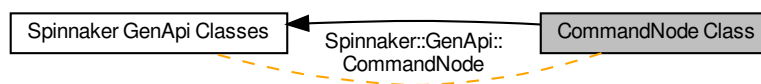
### Classes

- class [CChunkPort](#)  
*Port attachable to a chunk in a buffer.*

### 8.61.1 Detailed Description

## 8.62 CommandNode Class

Collaboration diagram for CommandNode Class:



### Classes

- class [CommandNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [CommandNode](#) [CCommandRef](#)

#### 8.62.1 Detailed Description

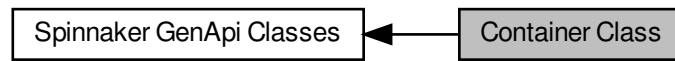
#### 8.62.2 Typedef Documentation

##### 8.62.2.1 CCommandRef

typedef [CommandNode](#) [CCommandRef](#)

## 8.63 Container Class

Collaboration diagram for Container Class:



## 8.64 Counter Class

Collaboration diagram for Counter Class:



### Classes

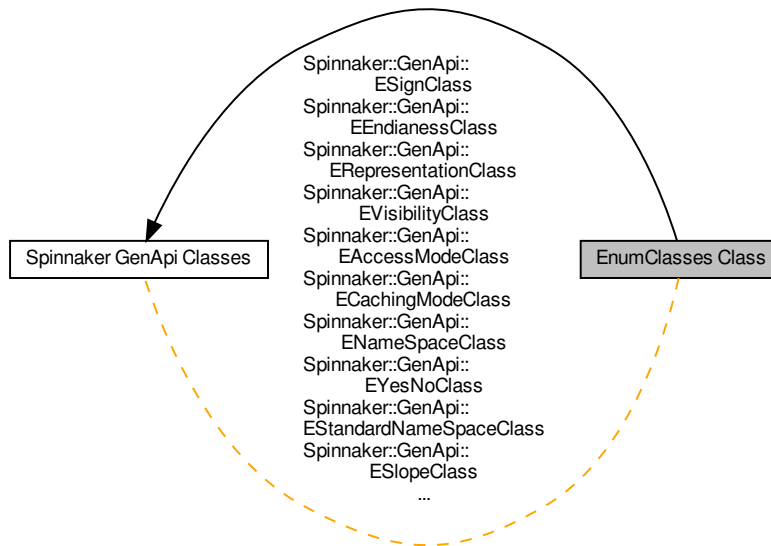
- class [Counter](#)

*Definition of a simple [Counter](#) class.*

### 8.64.1 Detailed Description

## 8.65 EnumClasses Class

Collaboration diagram for EnumClasses Class:



### Classes

- class [ESignClass](#)  
*Holds conversion methods for the sign enumeration.*
- class [EEndiannessClass](#)  
*Holds conversion methods for the endianness enumeration.*
- class [ERepresentationClass](#)  
*Holds conversion methods for the representation enumeration.*
- class [EVisibilityClass](#)  
*Holds conversion methods for the visibility enumeration.*
- class [EAccessModeClass](#)  
*Holds conversion methods for the access mode enumeration.*
- class [ECachingModeClass](#)  
*Holds conversion methods for the caching mode enumeration.*
- class [ENamespaceClass](#)  
*Holds conversion methods for the namespace enumeration.*
- class [EYesNoClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [EStandardNameSpaceClass](#)  
*Holds conversion methods for the standard namespace enumeration.*
- class [ESlopeClass](#)  
*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*

### 8.65.1 Detailed Description



## 8.66 EnumEntryNode Class

Collaboration diagram for EnumEntryNode Class:



### Classes

- class [EnumEntryNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [EnumEntryNode](#) [CEnumEntryRef](#)

#### 8.66.1 Detailed Description

#### 8.66.2 Typedef Documentation

##### 8.66.2.1 CEnumEntryRef

```
typedef EnumEntryNode CEnumEntryRef
```

## 8.67 EnumNode Class

Collaboration diagram for EnumNode Class:



### Classes

- class [EnumNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [EnumNode](#) [CEnumerationRef](#)

#### 8.67.1 Detailed Description

#### 8.67.2 Typedef Documentation

##### 8.67.2.1 CEnumerationRef

typedef [EnumNode](#) [CEnumerationRef](#)

## 8.68 EnumNodeT Class

Collaboration diagram for EnumNodeT Class:



### Classes

- class [CEnumerationTRef](#)< [EnumT](#) >  
*Interface for string properties.*

### 8.68.1 Detailed Description

## 8.69 EventAdapter Class

Collaboration diagram for EventAdapter Class:



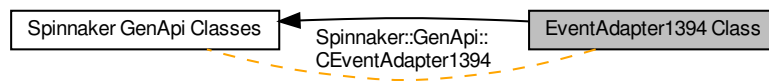
### Classes

- class [CEventAdapter](#)  
*Delivers Events to ports.*

### 8.69.1 Detailed Description

## 8.70 EventAdapter1394 Class

Collaboration diagram for EventAdapter1394 Class:



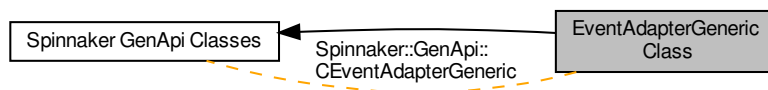
### Classes

- class [CEventAdapter1394](#)  
*Distribute the events to the node map.*

### 8.70.1 Detailed Description

## 8.71 EventAdapterGeneric Class

Collaboration diagram for EventAdapterGeneric Class:



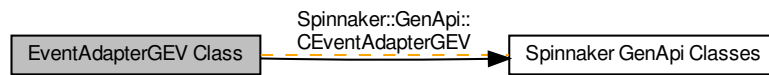
### Classes

- class [CEventAdapterGeneric](#)  
*Connects a generic event to a node map.*

### 8.71.1 Detailed Description

## 8.72 EventAdapterGEV Class

Collaboration diagram for EventAdapterGEV Class:



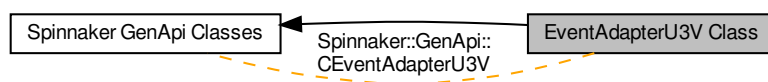
### Classes

- class [CEventAdapterGEV](#)  
*Connects a GigE [Event](#) to a node map.*

### 8.72.1 Detailed Description

## 8.73 EventAdapterU3V Class

Collaboration diagram for EventAdapterU3V Class:



### Classes

- class [CEventAdapterU3V](#)  
*Connects a U3V [Event](#) to a node map.*

#### 8.73.1 Detailed Description



## 8.74 EventPort Class

Collaboration diagram for EventPort Class:



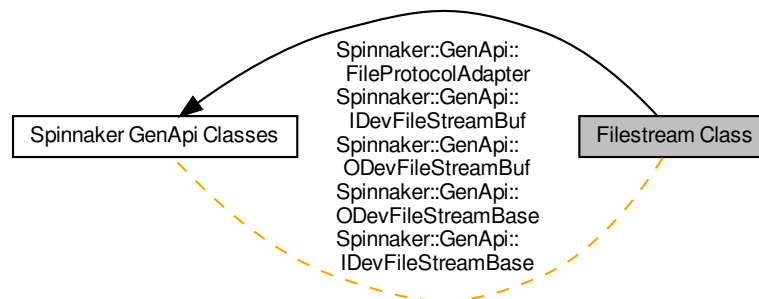
### Classes

- class [CEventPort](#)  
*Port attachable to an event.*

### 8.74.1 Detailed Description

## 8.75 Filestream Class

Collaboration diagram for Filestream Class:



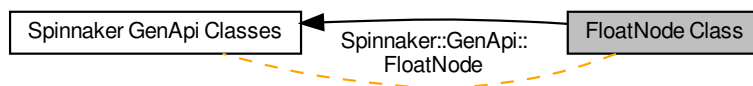
### Classes

- 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 >](#)

### 8.75.1 Detailed Description

## 8.76 FloatNode Class

Collaboration diagram for FloatNode Class:



### Classes

- class [FloatNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [FloatNode](#) [CFloatRef](#)

#### 8.76.1 Detailed Description

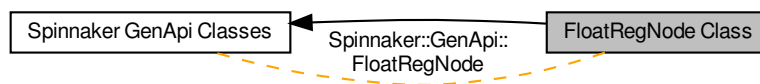
#### 8.76.2 Typedef Documentation

##### 8.76.2.1 CFloatRef

```
typedef FloatNode CFloatRef
```

## 8.77 FloatRegNode Class

Collaboration diagram for FloatRegNode Class:



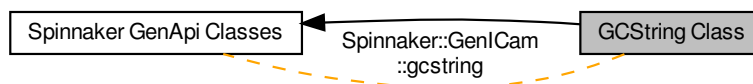
### Classes

- class [FloatRegNode](#)  
*Interface for string properties.*

### 8.77.1 Detailed Description

## 8.78 GCString Class

Collaboration diagram for GCString Class:



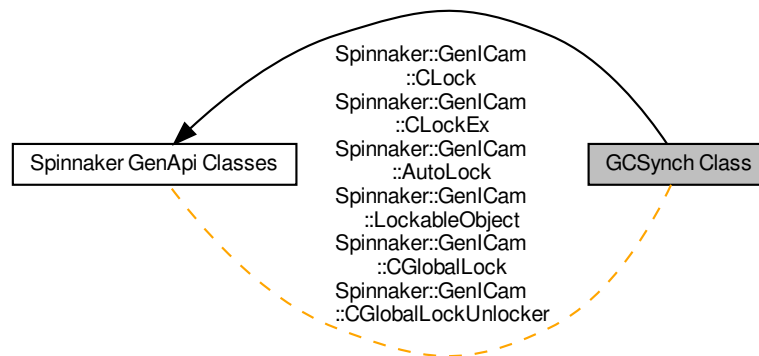
### Classes

- class [gcstring](#)

### 8.78.1 Detailed Description

## 8.79 GCSynch Class

Collaboration diagram for GCSynch Class:



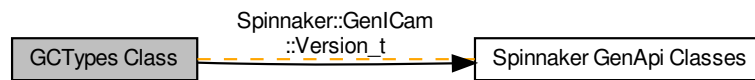
### Classes

- 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 [CGlobalLock](#)  
*Named global lock which can be used over process boundaries.*
- class [CGlobalLockUnlocker](#)  
*Unlocks the global lock object on destruction.*

### 8.79.1 Detailed Description

## 8.80 GTypes Class

Collaboration diagram for GTypes Class:



### Classes

- struct [Version\\_t](#)  
*Version.*

### Typedefs

- typedef float [float32\\_t](#)  
*32 bit floating point*
- typedef double [float64\\_t](#)  
*64 bit floating point*

#### 8.80.1 Detailed Description

#### 8.80.2 Typedef Documentation

##### 8.80.2.1 [float32\\_t](#)

```
typedef float float32_t
```

32 bit floating point

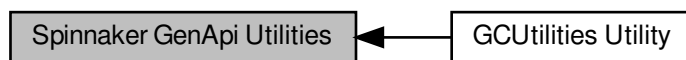
##### 8.80.2.2 [float64\\_t](#)

```
typedef double float64_t
```

64 bit floating point

## 8.81 Spinnaker GenApi Utilities

Collaboration diagram for Spinnaker GenApi Utilities:



### Modules

- [GCUtilities Utility](#)

### 8.81.1 Detailed Description



## 8.82 GCUtilities Utility

Collaboration diagram for GCUtilities Utility:



### 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.*

### 8.82.1 Detailed Description

### 8.82.2 Function Documentation

#### 8.82.2.1 DoesEnvironmentVariableExist()

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

Returns true if an environment variable exists.

#### 8.82.2.2 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            |

#### 8.82.2.3 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.

#### 8.82.2.4 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.

#### 8.82.2.5 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.

#### 8.82.2.6 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

#### 8.82.2.7 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 |
|--------------------------|--------------|

## 8.82.2.8 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

## 8.82.2.9 INTEGRAL\_CAST()

```
T Spinnaker::GenICam::INTEGRAL_CAST (
 int64_t II) [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)

## 8.82.2.10 INTEGRAL\_CAST2()

```
Td Spinnaker::GenICam::INTEGRAL_CAST2 (
 Ts s) [inline]
```

This verifies at runtime if there was no loss of data if an type Ts (e.g.

int64t) was downcast to type Td (e.g. int32\_t)

## 8.82.2.11 ReplaceEnvironmentVariables()

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

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

#### 8.82.2.12 SetGenICamCacheFolder()

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

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

#### 8.82.2.13 SetGenICamCLProtocolFolder()

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

Stores the path of the CLProtocol folder.

#### 8.82.2.14 SetGenICamLogConfig()

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

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

#### 8.82.2.15 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      |

#### 8.82.2.16 UriDecode()

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

Replaces xx escapes by their char equivalent.

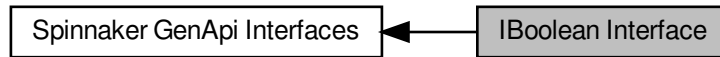
#### 8.82.2.17 UrlEncode()

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

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

## 8.83 IBoolean Interface

Collaboration diagram for IBoolean Interface:



### 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

#### 8.83.1 Detailed Description

#### 8.83.2 Function Documentation

##### 8.83.2.1 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

#### 8.83.2.2 operator()()

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

Get node value.

Execute the command.

#### 8.83.2.3 operator=()

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

Set node value.

### 8.83.3 Variable Documentation

#### 8.83.3.1 IBoolean

```
interface SPINNAKER_API_ABSTRACT IBoolean
```

[Interface](#) for Boolean properties.

#### 8.83.3.2 Verify

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



## 8.84 ICategory Interfaces

Collaboration diagram for ICategory Interfaces:



### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT ICategory](#)  
*Gives access to a category node.*

#### 8.84.1 Detailed Description

#### 8.84.2 Variable Documentation

##### 8.84.2.1 ICategory

```
interface SPINNAKER_API_ABSTRACT ICategory
```

Gives access to a category node.

## 8.85 IChunkPort Interface

Collaboration diagram for IChunkPort Interface:



### 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.*

### 8.85.1 Detailed Description

### 8.85.2 Macro Definition Documentation

#### 8.85.2.1 [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`)

### 8.85.2.2 CHUNK\_BASE\_ADDRESS\_REGISTER\_LEN

```
#define CHUNK_BASE_ADDRESS_REGISTER_LEN 8
```

Length of the CHUNK\_BASE\_ADDRESS\_REGISTER pseudo register.

### 8.85.2.3 CHUNK\_LENGTH\_REGISTER

```
#define CHUNK_LENGTH_REGISTER (GC_INT64_MAX-15)
```

Address of a int64\_t pseudo register containing the length of the chunk.

### 8.85.2.4 CHUNK\_LENGTH\_REGISTER\_LEN

```
#define CHUNK_LENGTH_REGISTER_LEN 8
```

Length of the CHUNK\_LENGTH\_REGISTER pseudo register.

## 8.85.3 Function Documentation

### 8.85.3.1 CacheChunkData()

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

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

## 8.85.4 Variable Documentation

### 8.85.4.1 IChunkPort

```
interface SPINNAKER_API_ABSTRACT IChunkPort
```

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

## 8.86 ICommand Interface

Collaboration diagram for ICommand Interface:



### Functions

- virtual bool `IsDone` (bool `Verify`=true)=0  
*Query whether the command is executed.*

### Variables

- `interface SPINNAKER_API_ABSTRACT ICommand`  
*Interface for command like properties.*

#### 8.86.1 Detailed Description

#### 8.86.2 Function Documentation

##### 8.86.2.1 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

#### 8.86.3 Variable Documentation

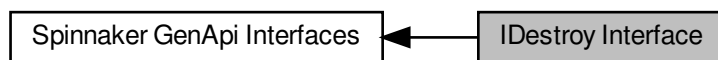
### 8.86.3.1 ICommand

```
interface SPINNAKER_API_ABSTRACT ICommand
```

**Interface** for command like properties.

## 8.87 IDestroy Interface

Collaboration diagram for IDestroy Interface:



### Variables

- `interface SPINNAKER_API_ABSTRACT IDestroy`  
*Interface to destroy an object.*

### 8.87.1 Detailed Description

### 8.87.2 Variable Documentation

#### 8.87.2.1 IDestroy

```
interface SPINNAKER_API_ABSTRACT IDestroy
```

#### Initial value:

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

*Interface to destroy an object.*

## 8.88 IDeviceInfo Interface

Collaboration diagram for IDeviceInfo Interface:



### 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)*

#### 8.88.1 Detailed Description

#### 8.88.2 Function Documentation

##### 8.88.2.1 GetDeviceVersion()

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

Get the version of the device description file.

#### 8.88.2.2 GetGenApiVersion()

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

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

#### 8.88.2.3 GetProductGuid()

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

Get the Guid describing the product.

#### 8.88.2.4 GetSchemaVersion()

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

Get the schema version number.

#### 8.88.2.5 GetStandardNameSpace()

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

Get the standard name space.

#### 8.88.2.6 GetToolTip()

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

Get tool tip.

Get a short description of the node.

#### 8.88.2.7 GetVendorName()

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

Get the vendor name.



### 8.88.2.8 GetVersionGuid()

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

Get the Guid describing the product version.

## 8.88.3 Variable Documentation

### 8.88.3.1 IDeviceInfo

```
interface SPINNAKER_API_ABSTRACT IDeviceInfo
```

**Initial value:**

```
{

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

[Interface](#) to get information about the device (= nodemap)

## 8.89 IEnumEntry Interface

Collaboration diagram for IEnumEntry Interface:



### 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.*

### 8.89.1 Detailed Description

### 8.89.2 Function Documentation

#### 8.89.2.1 GetNumericValue()

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

Get double number associated with the entry.

#### 8.89.2.2 GetSymbolic()

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

Get symbolic enum value.

### 8.89.2.3 IsSelfClearing()

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

Indicates if the corresponding EnumEntry is self clearing.

## 8.89.3 Variable Documentation

### 8.89.3.1 IEnumEntry

```
interface SPINNAKER_API_ABSTRACT IEnumEntry
```

**Interface** of single enum value.

Maps of Enum Values to symbolic values

## 8.90 IEnumeration Interface

Collaboration diagram for IEnumeration Interface:



### Functions

- virtual void [GetEntries](#) ([NodeList\\_t](#) &Entries)=0  
*Get list of entry nodes.*
- 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.*

### 8.90.1 Detailed Description

### 8.90.2 Function Documentation

#### 8.90.2.1 GetCurrentEntry()

```

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

```

Get the current entry.

### 8.90.2.2 GetEntries()

```
virtual void Spinnaker::GenApi::GetEntries (
 NodeList_t & Entries) [pure virtual]
```

Get list of entry nodes.

### 8.90.2.3 GetEntry()

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

Get an entry node by its IntValue.

### 8.90.2.4 GetEntryByName()

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

Get an entry node by name.

### 8.90.2.5 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

### 8.90.2.6 operator\*()

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

Get string node value.

Get node value.

#### 8.90.2.7 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) |

### 8.90.3 Variable Documentation

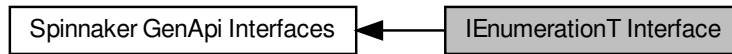
#### 8.90.3.1 IEnumeration

```
interface SPINNAKER_API_ABSTRACT IEnumeration
```

[Interface](#) for enumeration properties.

## 8.91 IEnumerationT Interface

Collaboration diagram for IEnumerationT Interface:



### Functions

- 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 [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)=0  
*Set string node value.*

### Variables

- template<typename EnumT >  
[interface SPINNAKER\\_API\\_ABSTRACT IEnumerationT](#)  
*Interface for enumeration properties.*
- template<typename EnumT >  
[interface SPINNAKER\\_API\\_ABSTRACT](#) virtual public [IEnumReference](#)  
*Interface to construct an enum reference.*

#### 8.91.1 Detailed Description

#### 8.91.2 Function Documentation

##### 8.91.2.1 GetEntry()

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

returns the EnumEntry object belonging to the Value

**8.91.2.2 operator=()** [1/2]

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

Set node value.

**8.91.2.3 operator=()** [2/2]

```
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

**8.91.3 Variable Documentation****8.91.3.1 IEnumerationT**

```
interface SPINNAKER_API_ABSTRACT IEnumerationT
```

Interface for enumeration properties.

**8.91.3.2 IEnumReference**

```
interface SPINNAKER_API_ABSTRACT IEnumReference
```

**Initial value:**

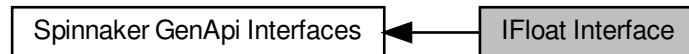
```
{
 virtual void SetValue(EnumT Value, bool Verify = true) = 0
```

Interface to construct an enum reference.



## 8.92 IFloat Interface

Collaboration diagram for IFloat Interface:



### Functions

- virtual [IFloat & operator=](#) (double Value)=0  
*Set node value.*
- virtual double [GetMin](#) ()=0  
*Get minimum value allowed.*
- virtual double [GetMax](#) ()=0  
*Get maximum value allowed.*
- virtual bool [HasInc](#) ()=0  
*True if the float has a constant increment.*
- virtual [EIncMode GetIncMode](#) ()=0  
*Get increment mode.*
- virtual double [GetInc](#) ()=0  
*Get the constant increment if there is any.*
- virtual [double\\_autovector\\_t GetListOfValidValues](#) (bool bounded=true)=0  
*Get list of valid value.*
- virtual [ERepresentation GetRepresentation](#) ()=0  
*Get recommended representation.*
- virtual [GenICam::gcstring GetUnit](#) () const =0  
*Get the physical unit name.*
- virtual [EDisplayNotation GetDisplayNotation](#) () const =0  
*Get the way the float should be converted to a string.*
- virtual [int64\\_t GetDisplayPrecision](#) () const =0  
*Get the precision to be used when converting the float to a string.*
- virtual void [ImposeMin](#) (double Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (double Value)=0  
*Restrict maximum value.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IFloat](#)  
*Interface for float properties.*

### 8.92.1 Detailed Description

### 8.92.2 Function Documentation

#### 8.92.2.1 GetDisplayNotation()

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

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

#### 8.92.2.2 GetDisplayPrecision()

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

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

#### 8.92.2.3 GetInc()

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

Get the constant increment if there is any.

Get increment.

#### 8.92.2.4 GetIncMode()

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

Get increment mode.

#### 8.92.2.5 GetListOfValidValues()

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

Get list of valid value.

**8.92.2.6 GetMax()**

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

Get maximum value allowed.

**8.92.2.7 GetMin()**

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

Get minimum value allowed.

**8.92.2.8 GetRepresentation()**

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

Get recommended representation.

**8.92.2.9 GetUnit()**

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

Get the physical unit name.

**8.92.2.10 HasInc()**

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

True if the float has a constant increment.

**8.92.2.11 ImposeMax()**

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

Restrict maximum value.

#### 8.92.2.12 `ImposeMin()`

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

Restrict minimum value.

#### 8.92.2.13 `operator=()`

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

Set node value.

### 8.92.3 Variable Documentation

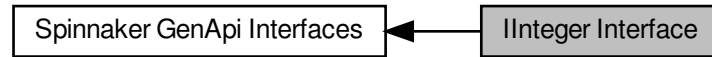
#### 8.92.3.1 `IFloat`

```
interface SPINNAKER_API_ABSTRACT IFloat
```

[Interface](#) for float properties.

## 8.93 Integer Interface

Collaboration diagram for Integer Interface:



### Functions

- virtual [Integer](#) & [operator=](#) (int64\_t Value)=0  
*Set node value.*
- virtual void [ImposeMin](#) (int64\_t Value)=0  
*Restrict minimum value.*
- virtual void [ImposeMax](#) (int64\_t Value)=0  
*Restrict maximum value.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT Integer](#)  
*Interface for integer properties.*

#### 8.93.1 Detailed Description

#### 8.93.2 Function Documentation

##### 8.93.2.1 ImposeMax()

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

Restrict maximum value.

##### 8.93.2.2 ImposeMin()

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

Restrict minimum value.

### 8.93.2.3 operator=()

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

Set node value.

## 8.93.3 Variable Documentation

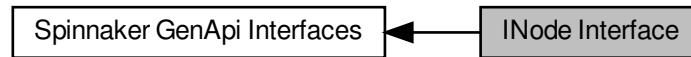
### 8.93.3.1 IInteger

```
interface SPINNAKER_API_ABSTRACT IInteger
```

[Interface](#) for integer properties.

## 8.94 INode Interface

Collaboration diagram for INode Interface:



### Functions

- virtual [GenApi::ENamespace GetNamespace \(\)](#) const =0  
*Get name space.*
- virtual [EVisibility GetVisibility \(\)](#) const =0  
*Get the recommended visibility of the node.*
- virtual void [InvalidateNode \(\)](#)=0  
*Indicates that the node's value may have changed.*
- virtual bool [IsCacheable \(\)](#) const =0  
*Is the node value cacheable.*
- virtual [EYesNo IsAccessModeCacheable \(\)](#) const =0  
*True if the AccessMode can be cached.*
- virtual [ECachingMode GetCachingMode \(\)](#) const =0  
*Get Caching Mode.*
- virtual [int64\\_t GetPollingTime \(\)](#) const =0  
*recommended polling time (for non-cacheable nodes)*
- virtual [GenICam::gcstring GetDescription \(\)](#) const =0  
*Get a long description of the node.*
- virtual [GenICam::gcstring GetDisplayName \(\)](#) const =0  
*Get a name string for display.*
- 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 `GenlCam::gcstring` &PropertyName, `GenlCam::gcstring` &ValueStr, `GenlCam::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 `GenlCam::gcstring` `GetDocuURL` () const =0  
*Gets a URL pointing to the documentation of that feature.*
- virtual bool `IsDeprecated` () const =0  
*True if the node should not be used any more.*
- virtual `EInterfaceType` `GetPrincipalInterfaceType` () const =0  
*Get the type of the main interface of a node.*
- virtual bool `IsFeature` () const =0  
*True if the node can be reached via category nodes from a category node named "Root".*
- virtual bool `operator==` (int nullPtr) const =0
- virtual bool `operator!=` (int nullPtr) const =0
- bool `IsReadable` (`EAccessMode` AccessMode)  
*Tests if readable.*
- bool `IsReadable` (const `IBase *`p)  
*Checks if a node is readable.*
- bool `IsReadable` (const `IBase` &r)  
*Checks if a node is readable.*
- bool `IsWritable` (`EAccessMode` AccessMode)  
*Tests if writable.*
- bool `IsWritable` (const `IBase *`p)  
*Checks if a node is writable.*
- bool `IsWritable` (const `IBase` &r)  
*Checks if a node is writable.*
- bool `IsImplemented` (`EAccessMode` AccessMode)  
*Tests if implemented.*
- bool `IsImplemented` (const `IBase *`p)  
*Checks if a node is implemented.*
- bool `IsImplemented` (const `IBase` &r)  
*Checks if a node is implemented.*
- bool `IsAvailable` (`EAccessMode` AccessMode)  
*Tests if available.*
- bool `IsAvailable` (const `IBase *`p)  
*Checks if a node is available.*
- bool `IsAvailable` (const `IBase` &r)  
*Checks if a node is available.*
- `EAccessMode` `Combine` (`EAccessMode` Peter, `EAccessMode` Paul)  
*Computes which access mode the two guards allow together.*
- bool `IsVisible` (`EVisibility` Visibility, `EVisibility` MaxVisibility)  
*Tests Visibility CAVE : this relies on the EVisibility enum's coding.*
- `EVisibility` `Combine` (`EVisibility` Peter, `EVisibility` Paul)



*Computes which visibility the two guards allow together.*

- `bool IsCacheable (ECachingMode CachingMode)`  
*Tests Cacheability.*
- `ECachingMode Combine (ECachingMode Peter, ECachingMode Paul)`  
*Computes which CachingMode results from a combination.*

## Variables

- `interface SPINNAKER_API_ABSTRACT INode`  
*Interface common to all nodes.*
- `interface SPINNAKER_API_ABSTRACT virtual public IReference`  
*Interface to construct a reference.*

## 8.94.1 Detailed Description

## 8.94.2 Function Documentation

### 8.94.2.1 Combine() [1/3]

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

Computes which access mode the two guards allow together.

### 8.94.2.2 Combine() [2/3]

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

Computes which visibility the two guards allow together.

### 8.94.2.3 Combine() [3/3]

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

Computes which CachingMode results from a combination.

#### 8.94.2.4 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

#### 8.94.2.5 GetAlias()

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

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

#### 8.94.2.6 GetCachingMode()

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

Get Caching Mode.

#### 8.94.2.7 GetCastAlias()

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

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

#### 8.94.2.8 GetChildren()

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

Get all nodes this node directly depends on.

##### Parameters

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

#### 8.94.2.9 GetDescription()

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

Get a long description of the node.

#### 8.94.2.10 GetDisplayName()

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

Get a name string for display.

#### 8.94.2.11 GetDocuURL()

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

Gets a URL pointing to the documentation of that feature.

#### 8.94.2.12 GetEventID()

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

Get the EventId of the node.

#### 8.94.2.13 GetNameSpace()

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

Get name space.

#### 8.94.2.14 GetNodeMap()

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

Retrieves the central node map.

#### 8.94.2.15 GetParents()

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

Gets all nodes this node is directly depending on.

## Parameters

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

**8.94.2.16 GetPollingTime()**

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

recommended polling time (for non-cacheable nodes)

**8.94.2.17 GetPrincipalInterfaceType()**

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

Get the type of the main interface of a node.

**8.94.2.18 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.

**8.94.2.19 GetPropertyNames()**

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

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

**8.94.2.20 GetVisibility()**

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

Get the recommended visibility of the node.

**8.94.2.21** `ImposeAccessMode()`

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

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

**8.94.2.22** `ImposeVisibility()`

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

Imposes a visibility to the natural visibility of the node.

**8.94.2.23** `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

**8.94.2.24** `IsAccessModeCacheable()`

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

True if the AccessMode can be cached.

**8.94.2.25** `IsAvailable()` [1/3]

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

Tests if available.

**8.94.2.26** `IsAvailable()` [2/3]

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

Checks if a node is available.

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

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

Checks if a node is available.

**8.94.2.28 IsCacheable()**

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

Is the node value cacheable.

**8.94.2.29 IsCacheable()**

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

Tests Cacheability.

**8.94.2.30 IsDeprecated()**

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

True if the node should not be used any more.

**8.94.2.31 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".

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

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

Tests if implemented.

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

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

Checks if a node is implemented.

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

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

Checks if a node is implemented.

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

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

Tests if readable.

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

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

Checks if a node is readable.

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

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

Checks if a node is readable.

**8.94.2.38 IsStreamable()**

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

True if the node is streamable.

#### 8.94.2.39 isVisible()

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

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

#### 8.94.2.40 IsWritable() [1/3]

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

Tests if writable.

#### 8.94.2.41 IsWritable() [2/3]

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

Checks if a node is writable.

#### 8.94.2.42 IsWritable() [3/3]

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

Checks if a node is writable.

#### 8.94.2.43 operator!=(())

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

#### 8.94.2.44 operator==(())

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



#### 8.94.2.45 RegisterCallback()

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

Register change callback Takes ownership of the [CNodeCallback](#) object.

### 8.94.3 Variable Documentation

#### 8.94.3.1 INode

```
interface SPINNAKER_API_ABSTRACT INode
```

[Interface](#) common to all nodes.

#### 8.94.3.2 IReference

```
interface SPINNAKER_API_ABSTRACT IReference
```

##### Initial value:

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

[Interface](#) to construct a reference.

## 8.95 INodeMap Interface

Collaboration diagram for INodeMap Interface:



### 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 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 [GenICam::gcstring](#) [GetDeviceName](#) () const =0  
*Get a name of the device.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT INodeMap](#)  
*Interface to access the node map.*

#### 8.95.1 Detailed Description

#### 8.95.2 Function Documentation

**8.95.2.1 Connect()** [1/2]

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

Connects a port to a port node with given name.

**8.95.2.2 Connect()** [2/2]

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

Connects a port to the standard port "Device".

**8.95.2.3 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".

**8.95.2.4 GetLock()**

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

Returns the lock which guards the node map.

**8.95.2.5 GetNode()**

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

Retrieves the node from the central map by Name.

#### 8.95.2.6 GetNumNodes()

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

Get the number of nodes in the map.

#### 8.95.2.7 InvalidateNodes()

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

Invalidates all nodes.

#### 8.95.2.8 Poll()

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

Fires nodes which have a polling time.

### 8.95.3 Variable Documentation

#### 8.95.3.1 INodeMap

```
interface SPINNAKER_API_ABSTRACT INodeMap
```

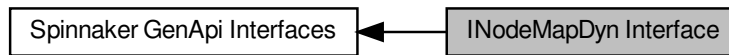
**Initial value:**

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

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

## 8.96 INodeMapDyn Interface

Collaboration diagram for INodeMapDyn Interface:



### 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=[xv↔Default](#))=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](#) &Extracted↔Subtree)=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=[xv↔Default](#))=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.*

## 8.96.1 Detailed Description

## 8.96.2 Function Documentation

### 8.96.2.1 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. |

### 8.96.2.2 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

### 8.96.2.3 LoadXMLFromFile()

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

Loads an XML from a file.

#### 8.96.2.4 LoadXMLFromFileInject()

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

Loads an XML from a file with injection.

#### 8.96.2.5 LoadXMLFromString()

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

Loads an XML from a string.

#### 8.96.2.6 LoadXMLFromStringInject()

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

Loads an XML from a string with injection.

#### 8.96.2.7 LoadXMLFromZIPData()

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

Loads an XML from a ZIP data buffer.

#### 8.96.2.8 LoadXMLFromZIPFile()

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

Loads an XML from a ZIP file.

#### 8.96.2.9 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 oputput file |

## 8.96.2.10 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>XML Validation</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 |

## 8.96.2.11 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 |

### 8.96.3 Variable Documentation

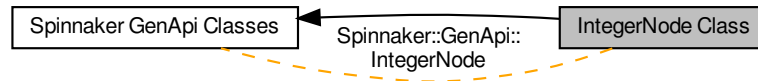
#### 8.96.3.1 INodeMapDyn

```
interface SPINNAKER_API_ABSTRACT INodeMapDyn
```

Interface to access the node map.

## 8.97 IntegerNode Class

Collaboration diagram for IntegerNode Class:



### Classes

- class [IntegerNode](#)  
*[Interface](#) for string properties.*

### Typedefs

- typedef [IntegerNode](#) [CIntegerRef](#)

#### 8.97.1 Detailed Description

#### 8.97.2 Typedef Documentation

##### 8.97.2.1 CIntegerRef

typedef [IntegerNode](#) [CIntegerRef](#)

## 8.98 IntRegNode Class

Collaboration diagram for IntRegNode Class:



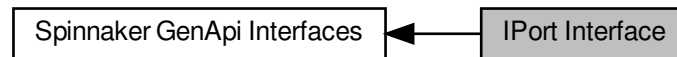
### Classes

- class [IntRegNode](#)  
*Interface for string properties.*

### 8.98.1 Detailed Description

## 8.99 IPort Interface

Collaboration diagram for IPort Interface:



### 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](#)

### 8.99.1 Detailed Description

### 8.99.2 Function Documentation

#### 8.99.2.1 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.

### 8.99.3 Variable Documentation

### 8.99.3.1 Address

```
interface SPINNAKER_API_ABSTRACT int64_t Address
```

### 8.99.3.2 IPort

```
interface SPINNAKER_API_ABSTRACT IPort
```

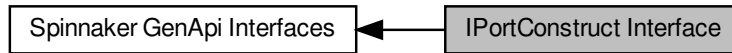
Interface for ports.

### 8.99.3.3 Length

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

## 8.100 IPortConstruct Interface

Collaboration diagram for IPortConstruct Interface:



### Functions

- virtual [EYesNo GetSwapEndianness \(\)](#)=0  
*Determines if the port adapter must perform an endianness swap.*

### Variables

- [interface SPINNAKER\\_API IPortConstruct](#)  
*Interface for ports.*

### 8.100.1 Detailed Description

### 8.100.2 Function Documentation

#### 8.100.2.1 GetSwapEndianness()

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

Determines if the port adapter must perform an endianness swap.

### 8.100.3 Variable Documentation

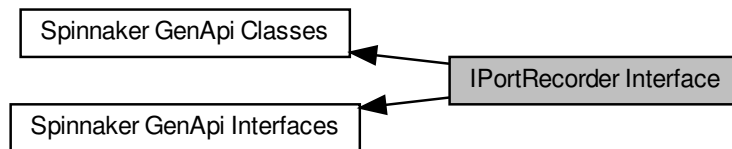
#### 8.100.3.1 IPortConstruct

```
interface SPINNAKER_API IPortConstruct
```

[Interface](#) for ports.

## 8.101 IPortRecorder Interface

Collaboration diagram for IPortRecorder Interface:



### 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.*

### 8.101.1 Detailed Description

### 8.101.2 Function Documentation

#### 8.101.2.1 GetCookie()

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

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

#### 8.101.2.2 Replay()

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

Replays the write command to the given port interface.

#### 8.101.2.3 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.

#### 8.101.2.4 StopRecording()

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

Stops recording.

### 8.101.3 Variable Documentation

#### 8.101.3.1 Invalidate

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

#### 8.101.3.2 IPortRecorder

```
interface SPINNAKER_API_ABSTRACT IPortRecorder
```

**Interface** for recording write commands on a port.

#### 8.101.3.3 IPortReplay

```
interface SPINNAKER_API_ABSTRACT IPortReplay
```

**Interface** for replaying write commands on a port.

#### 8.101.3.4 IPortWriteList

```
interface SPINNAKER_API_ABSTRACT IPortWriteList
```

**Initial value:**

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



## 8.102 IRegister Interfaces

Collaboration diagram for IRegister Interfaces:



### 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.*

### 8.102.1 Detailed Description

### 8.102.2 Function Documentation

#### 8.102.2.1 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

**8.102.2.2    GetAddress()**

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

Retrieves the Address of the register.

**8.102.2.3    GetLength()**

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

Retrieves the Length of the register [Bytes].

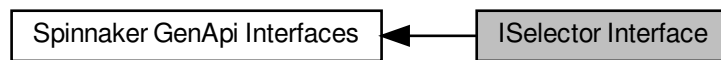
**8.102.3    Variable Documentation****8.102.3.1    IRegister**

```
interface SPINNAKER_API_ABSTRACT IRegister
```

[Interface](#) for registers.

## 8.103 ISelector Interface

Collaboration diagram for ISelector Interface:



### 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.*

#### 8.103.1 Detailed Description

#### 8.103.2 Function Documentation

##### 8.103.2.1 GetSelectedFeatures()

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

retrieve the group of selected features

##### 8.103.2.2 GetSelectingFeatures()

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

retrieve the group of features selecting this node

#### 8.103.3 Variable Documentation

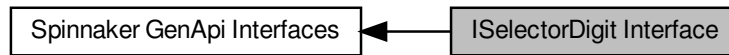
##### 8.103.3.1 ISelector

```
interface SPINNAKER_API_ABSTRACT ISelector
```

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

## 8.104 ISelectorDigit Interface

Collaboration diagram for ISelectorDigit Interface:



### 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.*

#### 8.104.1 Detailed Description

#### 8.104.2 Function Documentation

##### 8.104.2.1 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 seletor changed since the last GetNext are contained |
|--------------------|---------------------------------------------------------------------------------------------------------|

## 8.104.2.2 Restore()

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

Restores the selectors' values found at creation.

## 8.104.2.3 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

## 8.104.2.4 ToString()

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

Returns a string representation of the digit.

## 8.104.3 Variable Documentation

#### 8.104.3.1 ISelectorDigit

```
interface SPINNAKER_API_ABSTRACT ISelectorDigit
```

##### Initial value:

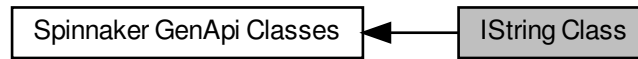
```
{

 virtual bool SetFirst() = 0
```

[Interface](#) of a "digit" of the "counter" formed by the selector set.

## 8.105 IString Class

Collaboration diagram for IString Class:



### Functions

- virtual int64\_t [GetMaxLength](#) ()=0  
*Retrieves the maximum length of the string in bytes.*

### Variables

- [interface SPINNAKER\\_API\\_ABSTRACT IString](#)  
*Interface for string properties.*

#### 8.105.1 Detailed Description

#### 8.105.2 Function Documentation

##### 8.105.2.1 GetMaxLength()

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

Retrieves the maximum length of the string in bytes.

#### 8.105.3 Variable Documentation

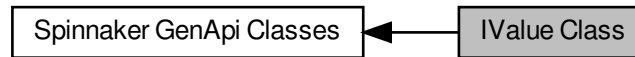
##### 8.105.3.1 IString

```
interface SPINNAKER_API_ABSTRACT IString
```

[Interface](#) for string properties.

## 8.106 IValue Class

Collaboration diagram for IValue Class:



### 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.*

### 8.106.1 Detailed Description

### 8.106.2 Function Documentation

#### 8.106.2.1 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) |



### 8.106.2.2 IsValueCacheValid()

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

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

### 8.106.2.3 ToString()

```
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

## 8.106.3 Variable Documentation

### 8.106.3.1 IValue

```
interface SPINNAKER_API_ABSTRACT IValue
```

[Interface](#) for value properties.

## 8.107 Node Class

Collaboration diagram for Node Class:



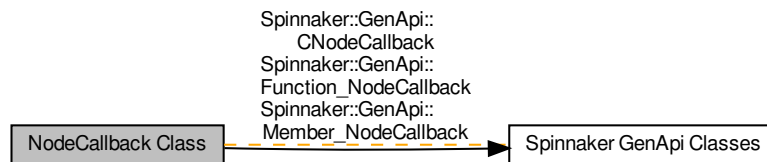
### Classes

- class [Node](#)  
*class common to all nodes*

### 8.107.1 Detailed Description

## 8.108 NodeCallback Class

Collaboration diagram for NodeCallback Class:



### 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.*

### 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.*

### 8.108.1 Detailed Description

### 8.108.2 Enumeration Type Documentation

#### 8.108.2.1 ECallbackType

enum [ECallbackType](#)

the type of callback

Enumerator

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

### 8.108.3 Function Documentation

#### 8.108.3.1 Deregister()

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

Unregistering callback by handle.

#### 8.108.3.2 make\_NodeCallback() [1/2]

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

make a new callback object for C functions

**8.108.3.3** `make_NodeCallback()` [2/2]

```
CNodeCallback* Spinnaker::GenApi::make_NodeCallback (
 INode * pNode,
 Client & client,
 Member member,
 ECallbackType CallbackType)
```

make a new callback object for member functions

**8.108.3.4** `Register()` [1/2]

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

Register a C-function as a callback.

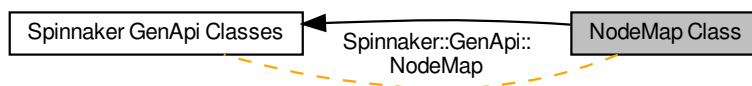
**8.108.3.5** `Register()` [2/2]

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

Register a C++-member function a callback.

## 8.109 NodeMap Class

Collaboration diagram for NodeMap Class:



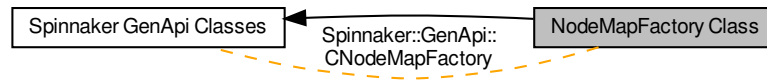
### Classes

- class [NodeMap](#)  
*Smart pointer template for NodeMaps with create function.*

### 8.109.1 Detailed Description

## 8.110 NodeMapFactory Class

Collaboration diagram for NodeMapFactory Class:



### Classes

- class [CNodeMapFactory](#)

*The node map factory is used for creating node maps from camera description files.*

### 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.*

#### 8.110.1 Detailed Description

#### 8.110.2 Enumeration Type Documentation

##### 8.110.2.1 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.                                                                          |

**8.110.2.2 EContentType\_t**

enum [EContentType\\_t](#)

Lists the processable file types.

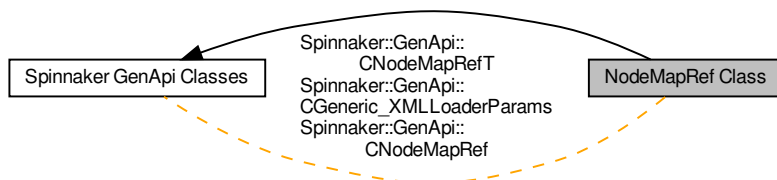
**Enumerator**

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



## 8.111 NodeMapRef Class

Collaboration diagram for NodeMapRef Class:



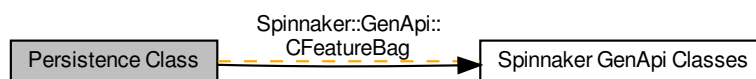
### Classes

- 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.*

#### 8.111.1 Detailed Description

## 8.112 Persistence Class

Collaboration diagram for Persistence Class:



### Classes

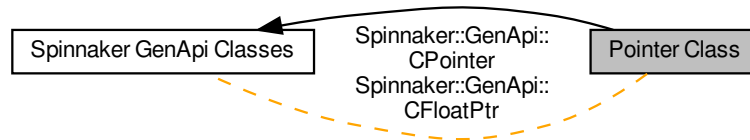
- class [CFeatureBag](#)

*Bag holding streamable features of a nodetree.*

### 8.112.1 Detailed Description

## 8.113 Pointer Class

Collaboration diagram for Pointer Class:



### Classes

- class `CPointer< T, B >`  
*Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.*
- class `CFloatPtr`  
*SmartPointer for IFloat interface pointer.*

### Typedefs

- typedef `CPointer< IBase > CBasePtr`  
*SmartPointer for IBase interface pointer.*
- typedef `CPointer< INode, IBase > CNodePtr`  
*SmartPointer for INode interface pointer.*
- typedef `CPointer< IValue > CValuePtr`  
*SmartPointer for IValue interface pointer.*
- typedef `CPointer< ICategory > CCategoryPtr`  
*SmartPointer for ICategory interface pointer.*
- typedef `CPointer< IBoolean > CBooleanPtr`  
*SmartPointer for IBoolean interface pointer.*
- typedef `CPointer< IInteger > CIntegerPtr`  
*SmartPointer for IInteger interface pointer.*
- typedef `CPointer< IString > CStringPtr`  
*SmartPointer for IString interface pointer.*
- typedef `CPointer< IRegister > CRegisterPtr`  
*SmartPointer for IRegister interface pointer.*
- typedef `CPointer< IEnumeration > CEnumerationPtr`  
*SmartPointer for IEnumeration interface pointer.*
- typedef `CPointer< IEnumEntry > CEnumEntryPtr`  
*SmartPointer for IEnumEntry interface pointer.*
- typedef `CPointer< IPort > CPortPtr`  
*SmartPointer for IPort interface pointer.*
- typedef `CPointer< IPortReplay > CPortReplayPtr`  
*SmartPointer for IPortReplay interface pointer.*
- typedef `CPointer< IPortRecorder > CPortRecorderPtr`

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

## Functions

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

### 8.113.1 Detailed Description

### 8.113.2 Typedef Documentation

#### 8.113.2.1 CBasePtr

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

SmartPointer for IBase interface pointer.

#### 8.113.2.2 CBooleanPtr

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

SmartPointer for IBoolean interface pointer.

#### 8.113.2.3 CCategoryPtr

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

SmartPointer for ICategory interface pointer.

#### 8.113.2.4 CChunkPortPtr

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

SmartPointer for IChunkPort interface pointer.

#### 8.113.2.5 CCommandPtr

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

SmartPointer for ICommand interface pointer.

#### 8.113.2.6 CDeviceInfoPtr

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

SmartPointer for IDeviceInfo interface pointer.

#### 8.113.2.7 CEnumEntryPtr

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

SmartPointer for IEnumEntry interface pointer.

#### 8.113.2.8 CEnumerationPtr

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

SmartPointer for IEnumeration interface pointer.

#### 8.113.2.9 CIntegerPtr

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

SmartPointer for IInteger interface pointer.

#### 8.113.2.10 CNodeMapDynPtr

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

SmartPointer for INodeMapDyn interface pointer.

#### 8.113.2.11 CNodeMapPtr

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

SmartPointer for INodeMap interface pointer.

#### 8.113.2.12 CNodePtr

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

SmartPointer for INode interface pointer.

#### 8.113.2.13 CPortConstructPtr

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

SmartPointer for IPortConstruct interface pointer.

**8.113.2.14 CPortPtr**

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

SmartPointer for IPort interface pointer.

**8.113.2.15 CPortRecorderPtr**

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

SmartPointer for IPortRecorder interface pointer.

**8.113.2.16 CPortReplayPtr**

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

SmartPointer for IPortReplay interface pointer.

**8.113.2.17 CPortWriteListPtr**

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

SmartPointer for IPortWriteList interface pointer.

**8.113.2.18 CRegisterPtr**

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

SmartPointer for IRegister interface pointer.

**8.113.2.19 CSelectorPtr**

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

SmartPointer for ISelector interface pointer.

#### 8.113.2.20 CStringPtr

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

SmartPointer for IString interface pointer.

#### 8.113.2.21 CValuePtr

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

SmartPointer for IValue interface pointer.

### 8.113.3 Function Documentation

#### 8.113.3.1 GetInterfaceName()

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

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

#### 8.113.3.2 IsAvailable()

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

Checks if a node is Available.

#### 8.113.3.3 IsImplemented()

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

Checks if a node is Implemented.

#### 8.113.3.4 IsReadable()

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

Checks if a node is readable.

#### 8.113.3.5 IsWritable()

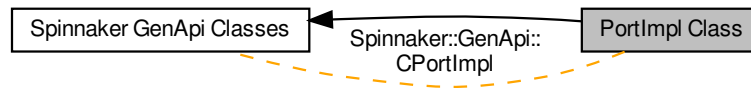
```
bool Spinnaker::GenApi::IsWritable (
 const Spinnaker::GenApi::CPointer< T, B > & ptr) [inline]
```

Checks if a node is Writable.



## 8.114 PortImpl Class

Collaboration diagram for PortImpl Class:



### Classes

- class [CPortImpl](#)  
*Standard implementation for a port.*

#### 8.114.1 Detailed Description

## 8.115 PortNode Class

Collaboration diagram for PortNode Class:



### Classes

- class [PortNode](#)  
*[Interface](#) for value properties.*

### Typedefs

- typedef [PortNode](#) [CPortRef](#)

#### 8.115.1 Detailed Description

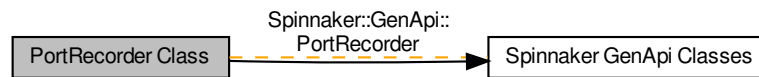
#### 8.115.2 Typedef Documentation

##### 8.115.2.1 CPortRef

```
typedef PortNode CPortRef
```

## 8.116 PortRecorder Class

Collaboration diagram for PortRecorder Class:



### Classes

- class [PortRecorder](#)  
*Interface for recording write commands on a port.*

### Typedefs

- typedef [PortRecorder](#) [CPortRecorderRef](#)  
*Reference to an IPortRecorder pointer.*

#### 8.116.1 Detailed Description

#### 8.116.2 Typedef Documentation

##### 8.116.2.1 CPortRecorderRef

typedef [PortRecorder](#) [CPortRecorderRef](#)

Reference to an IPortRecorder pointer.

## 8.117 PortReplay Class

Collaboration diagram for PortReplay Class:



### Classes

- class [PortReplay](#)  
*[Interface](#) for replaying write commands on a port.*

### 8.117.1 Detailed Description

## 8.118 PortWriteList Class

Collaboration diagram for PortWriteList Class:



### Classes

- class [CPortWriteList](#)  
*Container holding a list of port write commands.*

### 8.118.1 Detailed Description

## 8.119 Reference Interfaces

Collaboration diagram for Reference Interfaces:



### Functions

- virtual void [SetNumEnums](#) (int NumEnums)=0  
*sets the number of enum values*

#### 8.119.1 Detailed Description

#### 8.119.2 Function Documentation

##### 8.119.2.1 SetNumEnums()

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

sets the number of enum values

## 8.120 RegisterNode Class

Collaboration diagram for RegisterNode Class:



### Classes

- class [RegisterNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [RegisterNode](#) [CRegisterRef](#)

#### 8.120.1 Detailed Description

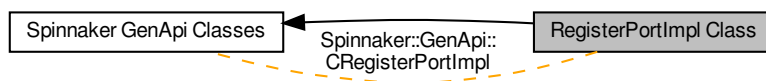
#### 8.120.2 Typedef Documentation

##### 8.120.2.1 CRegisterRef

```
typedef RegisterNode CRegisterRef
```

## 8.121 RegisterPortImpl Class

Collaboration diagram for RegisterPortImpl Class:



### Classes

- class [CRegisterPortImpl](#)

*Standard implementation for a port using a register based transport layer.*

### 8.121.1 Detailed Description



## 8.122 SelectorSet Class

Collaboration diagram for SelectorSet Class:



### Classes

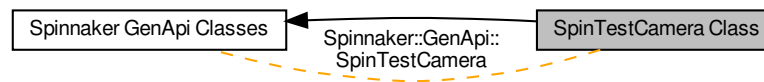
- class [CSelectorSet](#)

*The set of selectors selecting a given node.*

### 8.122.1 Detailed Description

## 8.123 SpinTestCamera Class

Collaboration diagram for SpinTestCamera Class:



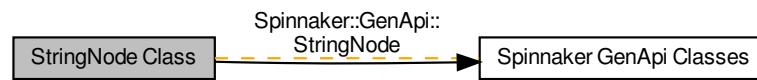
### Classes

- class [SpinTestCamera](#)

### 8.123.1 Detailed Description

## 8.124 StringNode Class

Collaboration diagram for StringNode Class:



### Classes

- class [StringNode](#)  
*Interface for string properties.*

### Typedefs

- typedef [StringNode](#) [CStringRef](#)

#### 8.124.1 Detailed Description

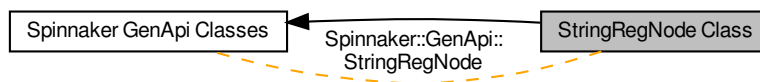
#### 8.124.2 Typedef Documentation

##### 8.124.2.1 CStringRef

```
typedef StringNode CStringRef
```

## 8.125 StringRegNode Class

Collaboration diagram for StringRegNode Class:



### Classes

- class [StringRegNode](#)  
*Interface for string properties.*

### 8.125.1 Detailed Description

## 8.126 StructPort Class

Collaboration diagram for StructPort Class:



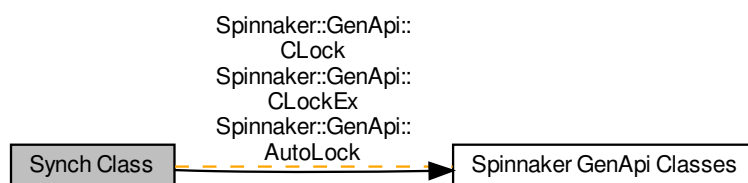
### Classes

- class `CTestPortStruct< CDataStruct >`  
*Implements a register spaces based on a C++ struct.*

### 8.126.1 Detailed Description

## 8.127 Synch Class

Collaboration diagram for Synch Class:



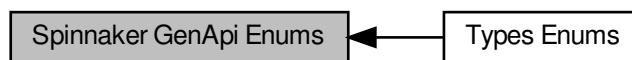
### Classes

- class [CLock](#)  
*A lock class.*
- class [CLockEx](#)  
*This class is for testing purposes only.*
- class [AutoLock](#)

### 8.127.1 Detailed Description

## 8.128 Spinnaker GenApi Enums

Collaboration diagram for Spinnaker GenApi Enums:



### Modules

- [Types Enums](#)

### 8.128.1 Detailed Description

## 8.129 Types Enums

Collaboration diagram for Types Enums:



### Macros

- `#define \_UndefinedRepresentation _UndefinedRepresentation`

### Typedefs

- `typedef GenICam::gcstring_vector StringList\_t`  
*A list of strings.*

### Enumerations

- `enum ESign {`  
`Signed,`  
`Unsigned,`  
`\_UndefinedSign }`  
*signed or unsigned integers*
- `enum EAccessMode {`  
`NI,`  
`NA,`  
`WO,`  
`RO,`  
`RW,`  
`\_UndefinedAccesMode,`  
`\_CycleDetectAccesMode }`  
*access mode of a node*
- `enum EVisibility {`  
`Beginner = 0,`  
`Expert = 1,`  
`Guru = 2,`  
`Invisible = 3,`  
`\_UndefinedVisibility = 99 }`  
*recommended visibility of a node*
- `enum ECachingMode {`  
`NoCache,`  
`WriteThrough,`  
`WriteAround,`  
`\_UndefinedCachingMode }`  
*caching mode of a register*



- enum `ERepresentation` {  
`Linear`,  
`Logarithmic`,  
`Boolean`,  
`PureNumber`,  
`HexNumber`,  
`IPV4Address`,  
`MACAddress`,  
`_UndefinedRepresentation` }  
*recommended representation of a node value*
- enum `EEndianess` {  
`BigEndian`,  
`LittleEndian`,  
`_UndefinedEndian` }  
*Endianess of a value in a register.*
- enum `ENameSpace` {  
`Custom`,  
`Standard`,  
`_UndefinedNameSpace` }  
*Defines if a node name is standard or custom.*
- enum `EStandardNameSpace` {  
`None`,  
`GEV`,  
`IIDC`,  
`CL`,  
`USB`,  
`_UndefinedStandardNameSpace` }  
*Defines from which standard namespace a node name comes from.*
- enum `EYesNo` {  
`Yes` = 1,  
`No` = 0,  
`_UndefinedYesNo` = 2 }  
*Defines the choices of a Yes/No alternative.*
- enum `ESlope` {  
`Increasing`,  
`Decreasing`,  
`Varying`,  
`Automatic`,  
`_UndefinedESlope` }  
*typedef for formula type*
- enum `EXMLValidation` {  
`xvLoad` = 0x00000001L,  
`xvCycles` = 0x00000002L,  
`xvSFNC` = 0x00000004L,  
`xvDefault` = 0x00000000L,  
`xvAll` = 0xffffffffL,  
`_UndefinedEXMLValidation` = 0x80000000L }  
*typedef describing the different validity checks which can be performed on an XML file*
- enum `EDisplayNotation` {  
`fnAutomatic`,  
`fnFixed`,  
`fnScientific`,  
`_UndefinedEDisplayNotation` }  
*typedef for float notation*
- enum `EInterfaceType` {  
`intfIValue`,

```

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.*

### 8.129.1 Detailed Description

### 8.129.2 Macro Definition Documentation

#### 8.129.2.1 `_UndefinedRepresentation`

```
#define _UndefinedRepresentation _UndefinedRepresentation
```

### 8.129.3 Typedef Documentation

### 8.129.3.1 StringList\_t

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

A list of strings.

## 8.129.4 Enumeration Type Documentation

### 8.129.4.1 EAccessMode

```
enum EAccessMode
```

access mode of a node

#### Enumerator

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

### 8.129.4.2 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 |

### 8.129.4.3 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  |

## 8.129.4.4 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 |

## 8.129.4.5 EGenApiSchemaVersion

enum [EGenApiSchemaVersion](#)

[GenApi](#) schema version.

## Enumerator

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

## 8.129.4.6 EIncMode

enum [EIncMode](#)

typedef for increment mode

## Enumerator

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

#### 8.129.4.7 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 |

#### 8.129.4.8 EInterfaceType

enum [EInterfaceType](#)

typedef for interface type

##### Enumerator

|                  |                                       |
|------------------|---------------------------------------|
| intflValue       |                                       |
| intflBase        | IValue interface.                     |
| intflInteger     | IBase interface.                      |
| intflBoolean     | IInteger interface.                   |
| intflCommand     | IBoolean interface.                   |
| intflFloat       | ICommand interface.                   |
| intflString      | IFloat interface.                     |
| intflRegister    | IString interface.                    |
| intflCategory    | IRegister interface.                  |
| intflEnumeration | ICategory interface.                  |
| intflEnumEntry   | IEnumeration interface.               |
| intflPort        | IEnumEntry interface. IPort interface |

#### 8.129.4.9 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 |

## 8.129.4.10 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 |

## 8.129.4.11 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.                       |

## 8.129.4.12 ESign

enum [ESign](#)

signed or unsigned integers

#### Enumerator

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

#### 8.129.4.13 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 |

#### 8.129.4.14 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 |

#### 8.129.4.15 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.                  |

## 8.129.4.16 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                                                       |

## 8.129.4.17 EYesNo

enum [EYesNo](#)

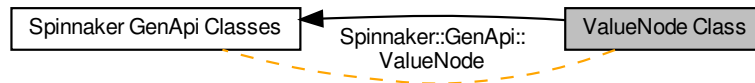
Defines the choices of a Yes/No alternative.

## Enumerator

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

## 8.130 ValueNode Class

Collaboration diagram for ValueNode Class:



### Classes

- class [ValueNode](#)  
*Interface for value properties.*

### Typedefs

- typedef [ValueNode](#) [CValueRef](#)

#### 8.130.1 Detailed Description

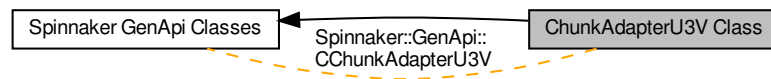
#### 8.130.2 Typedef Documentation

##### 8.130.2.1 CValueRef

```
typedef ValueNode CValueRef
```

## 8.131 ChunkAdapterU3V Class

Collaboration diagram for ChunkAdapterU3V Class:



### Classes

- class [CChunkAdapterU3V](#)  
*Connects a chunked U3V buffer to a node map.*

#### 8.131.1 Detailed Description



## Chapter 9

# Namespace Documentation

### 9.1 Spinnaker Namespace Reference

#### Namespaces

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

#### Classes

- struct [ActionCommandResult](#)  
*Action Command Result.*
- class [ArrivalEvent](#)  
*An event handler for capturing the device arrival event.*
- 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 [DeviceEvent](#)  
*A handler to device events.*
- class [Event](#)  
*The base class for all event types.*
- class [Exception](#)  
*The [Exception](#) object represents an error that is returned from the library.*

- class [IArrivalEvent](#)
- 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 [IDeviceEvent](#)
- class [IImage](#)
  - The interface file for [Image](#).*
- class [IImageEvent](#)
- class [IImageStatistics](#)
  - The interface file for image statistics.*
- class [IInterface](#)
  - The interface file for [Interface](#).*
- class [IInterfaceEvent](#)
- class [IInterfaceList](#)
  - The interface file for [InterfaceList](#) class.*
- class [ILoggingEvent](#)
- class [Image](#)
  - The image object class.*
- class [ImageEvent](#)
  - A handler for capturing image arrival events.*
- class [ImagePtr](#)
  - A reference tracked pointer to an image object.*
- class [ImageStatistics](#)
  - Represents image statistics for an image.*
- class [ImageUtility](#)
  - Static helper functions for the image object class.*
- class [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.*
- class [Interface](#)
  - An interface object which holds a list of cameras.*
- class [InterfaceEvent](#)
  - 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 [IRemovalEvent](#)
- class [ISystem](#)
  - The interface file for [System](#).*
- 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 [LoggingEvent](#)  
*An event handler for capturing the device logging event.*
- class [LoggingEventData](#)  
*The [LoggingEventData](#) object.*
- class [LoggingEventDataPtr](#)  
*A reference tracked pointer to the [LoggingEvent](#) object.*
- struct [PGMOption](#)  
*Options for saving PGM images.*
- struct [PNGOption](#)  
*Options for saving PNG images.*
- struct [PPMOption](#)  
*Options for saving PPM images.*
- class [RemovalEvent](#)  
*An event handler for capturing the device removal event.*
- class [System](#)  
*The system object is used to retrieve the list of interfaces and cameras available.*
- 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_DEVICEREGISTERSENDIANNES }

• 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\_DEVICESTREAMCHANNELENDIANNESS }
- enum DeviceClockSelectorEnums {
  - DeviceClockSelector\_Sensor,
  - DeviceClockSelector\_SensorDigitization,
  - DeviceClockSelector\_CameraLink,
  - NUM\_DEVICECLOCKSELECTOR }
- enum DeviceSerialPortSelectorEnums {
  - DeviceSerialPortSelector\_CameraLink,
  - NUM\_DEVICESERIALPORTSELECTOR }
- enum DeviceSerialPortBaudRateEnums {
  - DeviceSerialPortBaudRate\_Baud9600,
  - DeviceSerialPortBaudRate\_Baud19200,
  - DeviceSerialPortBaudRate\_Baud38400,
  - DeviceSerialPortBaudRate\_Baud57600,
  - DeviceSerialPortBaudRate\_Baud115200,
  - DeviceSerialPortBaudRate\_Baud230400,
  - DeviceSerialPortBaudRate\_Baud460800,
  - DeviceSerialPortBaudRate\_Baud921600,
  - NUM\_DEVICESERIALPORTBAUDRATE }
- enum SensorTapsEnums {
  - SensorTaps\_One,
  - SensorTaps\_Two,
  - SensorTaps\_Three,
  - SensorTaps\_Four,
  - SensorTaps\_Eight,
  - SensorTaps\_Ten,
  - NUM\_SENSORTAPS }
- enum SensorDigitizationTapsEnums {
  - SensorDigitizationTaps\_One,
  - SensorDigitizationTaps\_Two,

```
SensorDigitizationTaps_Three,
SensorDigitizationTaps_Four,
SensorDigitizationTaps_Eight,
SensorDigitizationTaps_Ten,
NUM_SENSORDIGITIZATIONTAPS }

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

• enum RegionModeEnums {
 RegionMode_Off,
 RegionMode_On,
 NUM_REGIONMODE }

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

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

• enum PixelFormatInfoSelectorEnums {
 PixelFormatInfoSelector_Mono1p,
 PixelFormatInfoSelector_Mono2p,
 PixelFormatInfoSelector_Mono4p,
 PixelFormatInfoSelector_Mono8,
 PixelFormatInfoSelector_Mono8s,
 PixelFormatInfoSelector_Mono10,
 PixelFormatInfoSelector_Mono10p,
 PixelFormatInfoSelector_Mono12,
 PixelFormatInfoSelector_Mono12p,
 PixelFormatInfoSelector_Mono14,
 PixelFormatInfoSelector_Mono16,
 PixelFormatInfoSelector_Mono16s,
 PixelFormatInfoSelector_Mono32f,
 PixelFormatInfoSelector_BayerBG8,
 PixelFormatInfoSelector_BayerBG10,
 PixelFormatInfoSelector_BayerBG10p,
 PixelFormatInfoSelector_BayerBG12,
 PixelFormatInfoSelector_BayerBG12p,
 PixelFormatInfoSelector_BayerBG16,
 PixelFormatInfoSelector_BayerGB8,
 PixelFormatInfoSelector_BayerGB10,
 PixelFormatInfoSelector_BayerGB10p,
 PixelFormatInfoSelector_BayerGB12,
 PixelFormatInfoSelector_BayerGB12p,
 PixelFormatInfoSelector_BayerGB16,
 PixelFormatInfoSelector_BayerGR8,
 PixelFormatInfoSelector_BayerGR10,
```



[PixelFormatInfoSelector\\_BayerGR10p](#),  
[PixelFormatInfoSelector\\_BayerGR12](#),  
[PixelFormatInfoSelector\\_BayerGR12p](#),  
[PixelFormatInfoSelector\\_BayerGR16](#),  
[PixelFormatInfoSelector\\_BayerRG8](#),  
[PixelFormatInfoSelector\\_BayerRG10](#),  
[PixelFormatInfoSelector\\_BayerRG10p](#),  
[PixelFormatInfoSelector\\_BayerRG12](#),  
[PixelFormatInfoSelector\\_BayerRG12p](#),  
[PixelFormatInfoSelector\\_BayerRG16](#),  
[PixelFormatInfoSelector\\_RGBa8](#),  
[PixelFormatInfoSelector\\_RGBa10](#),  
[PixelFormatInfoSelector\\_RGBa10p](#),  
[PixelFormatInfoSelector\\_RGBa12](#),  
[PixelFormatInfoSelector\\_RGBa12p](#),  
[PixelFormatInfoSelector\\_RGBa14](#),  
[PixelFormatInfoSelector\\_RGBa16](#),  
[PixelFormatInfoSelector\\_RGB8](#),  
[PixelFormatInfoSelector\\_RGB8\\_Planar](#),  
[PixelFormatInfoSelector\\_RGB10](#),  
[PixelFormatInfoSelector\\_RGB10\\_Planar](#),  
[PixelFormatInfoSelector\\_RGB10p](#),  
[PixelFormatInfoSelector\\_RGB10p32](#),  
[PixelFormatInfoSelector\\_RGB12](#),  
[PixelFormatInfoSelector\\_RGB12\\_Planar](#),  
[PixelFormatInfoSelector\\_RGB12p](#),  
[PixelFormatInfoSelector\\_RGB14](#),  
[PixelFormatInfoSelector\\_RGB16](#),  
[PixelFormatInfoSelector\\_RGB16s](#),  
[PixelFormatInfoSelector\\_RGB32f](#),  
[PixelFormatInfoSelector\\_RGB16\\_Planar](#),  
[PixelFormatInfoSelector\\_RGB565p](#),  
[PixelFormatInfoSelector\\_BGRa8](#),  
[PixelFormatInfoSelector\\_BGRa10](#),  
[PixelFormatInfoSelector\\_BGRa10p](#),  
[PixelFormatInfoSelector\\_BGRa12](#),  
[PixelFormatInfoSelector\\_BGRa12p](#),  
[PixelFormatInfoSelector\\_BGRa14](#),  
[PixelFormatInfoSelector\\_BGRa16](#),  
[PixelFormatInfoSelector\\_RGBa32f](#),  
[PixelFormatInfoSelector\\_BGR8](#),  
[PixelFormatInfoSelector\\_BGR10](#),  
[PixelFormatInfoSelector\\_BGR10p](#),  
[PixelFormatInfoSelector\\_BGR12](#),  
[PixelFormatInfoSelector\\_BGR12p](#),  
[PixelFormatInfoSelector\\_BGR14](#),  
[PixelFormatInfoSelector\\_BGR16](#),  
[PixelFormatInfoSelector\\_BGR565p](#),  
[PixelFormatInfoSelector\\_R8](#),  
[PixelFormatInfoSelector\\_R10](#),  
[PixelFormatInfoSelector\\_R12](#),  
[PixelFormatInfoSelector\\_R16](#),  
[PixelFormatInfoSelector\\_G8](#),  
[PixelFormatInfoSelector\\_G10](#),  
[PixelFormatInfoSelector\\_G12](#),  
[PixelFormatInfoSelector\\_G16](#),  
[PixelFormatInfoSelector\\_B8](#),  
[PixelFormatInfoSelector\\_B10](#),

[PixelFormatInfoSelector\\_B12,](#)  
[PixelFormatInfoSelector\\_B16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC8,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC8\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC10p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC10p\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC12p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC12p\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC16\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC32f,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_ABC32f\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC8,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC8\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC10p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC10p\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC12p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC12p\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC16\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC32f,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_AC32f\\_Planar,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_A8,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_A10p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_A12p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_A16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_A32f,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_B8,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_B10p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_B12p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_B16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_B32f,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_C8,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_C10p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_C12p,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_C16,](#)  
[PixelFormatInfoSelector\\_Coord3D\\_C32f,](#)  
[PixelFormatInfoSelector\\_Confidence1,](#)  
[PixelFormatInfoSelector\\_Confidence1p,](#)  
[PixelFormatInfoSelector\\_Confidence8,](#)  
[PixelFormatInfoSelector\\_Confidence16,](#)  
[PixelFormatInfoSelector\\_Confidence32f,](#)  
[PixelFormatInfoSelector\\_BiColorBGRG8,](#)  
[PixelFormatInfoSelector\\_BiColorBGRG10,](#)  
[PixelFormatInfoSelector\\_BiColorBGRG10p,](#)  
[PixelFormatInfoSelector\\_BiColorBGRG12,](#)  
[PixelFormatInfoSelector\\_BiColorBGRG12p,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG8,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG10,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG10p,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG12,](#)  
[PixelFormatInfoSelector\\_BiColorRGBG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG8,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG10,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG10p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG12,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WBWG14,](#)

[PixelFormatInfoSelector\\_SCF1WBWG16,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB8,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB10,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB10p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB12,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB12p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB14,](#)  
[PixelFormatInfoSelector\\_SCF1WGWB16,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR8,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR10,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR10p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR12,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR12p,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR14,](#)  
[PixelFormatInfoSelector\\_SCF1WGWR16,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG8,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG10,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG10p,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG12,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG12p,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG14,](#)  
[PixelFormatInfoSelector\\_SCF1WRWG16,](#)  
[PixelFormatInfoSelector\\_YCbCr8,](#)  
[PixelFormatInfoSelector\\_YCbCr8\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr10\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr10p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr12\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr12p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr411\\_8,](#)  
[PixelFormatInfoSelector\\_YCbCr411\\_8\\_CbYYCrYY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_8,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_8\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10p,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_10p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12p,](#)  
[PixelFormatInfoSelector\\_YCbCr422\\_12p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_8\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_10\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_10p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_12\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_12p\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_411\\_8\\_CbYYCrYY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_8,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_8\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_10,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_10\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_10p,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_10p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_12,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_12\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_12p,](#)  
[PixelFormatInfoSelector\\_YCbCr601\\_422\\_12p\\_CbYCrY,](#)  
[PixelFormatInfoSelector\\_YCbCr709\\_8\\_CbYCr,](#)  
[PixelFormatInfoSelector\\_YCbCr709\\_10\\_CbYCr,](#)

```

PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

• enum DeinterlacingEnums {
 Deinterlacing_Off,
 Deinterlacing_LineDuplication,
 Deinterlacing_Weave,
 NUM_DEINTERLACING }

• enum ImageCompressionRateOptionEnums {
 ImageCompressionRateOption_FixBitrate,
 ImageCompressionRateOption_FixQuality,
 NUM_IMAGECOMPRESSIONRATEOPTION }

• enum ImageCompressionJPEGFormatOptionEnums {
 ImageCompressionJPEGFormatOption_Lossless,
 ImageCompressionJPEGFormatOption_BaselineStandard,
 ImageCompressionJPEGFormatOption_BaselineOptimized,
 ImageCompressionJPEGFormatOption_Progressive,
 NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }

• enum AcquisitionStatusSelectorEnums {
 AcquisitionStatusSelector_AcquisitionTriggerWait,
 AcquisitionStatusSelector_AcquisitionActive,
 AcquisitionStatusSelector_AcquisitionTransfer,
 AcquisitionStatusSelector_FrameTriggerWait,
 AcquisitionStatusSelector_FrameActive,
 AcquisitionStatusSelector_ExposureActive,
 NUM_ACQUISITIONSTATUSSELECTOR }

• enum ExposureTimeModeEnums {
 ExposureTimeMode_Common,
 ExposureTimeMode_Individual,
 NUM_EXPOSURETIMEMODE }

```

- enum ExposureTimeSelectorEnums {  
ExposureTimeSelector\_Common,  
ExposureTimeSelector\_Red,  
ExposureTimeSelector\_Green,  
ExposureTimeSelector\_Blue,  
ExposureTimeSelector\_Cyan,  
ExposureTimeSelector\_Magenta,  
ExposureTimeSelector\_Yellow,  
ExposureTimeSelector\_Infrared,  
ExposureTimeSelector\_Ultraviolet,  
ExposureTimeSelector\_Stage1,  
ExposureTimeSelector\_Stage2,  
NUM\_EXPOSURETIMESELECTOR }
- enum GainAutoBalanceEnums {  
GainAutoBalance\_Off,  
GainAutoBalance\_Once,  
GainAutoBalance\_Continuous,  
NUM\_GAINAUTOBALANCE }
- enum BlackLevelAutoEnums {  
BlackLevelAuto\_Off,  
BlackLevelAuto\_Once,  
BlackLevelAuto\_Continuous,  
NUM\_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {  
BlackLevelAutoBalance\_Off,  
BlackLevelAutoBalance\_Once,  
BlackLevelAutoBalance\_Continuous,  
NUM\_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {  
WhiteClipSelector\_All,  
WhiteClipSelector\_Red,  
WhiteClipSelector\_Green,  
WhiteClipSelector\_Blue,  
WhiteClipSelector\_Y,  
WhiteClipSelector\_U,  
WhiteClipSelector\_V,  
WhiteClipSelector\_Tap1,  
WhiteClipSelector\_Tap2,  
NUM\_WHITECLIPSELECTOR }
- enum TimerSelectorEnums {  
TimerSelector\_Timer0,  
TimerSelector\_Timer1,  
TimerSelector\_Timer2,  
NUM\_TIMERSELECTOR }
- enum TimerStatusEnums {  
TimerStatus\_TimerIdle,  
TimerStatus\_TimerTriggerWait,  
TimerStatus\_TimerActive,  
TimerStatus\_TimerCompleted,  
NUM\_TIMERSTATUS }
- enum TimerTriggerSourceEnums {  
TimerTriggerSource\_Off,  
TimerTriggerSource\_AcquisitionTrigger,  
TimerTriggerSource\_AcquisitionStart,  
TimerTriggerSource\_AcquisitionEnd,  
TimerTriggerSource\_FrameTrigger,  
TimerTriggerSource\_FrameStart,  
TimerTriggerSource\_FrameEnd,

```

TimerTriggerSource_FrameBurstStart,
TimerTriggerSource_FrameBurstEnd,
TimerTriggerSource_LineTrigger,
TimerTriggerSource_LineStart,
TimerTriggerSource_LineEnd,
TimerTriggerSource_ExposureStart,
TimerTriggerSource_ExposureEnd,
TimerTriggerSource_Line0,
TimerTriggerSource_Line1,
TimerTriggerSource_Line2,
TimerTriggerSource_UserOutput0,
TimerTriggerSource_UserOutput1,
TimerTriggerSource_UserOutput2,
TimerTriggerSource_Counter0Start,
TimerTriggerSource_Counter1Start,
TimerTriggerSource_Counter2Start,
TimerTriggerSource_Counter0End,
TimerTriggerSource_Counter1End,
TimerTriggerSource_Counter2End,
TimerTriggerSource_Timer0Start,
TimerTriggerSource_Timer1Start,
TimerTriggerSource_Timer2Start,
TimerTriggerSource_Timer0End,
TimerTriggerSource_Timer1End,
TimerTriggerSource_Timer2End,
TimerTriggerSource_Encoder0,
TimerTriggerSource_Encoder1,
TimerTriggerSource_Encoder2,
TimerTriggerSource_SoftwareSignal0,
TimerTriggerSource_SoftwareSignal1,
TimerTriggerSource_SoftwareSignal2,
TimerTriggerSource_Action0,
TimerTriggerSource_Action1,
TimerTriggerSource_Action2,
TimerTriggerSource_LinkTrigger0,
TimerTriggerSource_LinkTrigger1,
TimerTriggerSource_LinkTrigger2,
NUM_TIMERTRIGGERSOURCE }
• enum TimerTriggerActivationEnums {
 TimerTriggerActivation_RisingEdge,
 TimerTriggerActivation_FallingEdge,
 TimerTriggerActivation_AnyEdge,
 TimerTriggerActivation_LevelHigh,
 TimerTriggerActivation_LevelLow,
 NUM_TIMERTRIGGERACTIVATION }
• enum EncoderSelectorEnums {
 EncoderSelector_Encoder0,
 EncoderSelector_Encoder1,
 EncoderSelector_Encoder2,
 NUM_ENCODERSELECTOR }
• enum EncoderSourceAEnums {
 EncoderSourceA_Off,
 EncoderSourceA_Line0,
 EncoderSourceA_Line1,
 EncoderSourceA_Line2,
 NUM_ENCODERSOURCEA }
• enum EncoderSourceBEnums {
 EncoderSourceB_Off,

```

```
EncoderSourceB_Line0,
EncoderSourceB_Line1,
EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }

• enum EncoderModeEnums {
EncoderMode_FourPhase,
EncoderMode_HighResolution,
NUM_ENCODERMODE }

• enum EncoderOutputModeEnums {
EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,
EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }

• enum EncoderStatusEnums {
EncoderStatus_EncoderUp,
EncoderStatus_EncoderDown,
EncoderStatus_EncoderIdle,
EncoderStatus_EncoderStatic,
NUM_ENCODERSTATUS }

• enum EncoderResetSourceEnums {
EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,
EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
```

```

EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }
• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }
• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }
• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }
• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,
NUM_TRANSFERTRIGGERSELECTOR }
• enum TransferTriggerModeEnums {
TransferTriggerMode_Off,
TransferTriggerMode_On,
NUM_TRANSFERTRIGGERMODE }
• enum TransferTriggerSourceEnums {
TransferTriggerSource_Line0,
TransferTriggerSource_Line1,
TransferTriggerSource_Line2,
TransferTriggerSource_Counter0Start,
TransferTriggerSource_Counter1Start,
TransferTriggerSource_Counter2Start,
TransferTriggerSource_Counter0End,
TransferTriggerSource_Counter1End,
TransferTriggerSource_Counter2End,
TransferTriggerSource_Timer0Start,
TransferTriggerSource_Timer1Start,
TransferTriggerSource_Timer2Start,
TransferTriggerSource_Timer0End,

```



```

TransferTriggerSource_Timer1End,
TransferTriggerSource_Timer2End,
TransferTriggerSource_SoftwareSignal0,
TransferTriggerSource_SoftwareSignal1,
TransferTriggerSource_SoftwareSignal2,
TransferTriggerSource_Action0,
TransferTriggerSource_Action1,
TransferTriggerSource_Action2,
NUM_TRANSFERTRIGGERSOURCE }
• enum TransferTriggerActivationEnums {
TransferTriggerActivation_RisingEdge,
TransferTriggerActivation_FallingEdge,
TransferTriggerActivation_AnyEdge,
TransferTriggerActivation_LevelHigh,
TransferTriggerActivation_LevelLow,
NUM_TRANSFERTRIGGERACTIVATION }
• enum TransferStatusSelectorEnums {
TransferStatusSelector_Streaming,
TransferStatusSelector_Paused,
TransferStatusSelector_Stopping,
TransferStatusSelector_Stopped,
TransferStatusSelector_QueueOverflow,
NUM_TRANSFERSTATUSSELECTOR }
• enum TransferComponentSelectorEnums {
TransferComponentSelector_Red,
TransferComponentSelector_Green,
TransferComponentSelector_Blue,
TransferComponentSelector_All,
NUM_TRANSFERCOMPONENTSELECTOR }
• enum Scan3dDistanceUnitEnums {
Scan3dDistanceUnit_Millimeter,
Scan3dDistanceUnit_Inch,
NUM_SCAN3DDISTANCEUNIT }
• enum Scan3dCoordinateSystemEnums {
Scan3dCoordinateSystem_Cartesian,
Scan3dCoordinateSystem_Spherical,
Scan3dCoordinateSystem_Cylindrical,
NUM_SCAN3DCOORDINATESYSTEM }
• enum Scan3dOutputModeEnums {
Scan3dOutputMode_UncalibratedC,
Scan3dOutputMode_CalibratedABC_Grid,
Scan3dOutputMode_CalibratedABC_PointCloud,
Scan3dOutputMode_CalibratedAC,
Scan3dOutputMode_CalibratedAC_Linescan,
Scan3dOutputMode_CalibratedC,
Scan3dOutputMode_CalibratedC_Linescan,
Scan3dOutputMode_RectifiedC,
Scan3dOutputMode_RectifiedC_Linescan,
Scan3dOutputMode_DisparityC,
Scan3dOutputMode_DisparityC_Linescan,
NUM_SCAN3DOUTPUTMODE }
• enum Scan3dCoordinateSystemReferenceEnums {
Scan3dCoordinateSystemReference_Anchor,
Scan3dCoordinateSystemReference_Transformed,
NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
• enum Scan3dCoordinateSelectorEnums {
Scan3dCoordinateSelector_CoordinateA,
Scan3dCoordinateSelector_CoordinateB,

```

```

Scan3dCoordinateSelector_CoordinateC,
NUM_SCAN3DCOORDINATESELECTOR }
• enum Scan3dCoordinateTransformSelectorEnums {
Scan3dCoordinateTransformSelector_RotationX,
Scan3dCoordinateTransformSelector_RotationY,
Scan3dCoordinateTransformSelector_RotationZ,
Scan3dCoordinateTransformSelector_TranslationX,
Scan3dCoordinateTransformSelector_TranslationY,
Scan3dCoordinateTransformSelector_TranslationZ,
NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }
• enum Scan3dCoordinateReferenceSelectorEnums {
Scan3dCoordinateReferenceSelector_RotationX,
Scan3dCoordinateReferenceSelector_RotationY,
Scan3dCoordinateReferenceSelector_RotationZ,
Scan3dCoordinateReferenceSelector_TranslationX,
Scan3dCoordinateReferenceSelector_TranslationY,
Scan3dCoordinateReferenceSelector_TranslationZ,
NUM_SCAN3DCOORDINATEREFERENCESELECTOR }
• enum ChunkImageComponentEnums {
ChunkImageComponent_Intensity,
ChunkImageComponent_Color,
ChunkImageComponent_Infrared,
ChunkImageComponent_Ultraviolet,
ChunkImageComponent_Range,
ChunkImageComponent_Disparity,
ChunkImageComponent_Confidence,
ChunkImageComponent_Scatter,
NUM_CHUNKIMAGECOMPONENT }
• enum ChunkCounterSelectorEnums {
ChunkCounterSelector_Counter0,
ChunkCounterSelector_Counter1,
ChunkCounterSelector_Counter2,
NUM_CHUNKCOUNTERSELECTOR }
• enum ChunkTimerSelectorEnums {
ChunkTimerSelector_Timer0,
ChunkTimerSelector_Timer1,
ChunkTimerSelector_Timer2,
NUM_CHUNKTIMERSELECTOR }
• enum ChunkEncoderSelectorEnums {
ChunkEncoderSelector_Encoder0,
ChunkEncoderSelector_Encoder1,
ChunkEncoderSelector_Encoder2,
NUM_CHUNKENCODERSELECTOR }
• enum ChunkEncoderStatusEnums {
ChunkEncoderStatus_EncoderUp,
ChunkEncoderStatus_EncoderDown,
ChunkEncoderStatus_EncoderIdle,
ChunkEncoderStatus_EncoderStatic,
NUM_CHUNKENCODERSTATUS }
• enum ChunkExposureTimeSelectorEnums {
ChunkExposureTimeSelector_Common,
ChunkExposureTimeSelector_Red,
ChunkExposureTimeSelector_Green,
ChunkExposureTimeSelector_Blue,
ChunkExposureTimeSelector_Cyan,
ChunkExposureTimeSelector_Magenta,
ChunkExposureTimeSelector_Yellow,
ChunkExposureTimeSelector_Infrared,

```

- ChunkExposureTimeSelector\_Ultraviolet,
- ChunkExposureTimeSelector\_Stage1,
- ChunkExposureTimeSelector\_Stage2,
- NUM\_CHUNKEXPOSURETIMESELECTOR }
- enum ChunkSourceIDEnums {
  - ChunkSourceID\_Source0,
  - ChunkSourceID\_Source1,
  - ChunkSourceID\_Source2,
  - NUM\_CHUNKSOURCEID }
- enum ChunkRegionIDEnums {
  - ChunkRegionID\_Region0,
  - ChunkRegionID\_Region1,
  - ChunkRegionID\_Region2,
  - NUM\_CHUNKREGIONID }
- enum ChunkTransferStreamIDEnums {
  - ChunkTransferStreamID\_Stream0,
  - ChunkTransferStreamID\_Stream1,
  - ChunkTransferStreamID\_Stream2,
  - ChunkTransferStreamID\_Stream3,
  - NUM\_CHUNKTRANSFERSTREAMID }
- enum ChunkScan3dDistanceUnitEnums {
  - ChunkScan3dDistanceUnit\_Millimeter,
  - ChunkScan3dDistanceUnit\_Inch,
  - NUM\_CHUNKSCAN3DDISTANCEUNIT }
- enum ChunkScan3dOutputModeEnums {
  - ChunkScan3dOutputMode\_UncalibratedC,
  - ChunkScan3dOutputMode\_CalibratedABC\_Grid,
  - ChunkScan3dOutputMode\_CalibratedABC\_PointCloud,
  - ChunkScan3dOutputMode\_CalibratedAC,
  - ChunkScan3dOutputMode\_CalibratedAC\_Linescan,
  - ChunkScan3dOutputMode\_CalibratedC,
  - ChunkScan3dOutputMode\_CalibratedC\_Linescan,
  - ChunkScan3dOutputMode\_RectifiedC,
  - ChunkScan3dOutputMode\_RectifiedC\_Linescan,
  - ChunkScan3dOutputMode\_DisparityC,
  - ChunkScan3dOutputMode\_DisparityC\_Linescan,
  - NUM\_CHUNKSCAN3DOUTPUTMODE }
- enum ChunkScan3dCoordinateSystemEnums {
  - ChunkScan3dCoordinateSystem\_Cartesian,
  - ChunkScan3dCoordinateSystem\_Spherical,
  - ChunkScan3dCoordinateSystem\_Cylindrical,
  - NUM\_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {
  - ChunkScan3dCoordinateSystemReference\_Anchor,
  - ChunkScan3dCoordinateSystemReference\_Transformed,
  - NUM\_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {
  - ChunkScan3dCoordinateSelector\_CoordinateA,
  - ChunkScan3dCoordinateSelector\_CoordinateB,
  - ChunkScan3dCoordinateSelector\_CoordinateC,
  - NUM\_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {
  - ChunkScan3dCoordinateTransformSelector\_RotationX,
  - ChunkScan3dCoordinateTransformSelector\_RotationY,
  - ChunkScan3dCoordinateTransformSelector\_RotationZ,
  - ChunkScan3dCoordinateTransformSelector\_TranslationX,
  - ChunkScan3dCoordinateTransformSelector\_TranslationY,

- ```

ChunkScan3dCoordinateTransformSelector_TranslationZ,
NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }

• enum ChunkScan3dCoordinateReferenceSelectorEnums {
    ChunkScan3dCoordinateReferenceSelector_RotationX,
    ChunkScan3dCoordinateReferenceSelector_RotationY,
    ChunkScan3dCoordinateReferenceSelector_RotationZ,
    ChunkScan3dCoordinateReferenceSelector_TranslationX,
    ChunkScan3dCoordinateReferenceSelector_TranslationY,
    ChunkScan3dCoordinateReferenceSelector_TranslationZ,
    NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }

• enum DeviceTapGeometryEnums {
    DeviceTapGeometry_Geometry_1X_1Y,
    DeviceTapGeometry_Geometry_1X2_1Y,
    DeviceTapGeometry_Geometry_1X2_1Y2,
    DeviceTapGeometry_Geometry_2X_1Y,
    DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
    DeviceTapGeometry_Geometry_2XE_1Y2,
    DeviceTapGeometry_Geometry_2XM_1Y,
    DeviceTapGeometry_Geometry_2XM_1Y2,
    DeviceTapGeometry_Geometry_1X_1Y2,
    DeviceTapGeometry_Geometry_1X_2YE,
    DeviceTapGeometry_Geometry_1X3_1Y,
    DeviceTapGeometry_Geometry_3X_1Y,
    DeviceTapGeometry_Geometry_1X,
    DeviceTapGeometry_Geometry_1X2,
    DeviceTapGeometry_Geometry_2X,
    DeviceTapGeometry_Geometry_2XE,
    DeviceTapGeometry_Geometry_2XM,
    DeviceTapGeometry_Geometry_1X3,
    DeviceTapGeometry_Geometry_3X,
    DeviceTapGeometry_Geometry_1X4_1Y,
    DeviceTapGeometry_Geometry_4X_1Y,
    DeviceTapGeometry_Geometry_2X2_1Y,
    DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
    DeviceTapGeometry_Geometry_1X2_2YE,
    DeviceTapGeometry_Geometry_2X_2YE,
    DeviceTapGeometry_Geometry_2XE_2YE,
    DeviceTapGeometry_Geometry_2XM_2YE,
    DeviceTapGeometry_Geometry_1X4,
    DeviceTapGeometry_Geometry_4X,
    DeviceTapGeometry_Geometry_2X2,
    DeviceTapGeometry_Geometry_2X2E,
    DeviceTapGeometry_Geometry_2X2M,
    DeviceTapGeometry_Geometry_1X8_1Y,
    DeviceTapGeometry_Geometry_8X_1Y,
    DeviceTapGeometry_Geometry_4X2_1Y,
    DeviceTapGeometry_Geometry_2X2E_2YE,
    DeviceTapGeometry_Geometry_1X8,
    DeviceTapGeometry_Geometry_8X,
    DeviceTapGeometry_Geometry_4X2,
    DeviceTapGeometry_Geometry_4X2E,
    DeviceTapGeometry_Geometry_4X2E_1Y,
    DeviceTapGeometry_Geometry_1X10_1Y,
    DeviceTapGeometry_Geometry_10X_1Y,
    DeviceTapGeometry_Geometry_1X10,
    DeviceTapGeometry_Geometry_10X,
    NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {

```

- GevPhysicalLinkConfiguration_SingleLink,
- GevPhysicalLinkConfiguration_MultiLink,
- GevPhysicalLinkConfiguration_StaticLAG,
- GevPhysicalLinkConfiguration_DynamicLAG,
- NUM_GEVPHYSICALLINKCONFIGURATION }
- enum GevCurrentPhysicalLinkConfigurationEnums {
 - GevCurrentPhysicalLinkConfiguration_SingleLink,
 - GevCurrentPhysicalLinkConfiguration_MultiLink,
 - GevCurrentPhysicalLinkConfiguration_StaticLAG,
 - GevCurrentPhysicalLinkConfiguration_DynamicLAG,
 - NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }
- enum GevIPConfigurationStatusEnums {
 - GevIPConfigurationStatus_None,
 - GevIPConfigurationStatus_PersistentIP,
 - GevIPConfigurationStatus_DHCP,
 - GevIPConfigurationStatus_LLA,
 - GevIPConfigurationStatus_ForceIP,
 - NUM_GEVIPCONFIGURATIONSTATUS }
- enum GevGVCPExtendedStatusCodesSelectorEnums {
 - GevGVCPExtendedStatusCodesSelector_Version1_1,
 - GevGVCPExtendedStatusCodesSelector_Version2_0,
 - NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }
- enum GevGVSPEExtendedIDModeEnums {
 - GevGVSPEExtendedIDMode_Off,
 - GevGVSPEExtendedIDMode_On,
 - NUM_GEVGVSPEXTENDEDIDMODE }
- enum ClConfigurationEnums {
 - ClConfiguration_Base,
 - ClConfiguration_Medium,
 - ClConfiguration_Full,
 - ClConfiguration_DualBase,
 - ClConfiguration_EightyBit,
 - NUM_CLCONFIGURATION }
- enum ClTimeSlotsCountEnums {
 - ClTimeSlotsCount_One,
 - ClTimeSlotsCount_Two,
 - ClTimeSlotsCount_Three,
 - NUM_CLTIMESLOTSCOUNT }
- enum CxpLinkConfigurationStatusEnums {
 - CxpLinkConfigurationStatus_None,
 - CxpLinkConfigurationStatus_Pending,
 - CxpLinkConfigurationStatus_CXP1_X1,
 - CxpLinkConfigurationStatus_CXP2_X1,
 - CxpLinkConfigurationStatus_CXP3_X1,
 - CxpLinkConfigurationStatus_CXP5_X1,
 - CxpLinkConfigurationStatus_CXP6_X1,
 - CxpLinkConfigurationStatus_CXP1_X2,
 - CxpLinkConfigurationStatus_CXP2_X2,
 - CxpLinkConfigurationStatus_CXP3_X2,
 - CxpLinkConfigurationStatus_CXP5_X2,
 - CxpLinkConfigurationStatus_CXP6_X2,
 - CxpLinkConfigurationStatus_CXP1_X3,
 - CxpLinkConfigurationStatus_CXP2_X3,
 - CxpLinkConfigurationStatus_CXP3_X3,
 - CxpLinkConfigurationStatus_CXP5_X3,
 - CxpLinkConfigurationStatus_CXP6_X3,
 - CxpLinkConfigurationStatus_CXP1_X4,
 - CxpLinkConfigurationStatus_CXP2_X4,

```

CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,
CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }

```

- **enum** CxpLinkConfigurationPreferredEnums {


```

CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,
CxpLinkConfigurationPreferred_CXP5_X5,
CxpLinkConfigurationPreferred_CXP6_X5,
CxpLinkConfigurationPreferred_CXP1_X6,
CxpLinkConfigurationPreferred_CXP2_X6,
CxpLinkConfigurationPreferred_CXP3_X6,
CxpLinkConfigurationPreferred_CXP5_X6,
CxpLinkConfigurationPreferred_CXP6_X6,
NUM_CXPLINKCONFIGURATIONPREFERRED }

```
- **enum** CxpLinkConfigurationEnums {


```

CxpLinkConfiguration_Auto,
CxpLinkConfiguration_CXP1_X1,
CxpLinkConfiguration_CXP2_X1,
CxpLinkConfiguration_CXP3_X1,
CxpLinkConfiguration_CXP5_X1,
CxpLinkConfiguration_CXP6_X1,
CxpLinkConfiguration_CXP1_X2,
CxpLinkConfiguration_CXP2_X2,
CxpLinkConfiguration_CXP3_X2,
CxpLinkConfiguration_CXP5_X2,
CxpLinkConfiguration_CXP6_X2,

```

```
CxpLinkConfiguration_CXP1_X3,  
CxpLinkConfiguration_CXP2_X3,  
CxpLinkConfiguration_CXP3_X3,  
CxpLinkConfiguration_CXP5_X3,  
CxpLinkConfiguration_CXP6_X3,  
CxpLinkConfiguration_CXP1_X4,  
CxpLinkConfiguration_CXP2_X4,  
CxpLinkConfiguration_CXP3_X4,  
CxpLinkConfiguration_CXP5_X4,  
CxpLinkConfiguration_CXP6_X4,  
CxpLinkConfiguration_CXP1_X5,  
CxpLinkConfiguration_CXP2_X5,  
CxpLinkConfiguration_CXP3_X5,  
CxpLinkConfiguration_CXP5_X5,  
CxpLinkConfiguration_CXP6_X5,  
CxpLinkConfiguration_CXP1_X6,  
CxpLinkConfiguration_CXP2_X6,  
CxpLinkConfiguration_CXP3_X6,  
CxpLinkConfiguration_CXP5_X6,  
CxpLinkConfiguration_CXP6_X6,  
NUM_CXPLINKCONFIGURATION }  
  
• enum CxpConnectionTestModeEnums {  
    CxpConnectionTestMode_Off,  
    CxpConnectionTestMode_Mode1,  
    NUM_CXPCONNECTIONTESTMODE }  
  
• enum CxpPoCxpStatusEnums {  
    CxpPoCxpStatus_Auto,  
    CxpPoCxpStatus_Off,  
    CxpPoCxpStatus_Tripped,  
    NUM_CXPPOCXPSTATUS }  
  
• enum 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` }

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` = 0x7FFFFFFF }

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` = 0x8013,
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,

```
ACTION_COMMAND_STATUS_ACTION_LATE = 0x8016,
ACTION_COMMAND_STATUS_ERROR = 0x8FFF }
```

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_Mixed`,
`StreamType_Custom`,
`StreamType_GEV`,
`StreamType_CL`,
`StreamType_IIDC`,
`StreamType_UVC`,
`StreamType_CXP`,
`StreamType_CLHS`,
`StreamType_U3V`,
`StreamType_ETHERNET`,
`StreamType_PCI`,
`NUMSTREAMTYPE` }

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

- enum `StreamDefaultBufferCountModeEnum` {
`StreamDefaultBufferCountMode_Manual`,
`StreamDefaultBufferCountMode_Auto`,
`NUMSTREAMDEFAULTBUFFERCOUNTMODE` }
- enum `StreamBufferCountModeEnum` {
`StreamBufferCountMode_Manual`,
`StreamBufferCountMode_Auto`,
`NUMSTREAMBUFFERCOUNTMODE` }
- enum `StreamBufferHandlingModeEnum` {
`StreamBufferHandlingMode_OldestFirst`,
`StreamBufferHandlingMode_OldestFirstOverwrite`,
`StreamBufferHandlingMode_NewestFirst`,
`StreamBufferHandlingMode_NewestFirstOverwrite`,
`StreamBufferHandlingMode_NewestOnly`,
`NUMSTREAMBUFFERHANDLINGMODE` }
- enum `DeviceTypeEnum` {
`DeviceType_Mixed`,
`DeviceType_Custom`,
`DeviceType_GEV`,
`DeviceType_CL`,
`DeviceType_IIDC`,
`DeviceType_UVC`,

- DeviceType_CXP,
- DeviceType_CLHS,
- DeviceType_U3V,
- DeviceType_ETHERNET,
- DeviceType_PCI,
- NUMDEVICETYPE }
- enum DeviceAccessStatusEnum {
 - DeviceAccessStatus_Unknown,
 - DeviceAccessStatus_ReadWrite,
 - DeviceAccessStatus_ReadOnly,
 - DeviceAccessStatus_NoAccess,
 - 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 DeviceEndiannessMechanismEnum {
 - DeviceEndiannessMechanism_Legacy,
 - DeviceEndiannessMechanism_Standard,
 - NUMDEVICEENDIANESSMECHANISM }
- enum DeviceCurrentSpeedEnum {
 - DeviceCurrentSpeed_UnknownSpeed,
 - DeviceCurrentSpeed_LowSpeed,
 - DeviceCurrentSpeed_FullSpeed,
 - DeviceCurrentSpeed_HighSpeed,
 - DeviceCurrentSpeed_SuperSpeed,
 - NUMDEVICECURRENTSPEED }
- enum POEStatusEnum {
 - POEStatus_NotSupported,
 - POEStatus_PowerOff,
 - POEStatus_PowerOn,
 - NUMPOESTATUS }
- enum FilterDriverStatusEnum {
 - FilterDriverStatus_NotSupported,
 - FilterDriverStatus_Disabled,
 - FilterDriverStatus_Enabled,
 - NUMFILTERDRIVERSTATUS }

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.
- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. Polarization images are now created through specific functions the [ImageUtilityPolarization](#) class.") PolarizationAlgorithm
- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. [Image](#) scaling can now be applied through specific functions defined in the [ImageUtility](#) class.") PolarizationResolution
- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. Please use HeatmapColor in the [Image↔UtilityHeatmap](#) class.") HeatMapColor

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

9.2 Spinnaker::GenApi Namespace Reference

Classes

- class [AutoLock](#)
- class [BooleanNode](#)
Interface for string properties.
- class [CategoryNode](#)
Interface for string properties.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.
- class [CChunkAdapterDcam](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterGeneric](#)
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.
- class [CChunkAdapterU3V](#)
Connects a chunked U3V buffer to a node map.
- class [CChunkPort](#)
Port attachable to a chunk in a buffer.
- class [CEnumerationTRef](#)
Interface for string properties.
- class [CEventAdapter](#)
Delivers Events to ports.
- class [CEventAdapter1394](#)
Distribute the events to the node map.
- class [CEventAdapterGeneric](#)
Connects a generic event to a node map.
- class [CEventAdapterGEV](#)
Connects a GigE [Event](#) to a node map.
- class [CEventAdapterU3V](#)
Connects a U3V [Event](#) to a node map.
- class [CEventPort](#)
Port attachable to an event.
- class [CFeatureBag](#)
Bag holding streamable features of a nodetree.
- class [CFloatPtr](#)
SmartPointer for IFloat interface pointer.
- class [CGeneric_XMLLoaderParams](#)
Empty base class used by class [CNodeMapRef](#) as generic template argument.
- class [CLock](#)
A lock class.
- class [CLockEx](#)

- This class is for testing purposes only.*
- class [CNodeCallback](#)
callback body instance for INode pointers
- class [CNodeMapFactory](#)
The node map factory is used for creating node maps from camera description files.
- class [CNodeMapRef](#)
Smartpointer for NodeMaps with create function.
- class [CNodeMapRefT](#)
Smartpointer template for NodeMaps with create function.
- class [CommandNode](#)
Interface for string properties.
- class [Counter](#)
Definition of a simple Counter class.
- class [CPointer](#)
Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.
- class [CPortImpl](#)
Standard implementation for a port.
- class [CPortWriteList](#)
Container holding a list of port write commands.
- class [CRegisterPortImpl](#)
Standard implementation for a port using a register based transport layer.
- class [CSelectorSet](#)
The set of selectors selecting a given node.
- class [CTestPortStruct](#)
Implements a register spaces based on a C++ struct.
- class [double_autovector_t](#)
Vector of doubles with reference counting.
- class [EAccessModeClass](#)
Holds conversion methods for the access mode enumeration.
- class [ECachingModeClass](#)
Holds conversion methods for the caching mode enumeration.
- class [EDisplayNotationClass](#)
Holds conversion methods for the notation type of floats.
- class [EEndianessClass](#)
Holds conversion methods for the endianess enumeration.
- class [EGenApiSchemaVersionClass](#)
helper class converting EGenApiSchemaVersion from and to string
- class [EInputDirectionClass](#)
Holds conversion methods for the notation type of floats.
- class [ENamespaceClass](#)
Holds conversion methods for the namespace enumeration.
- class [EnumEntryNode](#)
Interface for string properties.
- class [EnumNode](#)
Interface for string properties.
- class [ERepresentationClass](#)
Holds conversion methods for the representation enumeration.
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [ESlopeClass](#)
Holds conversion methods for the converter formulas.

- class [EStandardNameSpaceClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [EVisibilityClass](#)
Holds conversion methods for the visibility enumeration.
- class [EYesNoClass](#)
Holds conversion methods for the standard namespace enumeration.
- class [FileProtocolAdapter](#)
Adapter between the std::iostreambuf and the SFNC Features representing the device file system.
- class [FloatNode](#)
Interface for string properties.
- class [FloatRegNode](#)
Interface for string properties.
- class [Function_NodeCallback](#)
Container for a function pointer.
- class [IDevFileStreamBase](#)
- class [IDevFileStreamBuf](#)
- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [IntegerNode](#)
Interface for string properties.
- class [IntRegNode](#)
Interface for string properties.
- class [Member_NodeCallback](#)
Container for a member function pointer.
- class [Node](#)
class common to all nodes
- class [NodeMap](#)
Smart pointer template for NodeMaps with create function.
- class [ODevFileStreamBase](#)
- class [ODevFileStreamBuf](#)
- class [PortNode](#)
Interface for value properties.
- class [PortRecorder](#)
Interface for recording write commands on a port.
- class [PortReplay](#)
Interface for replaying write commands on a port.
- class [RegisterNode](#)
Interface for string properties.
- class [SpinTestCamera](#)
- class [StringNode](#)
Interface for string properties.
- class [StringRegNode](#)
Interface for string properties.
- class [ValueNode](#)
Interface for value properties.

Typedefs

- typedef [BooleanNode](#) [CBooleanRef](#)
- typedef [CategoryNode](#) [CCategoryRef](#)
- typedef [CommandNode](#) [CCommandRef](#)
- typedef [EnumEntryNode](#) [CEnumEntryRef](#)
- typedef [EnumNode](#) [CEnumerationRef](#)
- typedef [ODevFileStreamBase](#)< char, std::char_traits< char > > [ODevFileStream](#)
- typedef [IDevFileStreamBase](#)< char, std::char_traits< char > > [IDevFileStream](#)
- typedef [FloatNode](#) [CFloatRef](#)
- typedef node_vector [NodeList_t](#)
a list of node references
- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes
- typedef [IntegerNode](#) [CIntegerRef](#)
- typedef [Node](#) [CNodeRef](#)
- typedef [Node](#) [CSelectorRef](#)
- typedef [NodeMap](#) [CNodeMapRef](#)
- typedef [CPointer](#)< [IBase](#) > [CBasePtr](#)
SmartPointer for IBase interface pointer.
- typedef [CPointer](#)< [INode](#), [IBase](#) > [CNodePtr](#)
SmartPointer for INode interface pointer.
- typedef [CPointer](#)< [IValue](#) > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef [CPointer](#)< [ICategory](#) > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef [CPointer](#)< [IBoolean](#) > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef [CPointer](#)< [IInteger](#) > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef [CPointer](#)< [IString](#) > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef [CPointer](#)< [IRegister](#) > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef [CPointer](#)< [IEnumeration](#) > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef [CPointer](#)< [IEnumEntry](#) > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef [CPointer](#)< [IPort](#) > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef [CPointer](#)< [IPortReplay](#) > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef [CPointer](#)< [IPortRecorder](#) > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef [CPointer](#)< [IPortWriteList](#), [IPortWriteList](#) > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef [CPointer](#)< [IChunkPort](#) > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef [CPointer](#)< [INodeMap](#), [INodeMap](#) > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef [CPointer](#)< [INodeMapDyn](#), [INodeMap](#) > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.

- typedef [CPointer](#)< [IDeviceInfo](#), [INodeMap](#) > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.
- typedef [CPointer](#)< [ISelector](#) > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef [CPointer](#)< [ICommand](#) > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef [CPointer](#)< [IPortConstruct](#) > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.
- typedef [PortNode](#) [CPortRef](#)
- typedef [PortRecorder](#) [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.
- typedef [RegisterNode](#) [CRegisterRef](#)
- typedef [StringNode](#) [CStringRef](#)
- typedef [GenlCam::gcstring_vector](#) [StringList_t](#)
A list of strings.
- typedef [ValueNode](#) [CValueRef](#)

Enumerations

- enum [GVCP_MESSAGE_TAGS](#) {
 [TAG_EVENT_CMD](#) = 0xc0,
 [TAG_EVENTDATA_CMD](#) = 0xc2 }
- enum [ECallbackType](#) {
 [cbPostInsideLock](#) = 1,
 [cbPostOutsideLock](#) = 2 }
the type of callback
- enum [ECacheUsage_t](#) {
 [CacheUsage_Automatic](#),
 [CacheUsage_ForceWrite](#),
 [CacheUsage_ForceRead](#),
 [CacheUsage_Ignore](#) }
- Lists the cache usage strategies.*
- enum [EContentType_t](#) {
 [ContentType_Xml](#),
 [ContentType_ZippedXml](#) }
- Lists the processable file types.*
- enum [ESign](#) {
 [Signed](#),
 [Unsigned](#),
 [_UndefinedSign](#) }
- signed or unsigned integers*
- enum [EAccessMode](#) {
 [NI](#),
 [NA](#),
 [WO](#),
 [RO](#),
 [RW](#),
 [_UndefinedAccesMode](#),
 [_CycleDetectAccesMode](#) }
- access mode of a node*

- enum [EVisibility](#) {
[Beginner](#) = 0,
[Expert](#) = 1,
[Guru](#) = 2,
[Invisible](#) = 3,
[_UndefinedVisibility](#) = 99 }
recommended visibility of a node
- enum [ECachingMode](#) {
[NoCache](#),
[WriteThrough](#),
[WriteAround](#),
[_UndefinedCachingMode](#) }
caching mode of a register
- enum [ERepresentation](#) {
[Linear](#),
[Logarithmic](#),
[Boolean](#),
[PureNumber](#),
[HexNumber](#),
[IPV4Address](#),
[MACAddress](#),
[_UndefinedRepresentation](#) }
recommended representation of a node value
- enum [EEndianness](#) {
[BigEndian](#),
[LittleEndian](#),
[_UndefinedEndian](#) }
Endianness of a value in a register.
- enum [ENamespace](#) {
[Custom](#),
[Standard](#),
[_UndefinedNameSpace](#) }
Defines if a node name is standard or custom.
- enum [EStandardNameSpace](#) {
[None](#),
[GEV](#),
[IIDC](#),
[CL](#),
[USB](#),
[_UndefinedStandardNameSpace](#) }
Defines from which standard namespace a node name comes from.
- enum [EYesNo](#) {
[Yes](#) = 1,
[No](#) = 0,
[_UndefinedYesNo](#) = 2 }
Defines the choices of a Yes/No alternative.
- enum [ESlope](#) {
[Increasing](#),
[Decreasing](#),
[Varying](#),
[Automatic](#),
[_UndefinedESlope](#) }
typedef for formula type
- enum [EXMLValidation](#) {
[xvLoad](#) = 0x00000001L,
[xvCycles](#) = 0x00000002L,

```

xvSFNC = 0x00000004L,
xvDefault = 0x00000000L,
xvAll = 0xffffffffL,
_UndefinedEXMLValidation = 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](#) {
[intflValue](#),
[intflBase](#),
[intflInteger](#),
[intflBoolean](#),
[intflCommand](#),
[intflFloat](#),
[intflString](#),
[intflRegister](#),
[intflCategory](#),
[intflEnumeration](#),
[intflEnumEntry](#),
[intflPort](#) }

typedef for interface type

- enum [ELinkType](#) {
[ctParentNodes](#),
[ctReadingChildren](#),
[ctWritingChildren](#),
[ctInvalidatingChildren](#),
[ctDependingNodes](#),
[ctTerminalNodes](#) }

typedef for link type

- enum [EIncMode](#) {
[noIncrement](#),
[fixedIncrement](#),
[listIncrement](#) }

typedef for increment mode

- enum [EInputDirection](#) {
[idFrom](#),
[idTo](#),
[idNone](#) }

typedef for link type

- enum [EGenApiSchemaVersion](#) {
[v1_0](#) = 1,
[v1_1](#) = 2,
[_Undefined](#) = -1 }

GenApi schema version.

Functions

- void [SPINNAKER_API SET_GUID](#) (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)
- virtual void [operator=](#) (bool Value)

Set node value.

- virtual bool [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false) const =0
Get node value.
- virtual bool [operator\(\)](#) () const
Get node value.
- virtual [EYesNo](#) [CacheChunkData](#) () const =0
Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual bool [IsDone](#) (bool [Verify](#)=true)=0
Query whether the command is executed.
- virtual [GenICam::gcstring](#) [GetVendorName](#) ()=0
Get the vendor name.
- virtual [GenICam::gcstring](#) [GetToolTip](#) ()=0
Get tool tip.
- virtual [GenICam::gcstring](#) [GetStandardNameSpace](#) ()=0
Get the standard name space.
- virtual void [GetGenApiVersion](#) ([GenICam::Version_t](#) &Version, uint16_t &Build)=0
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) ([GenICam::Version_t](#) &Version)=0
Get the schema version number.
- virtual void [GetDeviceVersion](#) ([GenICam::Version_t](#) &Version)=0
Get the version of the device description file.
- virtual [GenICam::gcstring](#) [GetProductGuid](#) ()=0
Get the Guid describing the product.
- virtual [GenICam::gcstring](#) [GetVersionGuid](#) ()=0
Get the Guid describing the product version.
- virtual [GenICam::gcstring](#) [GetSymbolic](#) () const =0
Get symbolic enum value.
- virtual double [GetNumericValue](#) ()=0
Get double number associated with the entry.
- virtual bool [IsSelfClearing](#) ()=0
Indicates if the corresponding EnumEntry is self clearing.
- virtual void [GetEntries](#) ([NodeList_t](#) &Entries)=0
Get list of entry nodes.
- virtual [IEnumeration](#) & [operator=](#) (const [GenICam::gcstring](#) &ValueStr)=0
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool [Verify](#)=true)=0
Set integer node value.
- virtual [GenICam::gcstring](#) [operator*](#) ()=0
Get string node value.
- virtual int64_t [GetIntValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get integer node value.
- virtual [IEnumEntry](#) * [GetEntryByName](#) (const [GenICam::gcstring](#) &Symbolic)=0
Get an entry node by name.
- virtual [IEnumEntry](#) * [GetEntry](#) (const int64_t IntValue)=0
Get an entry node by its IntValue.
- virtual [IEnumEntry](#) * [GetCurrentEntry](#) (bool [Verify](#)=false, bool IgnoreCache=false)=0
Get the current entry.
- virtual [IEnumeration](#) & [operator=](#) (EnumT Value)=0
Set node value.
- virtual [IEnumEntry](#) * [GetEntry](#) (const EnumT Value)=0
returns the EnumEntry object belonging to the Value
- virtual [IFloat](#) & [operator=](#) (double Value)=0

- Set node value.*

 - virtual double [GetMin](#) ()=0
 - Get minimum value allowed.*
 - virtual double [GetMax](#) ()=0
 - Get maximum value allowed.*
 - virtual bool [HasInc](#) ()=0
 - True if the float has a constant increment.*
 - virtual [EIncMode](#) [GetIncMode](#) ()=0
 - Get increment mode.*
 - virtual double [GetInc](#) ()=0
 - Get the constant increment if there is any.*
 - virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)=0
 - Get list of valid value.*
 - virtual [ERepresentation](#) [GetRepresentation](#) ()=0
 - Get recommended representation.*
 - virtual [GenICam::gcstring](#) [GetUnit](#) () const =0
 - Get the physical unit name.*
 - virtual [EDisplayNotation](#) [GetDisplayNotation](#) () const =0
 - Get the way the float should be converted to a string.*
 - virtual [int64_t](#) [GetDisplayPrecision](#) () const =0
 - Get the precision to be used when converting the float to a string.*
 - virtual void [ImposeMin](#) (double Value)=0
 - Restrict minimum value.*
 - virtual void [ImposeMax](#) (double Value)=0
 - Restrict maximum value.*
 - virtual [Integer](#) & [operator=](#) (int64_t Value)=0
 - Set node value.*
 - virtual void [ImposeMin](#) (int64_t Value)=0
 - Restrict minimum value.*
 - virtual void [ImposeMax](#) (int64_t Value)=0
 - Restrict maximum value.*
 - virtual [GenApi::ENamespace](#) [GetNamespace](#) () const =0
 - Get name space.*
 - virtual [EVisibility](#) [GetVisibility](#) () const =0
 - Get the recommended visibility of the node.*
 - virtual void [InvalidateNode](#) ()=0
 - Indicates that the node's value may have changed.*
 - virtual bool [IsCacheable](#) () const =0
 - Is the node value cacheable.*
 - virtual [EYesNo](#) [IsAccessModeCacheable](#) () const =0
 - True if the AccessMode can be cached.*
 - virtual [ECachingMode](#) [GetCachingMode](#) () const =0
 - Get Caching Mode.*
 - virtual [int64_t](#) [GetPollingTime](#) () const =0
 - recommended polling time (for non-cacheable nodes)*
 - virtual [GenICam::gcstring](#) [GetDescription](#) () const =0
 - Get a long description of the node.*
 - virtual [GenICam::gcstring](#) [GetDisplayName](#) () const =0
 - Get a name string for display.*
 - virtual [GenICam::gcstring](#) [GetDeviceName](#) () const =0
 - Get a name of the device.*

- virtual void [GetChildren](#) ([GenApi::NodeList_t](#) &Children, [ELinkType](#) LinkType=[ctReadingChildren](#)) const =0
Get all nodes this node directly depends on.
- virtual void [GetParents](#) ([GenApi::NodeList_t](#) &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual [CallbackHandleType](#) [RegisterCallback](#) ([CNodeCallback](#) *pCallback)=0
Register change callback Takes ownership of the [CNodeCallback](#) object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)=0
De register change callback Destroys [CNodeCallback](#) object.
- virtual [INodeMap](#) * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual [GenICam::gcstring](#) [GetEventId](#) () const =0
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const =0
True if the node is streamable.
- virtual void [GetPropertyNames](#) ([GenICam::gcstring_vector](#) &PropertyNames) const =0
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const [GenICam::gcstring](#) &PropertyName, [GenICam::gcstring](#) &ValueStr, [GenICam::gcstring](#) &AttributeStr)=0
Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) ([EAccessMode](#) ImposedAccessMode)=0
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) ([EVisibility](#) ImposedVisibility)=0
Imposes a visibility to the natural visibility of the node.
- virtual [INode](#) * [GetAlias](#) () const =0
Retrieves the a node which describes the same feature in a different way.
- virtual [INode](#) * [GetCastAlias](#) () const =0
Retrieves the a node which describes the same feature so that it can be casted.
- virtual [GenICam::gcstring](#) [GetDocuURL](#) () const =0
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const =0
True if the node should not be used any more.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const =0
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const =0
True if the node can be reached via category nodes from a category node named "Root".
- virtual bool [operator==](#) (int nullptr) const =0
- virtual bool [operator!=](#) (int nullptr) const =0
- bool [IsReadable](#) ([EAccessMode](#) AccessMode)
Tests if readable.
- bool [IsReadable](#) (const [IBase](#) *p)
Checks if a node is readable.
- bool [IsReadable](#) (const [IBase](#) &r)
Checks if a node is readable.
- bool [IsWritable](#) ([EAccessMode](#) AccessMode)
Tests if writable.
- bool [IsWritable](#) (const [IBase](#) *p)
Checks if a node is writable.
- bool [IsWritable](#) (const [IBase](#) &r)
Checks if a node is writable.
- bool [IsImplemented](#) ([EAccessMode](#) AccessMode)

- Tests if implemented.*

 - bool [IsImplemented](#) (const [IBase](#) *p)

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

Checks if a node is implemented.
- bool [IsAvailable](#) ([EAccessMode](#) AccessMode)

Tests if available.
- bool [IsAvailable](#) (const [IBase](#) *p)

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

Checks if a node is available.
- [EAccessMode](#) [Combine](#) ([EAccessMode](#) Peter, [EAccessMode](#) Paul)

Computes which access mode the two guards allow together.
- bool [IsVisible](#) ([EVisibility](#) Visibility, [EVisibility](#) MaxVisibility)

Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- [EVisibility](#) [Combine](#) ([EVisibility](#) Peter, [EVisibility](#) Paul)

Computes which visibility the two guards allow together.
- bool [IsCacheable](#) ([ECachingMode](#) CachingMode)

Tests Cacheability.
- [ECachingMode](#) [Combine](#) ([ECachingMode](#) Peter, [ECachingMode](#) Paul)

Computes which CachingMode results from a combination.
- virtual [INode](#) * [GetNode](#) (const [GenICam::gcstring](#) &Name) const =0

Retrieves the node from the central map by Name.
- virtual void [InvalidateNodes](#) () const =0

Invalidates all nodes.
- virtual bool [Connect](#) ([IPort](#) *pPort, const [GenICam::gcstring](#) &PortName) const =0

Connects a port to a port node with given name.
- virtual bool [Connect](#) ([IPort](#) *pPort) const =0

Connects a port to the standard port "Device".
- virtual void [Poll](#) (int64_t ElapsedTime)=0

Fires nodes which have a polling time.
- virtual [CLock](#) & [GetLock](#) () const =0

Returns the lock which guards the node map.
- virtual uint64_t [GetNumNodes](#) () const =0

Get the number of nodes in the map.
- virtual void [LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)=0

Loads an XML from a file.
- virtual void [LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)=0

Loads an XML from a file with injection.
- virtual void [LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)=0

Loads an XML from a string.
- virtual void [LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)=0

Loads an XML from a string with injection.
- virtual void [PreprocessXMLFromFile](#) (const [GenICam::gcstring](#) &XMLFileName, const [GenICam::gcstring](#) &StyleSheetFileName, const [GenICam::gcstring](#) &OutputFileName, const uint32_t XMLValidation=[xv↔Default](#))=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)
make a new callback object for member functions
- template<class Client , class Member >
intptr_t [Register](#) (INode *pNode, Client &c, Member m, [ECallbackType](#) CallbackType=[cbPostInsideLock](#))
Register a C++-member function a callback.
- [SPINNAKER_API](#) void [Deregister](#) ([GenApi::CallbackHandleType](#) pCallbackInfo)
Unregistering callback by handle.
- [SPINNAKER_API](#) IDestroy * [CastToIDestroy](#) (INodeMap *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- template<class TCameraParams >
void [_LoadXMLFromFile](#) (const [GenICam::gcstring](#) &FileName)
- template<class TCameraParams >
void [_LoadXMLFromZIPFile](#) (const [GenICam::gcstring](#) &ZipFileName)
- template<class TCameraParams >
void [_LoadXMLFromFileInject](#) (const [GenICam::gcstring](#) &TargetFileName, const [GenICam::gcstring](#) &InjectFileName)
- template<class TCameraParams >
void [_LoadXMLFromString](#) (const [GenICam::gcstring](#) &XMLData)
- template<class TCameraParams >
void [_LoadXMLFromZIPData](#) (const void *zipData, size_t zipSize)
- template<class TCameraParams >
void [_LoadXMLFromStringInject](#) (const [GenICam::gcstring](#) &TargetXMLData, const [GenICam::gcstring](#) &InjectXMLData)
- template<class TCameraParams >
void [_GetSupportedSchemaVersions](#) ([GenICam::gcstring_vector](#) &SchemaVersions)
- template<class TCameraParams >
[GenICam::gcstring](#) [_GetDeviceName](#) ()
- template<class TCameraParams >
void [_Poll](#) (int64_t ElapsedTime)
- template<class TCameraParams >
void [_GetNodes](#) ([NodeList_t](#) &Nodes)
- template<class TCameraParams >
INode * [_GetNode](#) (const [GenICam::gcstring](#) &key)
- template<class TCameraParams >
void [_InvalidateNodes](#) ()
- template<class TCameraParams >
bool [_Connect](#) (IPort *pPort, const [GenICam::gcstring](#) &PortName)
- template<class TCameraParams >
bool [_Connect](#) (IPort *pPort)
- template<class TCameraParams >
bool [_ClearXMLCache](#) ()
- virtual void [PersistFeature](#) (IValue &item)=0

Stores a feature.
- [SPINNAKER_API](#) std::istream & [EatComments](#) (std::istream &is)

- Helper function ignoring lines starting with comment character '#'.*
- [SPINNAKER_API](#) `std::istream & operator>> (std::istream &is, CFeatureBag &FeatureBag)`
Reads in persistent data from a stream.
- [SPINNAKER_API](#) `std::ostream & operator<< (std::ostream &os, const CFeatureBag &FeatureBag)`
writes out persistent data to a stream
- `template<class T, class B >`
`bool IsReadable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is readable.
- `template<class T, class B >`
`bool IsWritable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Writable.
- `template<class T, class B >`
`bool IsImplemented (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Implemented.
- `template<class T, class B >`
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Available.
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`
Returns the name of the main interface as string DEPRICATED, use [IBase::GetPrincipalInterfaceType\(\)](#) instead.
- `virtual void SetNumEnums (int NumEnums)=0`
sets the number of enum values

Variables

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

- `template<typename EnumT >`
`interface SPINNAKER_API_ABSTRACT` virtual public `IEnumReference`
Interface to construct an enum reference.
- `interface SPINNAKER_API_ABSTRACT IFloat`
Interface for float properties.
- `interface SPINNAKER_API_ABSTRACT IInteger`
Interface for integer properties.
- `interface SPINNAKER_API_ABSTRACT INode`
Interface common to all nodes.
- `interface SPINNAKER_API_ABSTRACT` virtual public `IReference`
Interface to construct a reference.
- `interface SPINNAKER_API_ABSTRACT INodeMap`
Interface to access the node map.
- `interface SPINNAKER_API_ABSTRACT INodeMapDyn`
Interface to access the node map.
- `interface SPINNAKER_API_ABSTRACT IPort`
Interface for ports.
- `interface SPINNAKER_API_ABSTRACT int64_t Address`
- `interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0`
- `interface SPINNAKER_API IPortConstruct`
Interface for ports.
- `interface SPINNAKER_API_ABSTRACT IPortWriteList`
- `interface SPINNAKER_API_ABSTRACT IPortReplay`
Interface for replaying write commands on a port.
- `interface SPINNAKER_API_ABSTRACT bool Invalidate = true) = 0`
- `interface SPINNAKER_API_ABSTRACT IPortRecorder`
Interface for recording write commands on a port.
- `interface SPINNAKER_API_ABSTRACT IRegister`
Interface for registers.
- `interface SPINNAKER_API_ABSTRACT ISelector`
Interface for groups of features selected by a single one.
- `interface SPINNAKER_API_ABSTRACT ISelectorDigit`
Interface of a "digit" of the "counter" formed by the selector set.
- `interface SPINNAKER_API_ABSTRACT IString`
Interface for string properties.
- `interface SPINNAKER_API_ABSTRACT IValue`
Interface for value properties.
- `interface SPINNAKER_API_ABSTRACT IPersistScript`
Basic interface to persist values to.

9.2.1 Typedef Documentation

9.2.1.1 IDevFileStream

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

9.2.1.2 ODevFileStream

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

9.2.2 Enumeration Type Documentation

9.2.2.1 GVCP_MESSAGE_TAGS

```
enum GVCP_MESSAGE_TAGS
```

Enumerator

| | |
|-------------------|--|
| TAG_EVENT_CMD | |
| TAG_EVENTDATA_CMD | |

9.2.3 Function Documentation

9.2.3.1 PersistFeature()

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

Stores a feature.

9.2.3.2 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 )
```

9.2.4 Variable Documentation

9.2.4.1 COMMAND_MAGIC

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

9.2.4.2 GENCP_COMMAND_HEADER_SIZE

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

9.2.4.3 GENCP_EVENT_BASIC_SIZE

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

9.2.4.4 GENCP_EVENT_CMD_ID

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

9.2.4.5 IPersistScript

```
interface SPINNAKER_API_ABSTRACT IPersistScript
```

Initial value:

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

Basic interface to persist values to.

9.2.4.6 U3V_EVENT_PREFIX

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

9.3 Spinnaker::GenICam Namespace Reference

Classes

- class [AutoLock](#)
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [gcstring](#)
- class [LockableObject](#)
Instance-Lock for an object.
- struct [Version_t](#)
Version.

Functions

- [SPINNAKER_API](#) void [ThrowBadAlloc](#) ()
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str)
STL getline.
- std::istream & [getline](#) (std::istream &is, [Spinnaker::GenICam::gcstring](#) &str, char delim)
STL getline.
- template<typename Td, typename Ts >
Td [INTEGRAL_CAST2](#) (Ts s)
This verifies at runtime if there was no loss of data if an type Ts (e.g.
- template<typename T >
T [INTEGRAL_CAST](#) (int64_t ll)
This verifies at runtime if there was no loss of data if an int64_t was downcast to type T (e.g.
- [SPINNAKER_API](#) bool [DoesEnvironmentVariableExist](#) (const [Spinnaker::GenICam::gcstring](#) &VariableName)
Returns true if an environment variable exists.
- [SPINNAKER_API](#) [gcstring](#) [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) bool [GetValueOfEnvironmentVariable](#) (const [gcstring](#) &VariableName, [gcstring](#) &VariableContent)
Retrieve the value of an environment variable.
- [SPINNAKER_API](#) [gcstring](#) [UrlEncode](#) (const [gcstring](#) &Input)
Converts \ to / and replaces all unsafe characters by their xx equivalent.
- [SPINNAKER_API](#) [gcstring](#) [UrlDecode](#) (const [gcstring](#) &Input)
Replaces xx escapes by their char equivalent.
- [SPINNAKER_API](#) void [ReplaceEnvironmentVariables](#) ([gcstring](#) &Buffer, bool ReplaceBlankBy20=false)
Replaces in a string and replace ' ' with %20.
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamCacheFolder](#) (void)
Retrieve the path of the GenICam cache folder The path to the cache folder can be stored by calling [SetGenICamCacheFolder\(\)](#).
- [SPINNAKER_API](#) [gcstring](#) [GetGenICamLogConfig](#) (void)

- Retrieve the path of the *GenICam* logging properties file.
- **SPINNAKER_API** `gcstring GetGenICamCLProtocolFolder` (void)
Retrieve the path of the CLProtocol folder The path to the CLProtocol folder can be stored by calling *SetGenICamCLProtocolFolder()*.
- **SPINNAKER_API** `void SetGenICamCacheFolder` (const `gcstring` &path)
Stores the path of the *GenICam* cache folder.
- **SPINNAKER_API** `void SetGenICamLogConfig` (const `gcstring` &path)
Stores the path of the *GenICam* logging properties file.
- **SPINNAKER_API** `void SetGenICamCLProtocolFolder` (const `gcstring` &path)
Stores the path of the CLProtocol folder.
- **SPINNAKER_API** `void Tokenize` (const `gcstring` &str, `gcstring_vector` &tokens, const `gcstring` &delimiters=" ")
splits str input string into a list of tokens using the delimiter
- **SPINNAKER_API** `void GetFiles` (const `gcstring` &FileTemplate, `gcstring_vector` &FileNames, const bool DirectoriesOnly=false)
Gets a list of files or directories matching a given FileTemplate.
- **SPINNAKER_API** `gcstring GetModulePathFromFunction` (void *pFunction)
Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

9.3.1 Function Documentation

9.3.1.1 `getline()` [1/2]

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

STL `getline`.

9.3.1.2 `getline()` [2/2]

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

STL `getline`.

9.3.1.3 `ThrowBadAlloc()`

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

9.4 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 10

Class Documentation

10.1 ActionCommandResult Struct Reference

Action Command Result.

Public Attributes

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

10.1.1 Detailed Description

Action Command Result.

10.1.2 Member Data Documentation

10.1.2.1 DeviceAddress

`unsigned int DeviceAddress`

10.1.2.2 Status

`ActionCommandStatus Status`

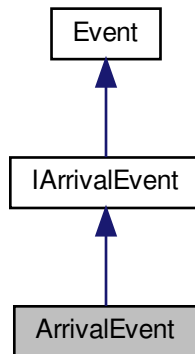
The documentation for this struct was generated from the following file:

- `include/SpinnakerDefs.h`

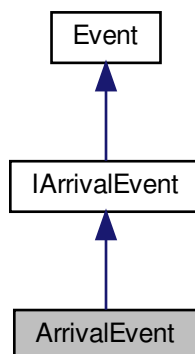
10.2 ArrivalEvent Class Reference

An event handler for capturing the device arrival event.

Inheritance diagram for ArrivalEvent:



Collaboration diagram for ArrivalEvent:



Public Member Functions

- [ArrivalEvent](#) ()
Default constructor.
- virtual [~ArrivalEvent](#) ()
Virtual destructor.
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0
Callback to the device arrival event.

Protected Member Functions

- [ArrivalEvent](#) & `operator=` (const [ArrivalEvent](#) &)
Assignment operator.

Additional Inherited Members

10.2.1 Detailed Description

An event handler for capturing the device arrival event.

10.2.2 Constructor & Destructor Documentation

10.2.2.1 `ArrivalEvent()`

```
ArrivalEvent ( )
```

Default constructor.

10.2.2.2 `~ArrivalEvent()`

```
virtual ~ArrivalEvent ( ) [virtual]
```

Virtual destructor.

10.2.3 Member Function Documentation

10.2.3.1 `OnDeviceArrival()`

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device arrival event.

Implements [IArrivalEvent](#).

10.2.3.2 operator=()

```
ArrivalEvent& operator= (
    const ArrivalEvent & ) [protected]
```

Assignment operator.

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

- include/ArrivalEvent.h

10.3 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.

10.3.1 Detailed Description

Delivers information about the attached chunks and nodes.

10.3.2 Member Data Documentation

10.3.2.1 NumAttachedChunks

```
int NumAttachedChunks
```

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

10.3.2.2 NumChunkPorts

```
int NumChunkPorts
```

Number of chunk ports found in the node map.

10.3.2.3 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`

10.4 AutoLock Class Reference

Public Member Functions

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

10.4.1 Constructor & Destructor Documentation

10.4.1.1 AutoLock()

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

10.4.1.2 ~AutoLock()

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

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

- `include/SpinGenApi/Synch.h`

10.5 AutoLock Class Reference

Public Member Functions

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

10.5.1 Constructor & Destructor Documentation

10.5.1.1 AutoLock()

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

10.5.1.2 ~AutoLock()

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

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

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

10.6 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.

10.6.1 Detailed Description

Options for saving AVI files.

10.6.2 Constructor & Destructor Documentation

10.6.2.1 AVIOption()

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

10.6.3 Member Data Documentation

10.6.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

10.6.3.2 reserved

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

Reserved for future use.

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

- [include/SpinVideoDefs.h](#)

10.7 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.

10.7.1 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.

10.7.2 Constructor & Destructor Documentation

10.7.2.1 [BasePtr](#)() [1/2]

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

Default constructor.

10.7.2.2 [~BasePtr](#)()

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


10.7.2.3 BasePtr() [2/2]

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

Copy constructor.

10.7.3 Member Function Documentation

10.7.3.1 get()

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

[get\(\)](#)

10.7.3.2 IsValid()

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

True if the pointer is valid.

10.7.3.3 operator bool()

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

True if the pointer is valid.

10.7.3.4 operator T*()

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

Dereferencing.

10.7.3.5 operator->()

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

Dereferencing.

10.7.3.6 operator=() [1/4]

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

Assign INode Pointer.

10.7.3.7 operator=() [2/4]

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

10.7.3.8 operator=() [3/4]

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

10.7.3.9 operator=() [4/4]

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

10.7.3.10 operator==() [1/4]

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

Pointer equal.

10.7.3.11 operator==() [2/4]

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

Pointer equal.

10.7.3.12 operator==() [3/4]

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

Pointer equal.

10.7.3.13 operator==() [4/4]

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

Pointer equal.

10.7.4 Member Data Documentation**10.7.4.1 m_pT**

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

Underlying raw pointer.

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

- include/[BasePtr.h](#)

10.8 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.

10.8.1 Detailed Description

Options for saving Bitmap image.

10.8.2 Constructor & Destructor Documentation

10.8.2.1 BMPOption()

```
BMPOption ( ) [inline]
```

10.8.3 Member Data Documentation

10.8.3.1 indexedColor_8bit

```
bool indexedColor_8bit
```

10.8.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

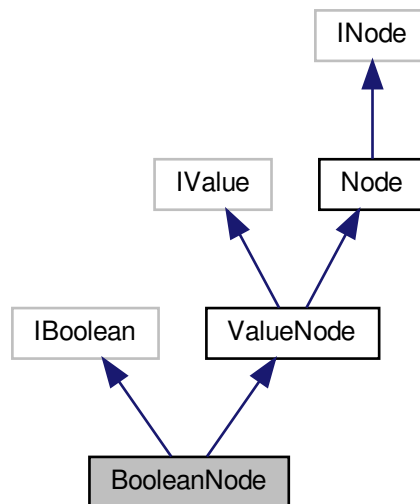
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

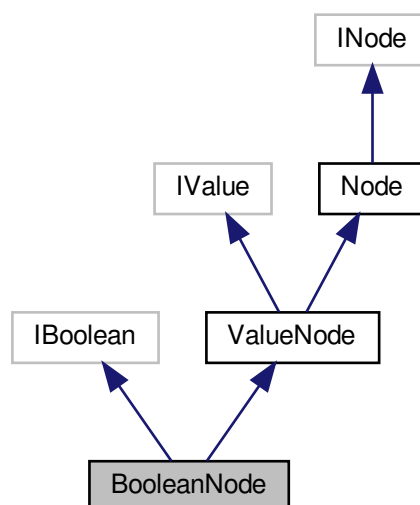
10.9 BooleanNode Class Reference

[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

10.9.1 Detailed Description

[Interface](#) for string properties.

10.9.2 Constructor & Destructor Documentation

10.9.2.1 BooleanNode() [1/2]

```
BooleanNode ( )
```

10.9.2.2 BooleanNode() [2/2]

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

10.9.2.3 ~BooleanNode()

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

10.9.3 Member Function Documentation

10.9.3.1 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.

10.9.3.2 operator=()

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

Set node value.

10.9.3.3 SetReference()

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

overload SetReference for Value

Reimplemented from [ValueNode](#).

10.9.3.4 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). |

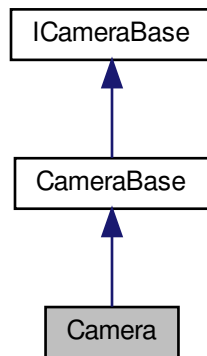
The documentation for this class was generated from the following file:

- `include/SpinGenApi/BooleanNode.h`

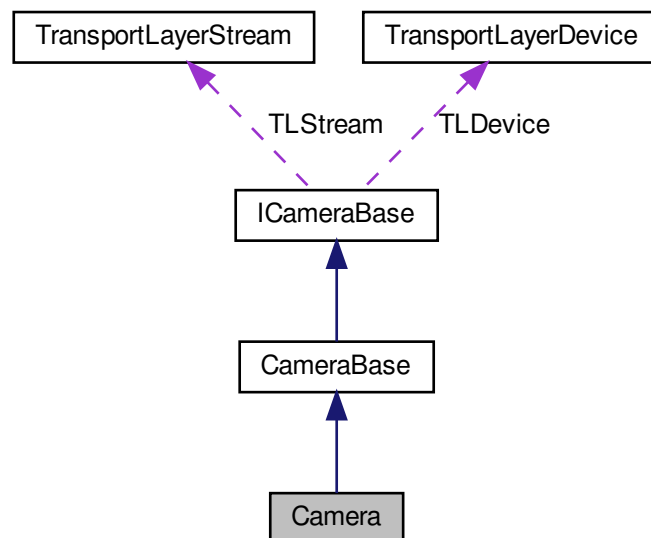
10.10 Camera Class Reference

The camera object class.

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

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

Public Attributes

- [GenApi::Integer](#) & [LUTIndex](#)
 Description:
Control the index (offset) of the coefficient to access in the selected LUT.
- [GenApi::Boolean](#) & [LUTEnable](#)
 Description:
Activates the selected LUT.
- [GenApi::Integer](#) & [LUTValue](#)
 Description:
Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.
- [GenApi::EnumerationT< LUTSelectorEnums >](#) & [LUTSelector](#)
 Description:
Selects which LUT to control.
- [GenApi::Float](#) & [ExposureTime](#)
 Description:
Exposure time in microseconds when Exposure Mode is Timed.
- [GenApi::Command](#) & [AcquisitionStop](#)
 Description: *This command stops the acquisition of images.*
- [GenApi::Float](#) & [AcquisitionResultingFrameRate](#)
 Description: *Resulting frame rate in Hertz.*
- [GenApi::Float](#) & [AcquisitionLineRate](#)
 Description: *Controls the rate (in Hertz) at which the Lines in a Frame are captured.*
- [GenApi::Command](#) & [AcquisitionStart](#)
 Description: *This command starts the acquisition of images.*
- [GenApi::Command](#) & [TriggerSoftware](#)
 Description:
Generates an internal trigger if Trigger Source is set to Software.
- [GenApi::EnumerationT< ExposureModeEnums >](#) & [ExposureMode](#)
 Description:
Sets the operation mode of the Exposure.
- [GenApi::EnumerationT< AcquisitionModeEnums >](#) & [AcquisitionMode](#)
 Description: *Sets the acquisition mode of the device.*
- [GenApi::Integer](#) & [AcquisitionFrameCount](#)
 Description:
Number of images to acquire during a multi frame acquisition.
- [GenApi::EnumerationT< TriggerSourceEnums >](#) & [TriggerSource](#)
 Description:
Specifies the internal signal or physical input line to use as the trigger source.
- [GenApi::EnumerationT< TriggerActivationEnums >](#) & [TriggerActivation](#)
 Description: *Specifies the activation mode of the trigger.*
- [GenApi::EnumerationT< SensorShutterModeEnums >](#) & [SensorShutterMode](#)
 Description: *Sets the shutter mode of the device.*
- [GenApi::Float](#) & [TriggerDelay](#)
 Description:
Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.
- [GenApi::EnumerationT< TriggerModeEnums >](#) & [TriggerMode](#)
 Description:
Controls whether or not trigger is active.
- [GenApi::Float](#) & [AcquisitionFrameRate](#)
 Description: *User controlled acquisition frame rate in Hertz Visibility:*
- [GenApi::EnumerationT< TriggerOverlapEnums >](#) & [TriggerOverlap](#)
 Description: *Specifies the overlap mode of the trigger.*
- [GenApi::EnumerationT< TriggerSelectorEnums >](#) & [TriggerSelector](#)

- Description: Selects the type of trigger to configure.*

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

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

- [GenApi::IEnumerationT< ExposureAutoEnums >](#) & [ExposureAuto](#)

Description: Sets the automatic exposure mode Visibility:

- [GenApi::Integer](#) & [AcquisitionBurstFrameCount](#)

Description:

This feature is used only if the FrameBurstStart trigger is enabled and the FrameBurstEnd trigger is disabled.

- [GenApi::Integer](#) & [EventTest](#)

Description: Returns the unique identifier of the Test type of [Event](#).

- [GenApi::Integer](#) & [EventTestTimestamp](#)

Description: Returns the Timestamp of the Test [Event](#).

- [GenApi::Integer](#) & [EventExposureEndFrameID](#)

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

- [GenApi::Integer](#) & [EventExposureEnd](#)

Description: Returns the unique identifier of the Exposure End type of [Event](#).

- [GenApi::Integer](#) & [EventExposureEndTimestamp](#)

Description: Returns the Timestamp of the Exposure End [Event](#).

- [GenApi::Integer](#) & [EventError](#)

Description: Returns the unique identifier of the Error type of [Event](#).

- [GenApi::Integer](#) & [EventErrorTimestamp](#)

Description: Returns the Timestamp of the Error [Event](#).

- [GenApi::Integer](#) & [EventErrorCode](#)

Description: Returns the error code for the error that happened Visibility:

- [GenApi::Integer](#) & [EventErrorFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Error [Event](#).

- [GenApi::IEnumerationT< EventSelectorEnums >](#) & [EventSelector](#)

Description: Selects which [Event](#) to enable or disable.

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

Description: Returns the status of the event serial receive overflow.

- [GenApi::Integer](#) & [EventSerialPortReceive](#)

Description: Returns the unique identifier of the Serial Port Receive type of [Event](#).

- [GenApi::Integer](#) & [EventSerialPortReceiveTimestamp](#)

Description: Returns the Timestamp of the Serial Port Receive [Event](#).

- [GenApi::IString](#) & [EventSerialData](#)

Description: Returns the serial data that was received.

- [GenApi::Integer](#) & [EventSerialDataLength](#)

Description: Returns the length of the received serial data that was included in the event payload.

- [GenApi::IEnumerationT< EventNotificationEnums >](#) & [EventNotification](#)

Description: Enables/Disables the selected event.

- [GenApi::Integer](#) & [LogicBlockLUTRowIndex](#)

Description: Controls the row of the truth table to access in the selected LUT.

- [GenApi::IEnumerationT< LogicBlockSelectorEnums >](#) & [LogicBlockSelector](#)

Description: Selects which LogicBlock to configure Visibility:

- [GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums >](#) & [LogicBlockLUTInputActivation](#)

Description: Selects the activation mode of the Logic Input Source signal.

- [GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums >](#) & [LogicBlockLUTInputSelector](#)

Description: Controls which LogicBlockLUT Input Source & Activation to access.

- [GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums >](#) & [LogicBlockLUTInputSource](#)

Description: Selects the source for the input into the Logic LUT.

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

- Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.*

 - [GenApi::Integer](#) & [LogicBlockLUTOutputValueAll](#)

Description: Sets the value of all the output bits in the selected LUT.

- [GenApi::EnumerationT](#) < [LogicBlockLUTSelectorEnums](#) > & [LogicBlockLUTSelector](#)

Description: Selects which LogicBlock LUT to configure Visibility:

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

Description:
Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

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

Description:
Enables/disables the color transform selected with ColorTransformationSelector.

- [GenApi::EnumerationT](#) < [ColorTransformationSelectorEnums](#) > & [ColorTransformationSelector](#)

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

- [GenApi::EnumerationT](#) < [RgbTransformLightSourceEnums](#) > & [RgbTransformLightSource](#)

Description:
Used to select from a set of RGBtoRGB transform matrices calibrated for different light sources.

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

Description: Controls the color saturation.

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

Description: Enables/disables Saturation adjustment.

- [GenApi::EnumerationT](#) < [ColorTransformationValueSelectorEnums](#) > & [ColorTransformationValueSelector](#)

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

- [GenApi::Integer](#) & [TimestampLatchValue](#)

Description: Returns the latched value of the timestamp counter.

- [GenApi::Command](#) & [TimestampReset](#)

Description: Resets the current value of the device timestamp counter.

- [GenApi::IString](#) & [DeviceUserID](#)

Description: User-programmable device identifier.

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

Description: Device temperature in degrees Celsius (C).

- [GenApi::Integer](#) & [MaxDeviceResetTime](#)

Description: Time to wait until device reset complete (ms).

- [GenApi::Integer](#) & [DeviceTLVersionMinor](#)

Description:
Minor version of the Transport Layer of the device.

- [GenApi::IString](#) & [DeviceSerialNumber](#)

Description:
Device's serial number.

- [GenApi::IString](#) & [DeviceVendorName](#)

Description: Name of the manufacturer of the device.

- [GenApi::EnumerationT](#) < [DeviceRegistersEndiannessEnums](#) > & [DeviceRegistersEndianness](#)

Description: Endianness of the registers of the device.

- [GenApi::IString](#) & [DeviceManufacturerInfo](#)

Description: Manufacturer information about the device.

- [GenApi::Integer](#) & [DeviceLinkSpeed](#)

Description:
Indicates the speed of transmission negotiated on the specified Link.

- [GenApi::Integer](#) & [LinkUptime](#)

Description: Time since the last phy negotiation (enumeration).

- [GenApi::Integer](#) & [DeviceEventChannelCount](#)

Description:
Indicates the number of event channels supported by the device.

- [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](#)
Description:
Character set used by the strings of the device's bootstrap registers.
- [GenApi::Integer](#) & [DeviceLinkThroughputLimit](#)
Description:
Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.
- [GenApi::IString](#) & [DeviceFirmwareVersion](#)
Description: Version of the firmware on the device.
- [GenApi::Integer](#) & [DeviceStreamChannelCount](#)
Description:
Indicates the number of streaming channels supported by the device.
- [GenApi::IEnumerationT< DeviceTLTypeEnums >](#) & [DeviceTLType](#)
Description: Transport Layer type of the device.
- [GenApi::IString](#) & [DeviceVersion](#)
Description: Version of the device.
- [GenApi::IEnumerationT< DevicePowerSupplySelectorEnums >](#) & [DevicePowerSupplySelector](#)
Description:
Selects the power supply source to control or read.
- [GenApi::IString](#) & [SensorDescription](#)
Description: Returns Sensor Description Visibility:
- [GenApi::IString](#) & [DeviceModelName](#)
Description: Model of the device.
- [GenApi::Integer](#) & [DeviceTLVersionMajor](#)
Description:
Major version of the Transport Layer of the device.
- [GenApi::IEnumerationT< DeviceTemperatureSelectorEnums >](#) & [DeviceTemperatureSelector](#)
Description:
Selects the location within the device, where the temperature will be measured.
- [GenApi::Integer](#) & [EnumerationCount](#)
Description: Number of enumerations since uptime.
- [GenApi::IFloat](#) & [PowerSupplyCurrent](#)
Description:
Indicates the output current of the selected power supply (A).
- [GenApi::IString](#) & [DeviceID](#)
Description: Device identifier (serial number).
- [GenApi::Integer](#) & [DeviceUptime](#)
Description: Total time since the device was powered up in seconds.
- [GenApi::Integer](#) & [DeviceLinkCurrentThroughput](#)
Description: Current bandwidth of streamed data.
- [GenApi::Integer](#) & [DeviceMaxThroughput](#)
Description:
Maximum bandwidth of the data that can be streamed out of the device.
- [GenApi::ICommand](#) & [FactoryReset](#)
Description: Returns all user tables to factory default Visibility:
- [GenApi::IFloat](#) & [PowerSupplyVoltage](#)
Description:
Indicates the current voltage of the selected power supply (V).

- [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](#)
*Description:
Percentage of streamed data bandwidth reserved for packet resend.*
- [GenApi::Integer & AasRoiOffsetY](#)
*Description:
Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::Integer & AasRoiOffsetX](#)
*Description:
Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IEnumerationT < AutoExposureControlPriorityEnums > & AutoExposureControlPriority](#)
*Description:
Selects whether to adjust gain or exposure first.*
- [GenApi::IFloat & BalanceWhiteAutoLowerLimit](#)
*Description:
Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.*
- [GenApi::IFloat & BalanceWhiteAutoDamping](#)
*Description:
Controls how quickly 'BalanceWhiteAuto' adjusts the values for Red and Blue BalanceRatio in response to changing conditions.*
- [GenApi::Integer & AasRoiHeight](#)
*Description:
Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.*
- [GenApi::IFloat & AutoExposureGreyValueUpperLimit](#)
*Description:
The highest value in percentage that the target mean may reach.*
- [GenApi::IFloat & AutoExposureTargetGreyValue](#)
*Description:
This is the user-specified target grey level (image mean) to apply to the current image.*
- [GenApi::IFloat & AutoExposureGainLowerLimit](#)
*Description:
The smallest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGreyValueLowerLimit](#)
*Description:
The lowest value in percentage that the target mean may reach.*
- [GenApi::IEnumerationT < AutoExposureMeteringModeEnums > & AutoExposureMeteringMode](#)
*Description:
Selects a metering mode: average, spot, or partial metering.*
- [GenApi::IFloat & AutoExposureExposureTimeUpperLimit](#)
*Description:
The largest exposure time that auto exposure can set.*
- [GenApi::IFloat & AutoExposureGainUpperLimit](#)
*Description:
The largest gain that auto exposure can set.*
- [GenApi::IFloat & AutoExposureControlLoopDamping](#)
*Description:
It controls how fast the exposure and gain get settled.*
- [GenApi::IFloat & AutoExposureEVCompensation](#)
*Description:
The EV compensation value used in the exposure compensation.*
- [GenApi::IFloat & AutoExposureExposureTimeLowerLimit](#)

Description:

The smallest exposure time that auto exposure can set.

- [GenApi::IEnumerationT](#) < [BalanceWhiteAutoProfileEnums](#) > & [BalanceWhiteAutoProfile](#)

Description: Selects the profile used by BalanceWhiteAuto.

- [GenApi::IEnumerationT](#) < [AutoAlgorithmSelectorEnums](#) > & [AutoAlgorithmSelector](#)

Description:

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

- [GenApi::IEnumerationT](#) < [AutoExposureTargetGreyValueAutoEnums](#) > & [AutoExposureTargetGreyValue↔Auto](#)

Description:

This indicates whether the target image grey level is automatically set by the camera or manually set by the user.

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

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the AutoAlgorithm↔Selector feature.

- [GenApi::IEnumerationT](#) < [AutoExposureLightingModeEnums](#) > & [AutoExposureLightingMode](#)

Description:

Selects a lighting mode: Backlight, Frontlight or Normal (default).

- [GenApi::Integer](#) & [AasRoiWidth](#)

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

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

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

- [GenApi::Integer](#) & [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::Integer](#) & [GevInterfaceSelector](#)

Description: Selects which logical link to control.

- [GenApi::Integer](#) & [GevSCPD](#)

Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.

- [GenApi::Integer](#) & [GevTimestampTickFrequency](#)

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

- [GenApi::Integer](#) & [GevSCPSPacketSize](#)

Description: Specifies the stream packet size (in bytes) to send on this channel.

- [GenApi::Integer](#) & [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::Integer](#) & [GevMCTT](#)

Description: Indicates the transmission timeout of the message channel.

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

Description: The state of this feature is copied into the "do not fragment" bit of the IP header of each stream packet.

- [GenApi::Integer](#) & [GevCurrentSubnetMask](#)

Description: Reports the subnet mask of the given logical link.

- [GenApi::Integer](#) & [GevStreamChannelSelector](#)

Description: Selects the stream channel to control.

- [GenApi::Integer](#) & [GevCurrentIPAddress](#)

Description: Reports the IP address for the given logical link.

- [GenApi::Integer](#) & [GevMCSP](#)

- Description: Indicates the source port of the message channel.*

 - [GenApi::Integer](#) & [GevGVCPPendingTimeout](#)

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.
- [GenApi::EnumerationT](#) < [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::Integer](#) & [GevMACAddress](#)

Description: MAC address of the logical link.
- [GenApi::Integer](#) & [GevPersistentSubnetMask](#)

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.
- [GenApi::Integer](#) & [GevMCPHostPort](#)

Description: The port to which the device must send messages Visibility:
- [GenApi::Integer](#) & [GevSCPHostPort](#)

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.
- [GenApi::IBoolean](#) & [GevGVCPPendingAck](#)

Description: Enables the generation of PENDING_ACK.
- [GenApi::Integer](#) & [GevSCPIInterfaceIndex](#)

Description: Index of the logical link to use.
- [GenApi::IBoolean](#) & [GevSupportedOption](#)

Description: Returns if the selected GEV option is supported.
- [GenApi::EnumerationT](#) < [GevIEEE1588ModeEnums](#) > & [GevIEEE1588Mode](#)

Description: Provides the mode of the IEEE 1588 clock.
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationLLA](#)

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.
- [GenApi::Integer](#) & [GevSCSP](#)

Description: Indicates the source port of the stream channel.
- [GenApi::IBoolean](#) & [GevIEEE1588](#)

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.
- [GenApi::IBoolean](#) & [GevSCCFGExtendedChunkData](#)

Description: Enables cameras to use the extended chunk data payload type for this stream channel.
- [GenApi::Integer](#) & [GevPersistentIPAddress](#)

Description: Controls the Persistent IP address for this logical link.
- [GenApi::IBoolean](#) & [GevCurrentIPConfigurationPersistentIP](#)

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.
- [GenApi::EnumerationT](#) < [GevIEEE1588ClockAccuracyEnums](#) > & [GevIEEE1588ClockAccuracy](#)

Description: Indicates the expected accuracy of the device clock when it is the grandmaster, or in the event it becomes the grandmaster.
- [GenApi::Integer](#) & [GevHeartbeatTimeout](#)

Description: Indicates the current heartbeat timeout in milliseconds.
- [GenApi::Integer](#) & [GevPersistentDefaultGateway](#)

Description: Controls the persistent default gateway for this logical link.
- [GenApi::EnumerationT](#) < [GevCCPEnums](#) > & [GevCCP](#)

Description: Controls the device access privilege of an application.
- [GenApi::Integer](#) & [GevMCDA](#)

Description: Controls the destination IP address of the message channel Visibility:
- [GenApi::Integer](#) & [GevSCDA](#)

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

- [GenApi::Integer](#) & [GevSCPDirection](#)
Description: Transmit or Receive of the channel Visibility:
- [GenApi::Boolean](#) & [GevSCPSFireTestPacket](#)
Description: Sends a test packet.
- [GenApi::String](#) & [GevSecondURL](#)
Description: The second choice of URL to the XML device description file.
- [GenApi::EnumerationT<GevSupportedOptionSelectorEnums>](#) & [GevSupportedOptionSelector](#)
Description: Selects the GEV option to interrogate for existing support.
- [GenApi::Boolean](#) & [GevGVCPHeartbeatDisable](#)
Description: Disables the GVCP heartbeat.
- [GenApi::Integer](#) & [GevMCRC](#)
Description: Indicates the number of retries of the message channel.
- [GenApi::Boolean](#) & [GevSCPSBigEndian](#)
Description: Endianess of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [GevNumberOfInterfaces](#)
Description: Indicates the number of physical network interfaces supported by this device.
- [GenApi::Integer](#) & [TLParamsLocked](#)
Description: Visibility:
- [GenApi::Integer](#) & [PayloadSize](#)
Description: Provides the number of bytes transferred for each image or chunk on the stream channel.
- [GenApi::Integer](#) & [PacketResendRequestCount](#)
Description: Counts the number of resend requests received from the host.
- [GenApi::Boolean](#) & [SharpeningEnable](#)
Description:
Enables/disables the sharpening feature.
- [GenApi::EnumerationT<BlackLevelSelectorEnums>](#) & [BlackLevelSelector](#)
Description:
Selects which black level to control.
- [GenApi::Boolean](#) & [GammaEnable](#)
Description: Enables/disables gamma correction.
- [GenApi::Boolean](#) & [SharpeningAuto](#)
Description:
Enables/disables the auto sharpening feature.
- [GenApi::Boolean](#) & [BlackLevelClampingEnable](#)
Description:
Enable the black level auto clamping feature which performing dark current compensation.
- [GenApi::Float](#) & [BalanceRatio](#)
Description:
Controls the balance ratio of the selected color relative to green.
- [GenApi::EnumerationT<BalanceWhiteAutoEnums>](#) & [BalanceWhiteAuto](#)
Description:
White Balance compensates for color shifts caused by different lighting conditions.
- [GenApi::Float](#) & [SharpeningThreshold](#)
Description:
Controls the minimum intensity gradient change to invoke sharpening.
- [GenApi::EnumerationT<GainAutoEnums>](#) & [GainAuto](#)
Description:
Sets the automatic gain mode.
- [GenApi::Float](#) & [Sharpening](#)
Description:
Controls the amount to sharpen a signal.
- [GenApi::Float](#) & [Gain](#)
Description:
Controls the amplification of the video signal in dB.

- [GenApi::IEnumerationT< BalanceRatioSelectorEnums > & BalanceRatioSelector](#)
 Description:
Selects a balance ratio to configure once a balance ratio control has been selected.
- [GenApi::IEnumerationT< GainSelectorEnums > & GainSelector](#)
 Description: *Selects which gain to control.*
- [GenApi::IFloat & BlackLevel](#)
 Description:
Controls the offset of the video signal in percent.
- [GenApi::Integer & BlackLevelRaw](#)
 Description:
Controls the offset of the video signal in camera specific units.
- [GenApi::IFloat & Gamma](#)
 Description: *Controls the gamma correction of pixel intensity.*
- [GenApi::Integer & DefectTableIndex](#)
 Description:
Controls the offset of the element to access in the defective pixel location table.
- [GenApi::ICommand & DefectTableFactoryRestore](#)
 Description: *Restores the Defective Pixel Table to its factory default state, which was calibrated during manufacturing.*
- [GenApi::Integer & DefectTableCoordinateY](#)
 Description:
Returns the Y coordinate of the defective pixel at DefectTableIndex within the defective pixel table.
- [GenApi::ICommand & DefectTableSave](#)
 Description: *Saves the current defective pixel table non-volatile memory, so that it is preserved when the camera boots up.*
- [GenApi::IEnumerationT< DefectCorrectionModeEnums > & DefectCorrectionMode](#)
 Description: *Controls the method used for replacing defective pixels.*
- [GenApi::Integer & DefectTableCoordinateX](#)
 Description:
Returns the X coordinate of the defective pixel at DefectTableIndex within the defective pixel table.
- [GenApi::Integer & DefectTablePixelCount](#)
 Description:
The number of defective pixel locations in the current table.
- [GenApi::IBoolean & DefectCorrectStaticEnable](#)
 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](#)
 Description:
Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.
- [GenApi::IEnumerationT< UserSetSelectorEnums > & UserSetSelector](#)
 Description:
Selects the feature User Set to load, save or configure.
- [GenApi::ICommand & UserSetLoad](#)
 Description:
Loads the User Set specified by UserSetSelector to the device and makes it active.
- [GenApi::IEnumerationT< UserSetDefaultEnums > & UserSetDefault](#)
 Description:
Selects the feature User Set to load and make active by default when the device is restarted.
- [GenApi::IEnumerationT< SerialPortBaudRateEnums > & SerialPortBaudRate](#)
 Description: *This feature controls the baud rate used by the selected serial port.*
- [GenApi::Integer & SerialPortDataBits](#)
 Description: *This feature controls the number of data bits used by the selected serial port.*

- [GenApi::IEnumerationT< SerialPortParityEnums > & SerialPortParity](#)
Description: This feature controls the parity used by the selected serial port.
- [GenApi::Integer & SerialTransmitQueueMaxCharacterCount](#)
Description: >Returns the maximum number of characters in the serial port transmit queue.
- [GenApi::Integer & SerialReceiveQueueCurrentCharacterCount](#)
Description: Returns the number of characters currently in the serial port receive queue.
- [GenApi::IEnumerationT< SerialPortSelectorEnums > & SerialPortSelector](#)
Description: Selects which serial port of the device to control.
- [GenApi::IEnumerationT< SerialPortStopBitsEnums > & SerialPortStopBits](#)
Description: This feature controls the number of stop bits used by the selected serial port.
- [GenApi::ICommand & SerialReceiveQueueClear](#)
Description: This is a command that clears the device serial port receive queue.
- [GenApi::Integer & SerialReceiveFramingErrorCount](#)
Description: Returns the number of framing errors that have occurred on the serial port.
- [GenApi::Integer & SerialTransmitQueueCurrentCharacterCount](#)
Description: Returns the number of characters currently in the serial port transmit queue.
- [GenApi::Integer & SerialReceiveParityErrorCount](#)
Description: Returns the number of parity errors that have occurred on the serial port.
- [GenApi::IEnumerationT< SerialPortSourceEnums > & SerialPortSource](#)
Description: Specifies the physical input Line on which to receive serial data.
- [GenApi::Integer & SerialReceiveQueueMaxCharacterCount](#)
Description: >Returns the maximum number of characters in the serial port receive queue.
- [GenApi::Integer & SequencerSetStart](#)
Description: Sets the first sequencer set to be used.
- [GenApi::IEnumerationT< SequencerModeEnums > & SequencerMode](#)
Description: Controls whether or not a sequencer is active.
- [GenApi::IEnumerationT< SequencerConfigurationValidEnums > & SequencerConfigurationValid](#)
Description:
Display whether the current sequencer configuration is valid to run.
- [GenApi::IEnumerationT< SequencerSetValidEnums > & SequencerSetValid](#)
Description:
Displays whether the currently selected sequencer set's register contents are valid to use.
- [GenApi::Integer & SequencerSetSelector](#)
Description:
Selects the sequencer set to which subsequent settings apply.
- [GenApi::IEnumerationT< SequencerTriggerActivationEnums > & SequencerTriggerActivation](#)
Description:
Specifies the activation mode of the sequencer trigger.
- [GenApi::IEnumerationT< SequencerConfigurationModeEnums > & SequencerConfigurationMode](#)
Description:
Controls whether or not a sequencer is in configuration mode.
- [GenApi::ICommand & SequencerSetSave](#)
Description:
Saves the current device configuration to the currently selected sequencer set.
- [GenApi::IEnumerationT< SequencerTriggerSourceEnums > & SequencerTriggerSource](#)
Description:
Specifies the internal signal or physical input line to use as the sequencer trigger source.
- [GenApi::Integer & SequencerSetActive](#)
Description: Displays the currently active sequencer set.
- [GenApi::Integer & SequencerSetNext](#)
Description: Specifies the next sequencer set.
- [GenApi::ICommand & SequencerSetLoad](#)

Description:

Loads currently selected sequencer to the current device configuration.

- [GenApi::Integer](#) & [SequencerPathSelector](#)

Description:

Selects branching path to be used for subsequent settings.

- [GenApi::Boolean](#) & [SequencerFeatureEnable](#)

Description:

Enables the selected feature and makes it active in all sequencer sets.

- [GenApi::Integer](#) & [TransferBlockCount](#)

Description: Specifies the number of data blocks (images) that the device should stream before stopping.

- [GenApi::Command](#) & [TransferStart](#)

Description: Starts the streaming of data blocks (images) out of the device.

- [GenApi::Integer](#) & [TransferQueueMaxBlockCount](#)

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

- [GenApi::Integer](#) & [TransferQueueCurrentBlockCount](#)

Description: Returns number of data blocks (images) currently in the transfer queue.

- [GenApi::EnumerationT< TransferQueueModeEnums >](#) & [TransferQueueMode](#)

Description: Specifies the operation mode of the transfer queue.

- [GenApi::EnumerationT< TransferOperationModeEnums >](#) & [TransferOperationMode](#)

Description: Selects the operation mode of the transfer.

- [GenApi::Command](#) & [TransferStop](#)

Description: Stops the streaming of data block (images).

- [GenApi::Integer](#) & [TransferQueueOverflowCount](#)

Description: Returns number of images that have been lost before being transmitted because the transmit queue hasn't been cleared fast enough.

- [GenApi::EnumerationT< TransferControlModeEnums >](#) & [TransferControlMode](#)

Description: Selects the control method for the transfers.

- [GenApi::Float](#) & [ChunkBlackLevel](#)

Description: Returns the black level used to capture the image.

- [GenApi::Integer](#) & [ChunkFrameID](#)

Description: Returns the image frame ID.

- [GenApi::String](#) & [ChunkSerialData](#)

Description: Returns the serial data that was received.

- [GenApi::Float](#) & [ChunkExposureTime](#)

Description: Returns the exposure time used to capture the image.

- [GenApi::Boolean](#) & [ChunkSerialReceiveOverflow](#)

Description: Returns the status of the chunk serial receive overflow.

- [GenApi::Integer](#) & [ChunkTimestamp](#)

Description: Returns the Timestamp of the image.

- [GenApi::Boolean](#) & [ChunkModeActive](#)

Description: Activates the inclusion of Chunk data in the payload of the image.

- [GenApi::Integer](#) & [ChunkExposureEndLineStatusAll](#)

Description: Returns the status of all the I/O lines at the end of exposure event.

- [GenApi::EnumerationT< ChunkGainSelectorEnums >](#) & [ChunkGainSelector](#)

Description: Selects which gain to retrieve Visibility:

- [GenApi::EnumerationT< ChunkSelectorEnums >](#) & [ChunkSelector](#)

Description: Selects which chunk data to enable or disable.

- [GenApi::EnumerationT< ChunkBlackLevelSelectorEnums >](#) & [ChunkBlackLevelSelector](#)

Description: Selects which black level to retrieve Visibility:

- [GenApi::Integer](#) & [ChunkWidth](#)

Description: Returns the width of the image included in the payload.

- [GenApi::Integer](#) & [ChunkImage](#)

- Description: Returns the image payload.*

 - [GenApi::Integer](#) & [ChunkHeight](#)

Description: Returns the height of the image included in the payload.
- [GenApi::EnumerationT< ChunkPixelFormatEnums >](#) & [ChunkPixelFormat](#)

Description: Format of the pixel provided by the camera Visibility:
- [GenApi::IFloat](#) & [ChunkGain](#)

Description: Returns the gain used to capture the image.
- [GenApi::Integer](#) & [ChunkSequencerSetActive](#)

Description: Returns the index of the active set of the running sequencer included in the payload.
- [GenApi::Integer](#) & [ChunkCRC](#)

Description: Returns the CRC of the image payload.
- [GenApi::Integer](#) & [ChunkOffsetX](#)

Description: Returns the Offset X of the image included in the payload.
- [GenApi::Integer](#) & [ChunkOffsetY](#)

Description: Returns the Offset Y of the image included in the payload.
- [GenApi::Boolean](#) & [ChunkEnable](#)

Description: Enables the inclusion of the selected Chunk data in the payload of the image.
- [GenApi::Integer](#) & [ChunkSerialDataLength](#)

Description: Returns the length of the received serial data that was included in the payload.
- [GenApi::Integer](#) & [FileAccessOffset](#)

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.
- [GenApi::Integer](#) & [FileAccessLength](#)

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.
- [GenApi::EnumerationT< FileOperationStatusEnums >](#) & [FileOperationStatus](#)

Description: Represents the file operation execution status.
- [GenApi::ICommand](#) & [FileOperationExecute](#)

Description:
This is a command that executes the selected file operation on the selected file.
- [GenApi::EnumerationT< FileOpenModeEnums >](#) & [FileOpenMode](#)

Description:
The mode of the file when it is opened.
- [GenApi::Integer](#) & [FileOperationResult](#)

Description: Represents the file operation result.
- [GenApi::EnumerationT< FileOperationSelectorEnums >](#) & [FileOperationSelector](#)

Description:
Sets operation to execute on the selected file when the execute command is given.
- [GenApi::EnumerationT< FileSelectorEnums >](#) & [FileSelector](#)

Description:
Selects which file is being operated on.
- [GenApi::Integer](#) & [FileSize](#)

Description: Represents the size of the selected file in bytes.
- [GenApi::EnumerationT< BinningSelectorEnums >](#) & [BinningSelector](#)

Description:
Selects which binning engine is controlled by the BinningHorizontal and BinningVertical features.
- [GenApi::Integer](#) & [PixelDynamicRangeMin](#)

Description: Minimum value that can be returned during the digitization process.
- [GenApi::Integer](#) & [PixelDynamicRangeMax](#)

Description: Maximum value that can be returned during the digitization process.
- [GenApi::Integer](#) & [OffsetY](#)

Description:
Vertical offset from the origin to the ROI (in pixels).
- [GenApi::Integer](#) & [BinningHorizontal](#)

- Description:
 - Number of horizontal photo-sensitive cells to combine together.
- [GenApi::Integer & Width](#)
 - Description:
 - Width of the image provided by the device (in pixels).
- [GenApi::EnumerationT< TestPatternGeneratorSelectorEnums > & TestPatternGeneratorSelector](#)
 - Description:
 - Selects which test pattern generator is controlled by the TestPattern feature.
- [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::EnumerationT< TestPatternEnums > & TestPattern](#)
 - Description:
 - Selects the type of test pattern that is generated by the device as image source.
- [GenApi::EnumerationT< PixelColorFilterEnums > & PixelColorFilter](#)
 - Description:
 - Type of color filter that is applied to the image.
- [GenApi::Integer & WidthMax](#)
 - Description:
 - Maximum width of the image (in pixels).
- [GenApi::EnumerationT< AdcBitDepthEnums > & AdcBitDepth](#)
 - Description:
 - Selects which ADC bit depth to use.
- [GenApi::Integer & BinningVertical](#)
 - Description:
 - Number of vertical photo-sensitive cells to combine together.
- [GenApi::EnumerationT< DecimationHorizontalModeEnums > & DecimationHorizontalMode](#)
 - Description:
 - The mode used to reduce the horizontal resolution when DecimationHorizontal is used.
- [GenApi::EnumerationT< BinningVerticalModeEnums > & BinningVerticalMode](#)
 - Description:
 - Visibility:
- [GenApi::Integer & OffsetX](#)
 - Description:
 - Horizontal offset from the origin to the ROI (in pixels).
- [GenApi::Integer & HeightMax](#)
 - Description:
 - Maximum height of the image (in pixels).
- [GenApi::Integer & DecimationHorizontal](#)
 - Description:
 - Horizontal decimation of the image.
- [GenApi::EnumerationT< PixelSizeEnums > & PixelSize](#)
 - Description:
 - Total size in bits of a pixel of the image.
- [GenApi::Integer & SensorHeight](#)
 - Description:
 - Effective height of the sensor in pixels.
- [GenApi::EnumerationT< DecimationSelectorEnums > & DecimationSelector](#)
 - Description:
 - Selects which decimation layer is controlled by the DecimationHorizontal and DecimationVertical features.
- [GenApi::IBoolean & IspEnable](#)
 - Description:
 - Controls whether the image processing core is used for optional pixel format mode (i.e.
- [GenApi::IBoolean & AdaptiveCompressionEnable](#)
 - Description:
 - Controls whether lossless compression adapts to the image content.
- [GenApi::EnumerationT< ImageCompressionModeEnums > & ImageCompressionMode](#)

- Description: Visibility:*

 - [GenApi::Integer](#) & [DecimationVertical](#)

Description:
Vertical decimation of the image.
 - [GenApi::Integer](#) & [Height](#)

Description:
Height of the image provided by the device (in pixels).
 - [GenApi::EnumerationT](#)< [BinningHorizontalModeEnums](#) > & [BinningHorizontalMode](#)

Description: Visibility:
 - [GenApi::EnumerationT](#)< [PixelFormatEnums](#) > & [PixelFormat](#)

Description: Format of the pixel provided by the camera.
 - [GenApi::Integer](#) & [SensorWidth](#)

Description: Effective width of the sensor in pixels.
 - [GenApi::EnumerationT](#)< [DecimationVerticalModeEnums](#) > & [DecimationVerticalMode](#)

Description:
The mode used to reduce the vertical resolution when DecimationVertical is used.
 - [GenApi::ICommand](#) & [TestEventGenerate](#)

Description: This command generates a test event and sends it to the host.
 - [GenApi::ICommand](#) & [TriggerEventTest](#)

Description: This command generates a test event and sends it to the host.
 - [GenApi::Integer](#) & [GuiXmlManifestAddress](#)

Description: Location of the GUI XML manifest table.
 - [GenApi::Integer](#) & [Test0001](#)

Description: For testing only.
 - [GenApi::Boolean](#) & [V3_3Enable](#)

Description: Internally generated 3.3V rail.
 - [GenApi::EnumerationT](#)< [LineModeEnums](#) > & [LineMode](#)

Description: Controls if the physical Line is used to Input or Output a signal.
 - [GenApi::EnumerationT](#)< [LineSourceEnums](#) > & [LineSource](#)

Description: Selects which internal acquisition or I/O source signal to output on the selected line.
 - [GenApi::EnumerationT](#)< [LineInputFilterSelectorEnums](#) > & [LineInputFilterSelector](#)

Description: Selects the kind of input filter to configure: Deglitch or Debounce.
 - [GenApi::Boolean](#) & [UserOutputValue](#)

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).
 - [GenApi::Integer](#) & [UserOutputValueAll](#)

Description: Returns the current status of all the user output status bits in a hexadecimal representation (UserOutput 0 status corresponds to bit 0, UserOutput 1 status with bit 1, etc).
 - [GenApi::EnumerationT](#)< [UserOutputSelectorEnums](#) > & [UserOutputSelector](#)

Description: Selects which bit of the User Output register is set by UserOutputValue.
 - [GenApi::Boolean](#) & [LineStatus](#)

Description: Returns the current status of the selected input or output Line Visibility:
 - [GenApi::EnumerationT](#)< [LineFormatEnums](#) > & [LineFormat](#)

Description: Displays the current electrical format of the selected physical input or output Line.
 - [GenApi::Integer](#) & [LineStatusAll](#)

Description: Returns the current status of all the line status bits in a hexadecimal representation (Line 0 status corresponds to bit 0, Line 1 status with bit 1, etc).
 - [GenApi::EnumerationT](#)< [LineSelectorEnums](#) > & [LineSelector](#)

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:
 - [GenApi::EnumerationT](#)< [ExposureActiveModeEnums](#) > & [ExposureActiveMode](#)

Description: Control sensor active exposure mode.
 - [GenApi::Boolean](#) & [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::Integer](#) & [CounterValue](#)
Description: Current counter value Visibility:
- [GenApi::IEnumerationT< CounterSelectorEnums >](#) & [CounterSelector](#)
Description: Selects which counter to configure Visibility:
- [GenApi::Integer](#) & [CounterValueAtReset](#)
Description: Value of the selected Counter when it was reset by a trigger.
- [GenApi::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::Integer](#) & [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::Integer](#) & [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::Integer](#) & [DeviceSFNCVersionMajor](#)
Description: Major version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionMinor](#)
Description: Minor version of the Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceSFNCVersionSubMinor](#)
Description: Sub minor version of Standard Features Naming Convention that was used to create the device's [GenICam XML](#).
- [GenApi::Integer](#) & [DeviceManifestEntrySelector](#)
Description: Selects the manifest entry to reference.
- [GenApi::Integer](#) & [DeviceManifestXMLMajorVersion](#)
Description: Indicates the major version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLMinorVersion](#)
Description: Indicates the minor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestXMLSubMinorVersion](#)
Description: Indicates the subminor version number of the [GenICam XML](#) file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMajorVersion](#)
Description: Indicates the major version number of the schema file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceManifestSchemaMinorVersion](#)
Description: Indicates the minor version number of the schema file of the selected manifest entry.

- [GenApi::IString](#) & [DeviceManifestPrimaryURL](#)
Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::IString](#) & [DeviceManifestSecondaryURL](#)
Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.
- [GenApi::Integer](#) & [DeviceTLVersionSubMinor](#)
Description: Sub minor version of the Transport Layer of the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMajor](#)
Description: Major version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceGenCPVersionMinor](#)
Description: Minor version of the GenCP protocol supported by the device.
- [GenApi::Integer](#) & [DeviceConnectionSelector](#)
Description: Selects which Connection of the device to control.
- [GenApi::Integer](#) & [DeviceConnectionSpeed](#)
Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.
- [GenApi::EnumerationT< DeviceConnectionStatusEnums >](#) & [DeviceConnectionStatus](#)
Description: Indicates the status of the specified Connection.
- [GenApi::Integer](#) & [DeviceLinkSelector](#)
Description: Selects which Link of the device to control.
- [GenApi::EnumerationT< DeviceLinkThroughputLimitModeEnums >](#) & [DeviceLinkThroughputLimitMode](#)
Description: Controls if the DeviceLinkThroughputLimit is active.
- [GenApi::Integer](#) & [DeviceLinkConnectionCount](#)
Description: Returns the number of physical connection of the device used by a particular Link.
- [GenApi::EnumerationT< DeviceLinkHeartbeatModeEnums >](#) & [DeviceLinkHeartbeatMode](#)
Description: Activate or deactivate the Link's heartbeat.
- [GenApi::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::Integer](#) & [DeviceStreamChannelSelector](#)
Description: Selects the stream channel to control.
- [GenApi::EnumerationT< DeviceStreamChannelTypeEnums >](#) & [DeviceStreamChannelType](#)
Description: Reports the type of the stream channel.
- [GenApi::Integer](#) & [DeviceStreamChannelLink](#)
Description: Index of device's Link to use for streaming the specified stream channel.
- [GenApi::EnumerationT< DeviceStreamChannelEndiannessEnums >](#) & [DeviceStreamChannelEndianness](#)
Description: Endianness of multi-byte pixel data for this stream.
- [GenApi::Integer](#) & [DeviceStreamChannelPacketSize](#)
Description: Specifies the stream packet size, in bytes, to send on the selected channel for a Transmitter or specifies the maximum packet size supported by a receiver.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceStart](#)
Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.
- [GenApi::ICommand](#) & [DeviceFeaturePersistenceEnd](#)
Description: Indicate to the device the end of feature persistence.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingStart](#)
Description: Prepare the device for registers streaming without checking for consistency.
- [GenApi::ICommand](#) & [DeviceRegistersStreamingEnd](#)
Description: Announce the end of registers streaming.
- [GenApi::ICommand](#) & [DeviceRegistersCheck](#)
Description: Perform the validation of the current register set for consistency.
- [GenApi::Boolean](#) & [DeviceRegistersValid](#)
Description: Returns if the current register set is valid and consistent.

- [GenApi::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::Integer & Timestamp](#)
Description: Reports the current value of the device timestamp counter.
- [GenApi::IEnumerationT< SensorTapsEnums > & SensorTaps](#)
Description: Number of taps of the camera sensor.
- [GenApi::IEnumerationT< SensorDigitizationTapsEnums > & SensorDigitizationTaps](#)
Description: Number of digitized samples outputted simultaneously by the camera A/D conversion stage.
- [GenApi::IEnumerationT< RegionSelectorEnums > & RegionSelector](#)
Description: Selects the Region of interest to control.
- [GenApi::IEnumerationT< RegionModeEnums > & RegionMode](#)
Description: Controls if the selected Region of interest is active and streaming.
- [GenApi::IEnumerationT< RegionDestinationEnums > & RegionDestination](#)
Description: Control the destination of the selected region.
- [GenApi::IEnumerationT< ImageComponentSelectorEnums > & ImageComponentSelector](#)
Description: Selects a component to activate data streaming from.
- [GenApi::IBoolean & ImageComponentEnable](#)
Description: Controls if the selected component streaming is active.
- [GenApi::Integer & LinePitch](#)
Description: Total number of bytes between 2 successive lines.
- [GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > & PixelFormatInfoSelector](#)
Description: Select the pixel format for which the information will be returned.
- [GenApi::Integer & PixelFormatInfoID](#)
Description: Returns the value used by the streaming channels to identify the selected pixel format.
- [GenApi::IEnumerationT< DeinterlacingEnums > & Deinterlacing](#)
Description: Controls how the device performs de-interlacing.
- [GenApi::IEnumerationT< ImageCompressionRateOptionEnums > & ImageCompressionRateOption](#)
Description: Two rate controlling options are offered: fixed bit rate or fixed quality.
- [GenApi::Integer & ImageCompressionQuality](#)
Description: Control the quality of the produced compressed stream.
- [GenApi::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::Integer & TriggerDivider](#)

- Description: Specifies a division factor for the incoming trigger pulses.*

 - [GenApi::Integer](#) & [TriggerMultiplier](#)
- Description: Specifies a multiplication factor for the incoming trigger pulses.*

 - [GenApi::EnumerationT< ExposureTimeModeEnums >](#) & [ExposureTimeMode](#)
- Description: Sets the configuration mode of the ExposureTime feature.*

 - [GenApi::EnumerationT< ExposureTimeSelectorEnums >](#) & [ExposureTimeSelector](#)
- Description: Selects which exposure time is controlled by the ExposureTime feature.*

 - [GenApi::EnumerationT< GainAutoBalanceEnums >](#) & [GainAutoBalance](#)
- Description: Sets the mode for automatic gain balancing between the sensor color channels or taps.*

 - [GenApi::EnumerationT< BlackLevelAutoEnums >](#) & [BlackLevelAuto](#)
- Description: Controls the mode for automatic black level adjustment.*

 - [GenApi::EnumerationT< BlackLevelAutoBalanceEnums >](#) & [BlackLevelAutoBalance](#)
- Description: Controls the mode for automatic black level balancing between the sensor color channels or taps.*

 - [GenApi::EnumerationT< WhiteClipSelectorEnums >](#) & [WhiteClipSelector](#)
- Description: Selects which White Clip to control.*

 - [GenApi::IFloat](#) & [WhiteClip](#)
- Description: Controls the maximal intensity taken by the video signal before being clipped as an absolute physical value.*

 - [GenApi::IRegister](#) & [LUTValueAll](#)
- Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.*

 - [GenApi::Integer](#) & [UserOutputValueAllMask](#)
- Description: Sets the write mask to apply to the value specified by UserOutputValueAll before writing it in the User Output register.*

 - [GenApi::ICommand](#) & [CounterReset](#)
- Description: Does a software reset of the selected Counter and starts it.*

 - [GenApi::EnumerationT< 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::EnumerationT< TimerStatusEnums >](#) & [TimerStatus](#)
- Description: Returns the current status of the Timer.*

 - [GenApi::EnumerationT< TimerTriggerSourceEnums >](#) & [TimerTriggerSource](#)
- Description: Selects the source of the trigger to start the Timer.*

 - [GenApi::EnumerationT< TimerTriggerActivationEnums >](#) & [TimerTriggerActivation](#)
- Description: Selects the activation mode of the trigger to start the Timer.*

 - [GenApi::EnumerationT< EncoderSelectorEnums >](#) & [EncoderSelector](#)
- Description: Selects which Encoder to configure.*

 - [GenApi::EnumerationT< EncoderSourceAEnums >](#) & [EncoderSourceA](#)
- Description: Selects the signal which will be the source of the A input of the Encoder.*

 - [GenApi::EnumerationT< EncoderSourceBEnums >](#) & [EncoderSourceB](#)
- Description: Selects the signal which will be the source of the B input of the Encoder.*

 - [GenApi::EnumerationT< EncoderModeEnums >](#) & [EncoderMode](#)
- Description: Selects if the count of encoder uses FourPhase mode with jitter filtering or the HighResolution mode without jitter filtering.*

- [GenApi::Integer](#) & [EncoderDivider](#)
Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.
- [GenApi::EnumerationT< EncoderOutputModeEnums >](#) & [EncoderOutputMode](#)
Description: Selects the conditions for the Encoder interface to generate a valid Encoder output signal.
- [GenApi::EnumerationT< 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::EnumerationT< EncoderResetSourceEnums >](#) & [EncoderResetSource](#)
Description: Selects the signals that will be the source to reset the Encoder.
- [GenApi::EnumerationT< EncoderResetActivationEnums >](#) & [EncoderResetActivation](#)
Description: Selects the Activation mode of the Encoder Reset Source signal.
- [GenApi::ICommand](#) & [EncoderReset](#)
Description: Does a software reset of the selected Encoder and starts it.
- [GenApi::Integer](#) & [EncoderValue](#)
Description: Reads or writes the current value of the position counter of the selected Encoder.
- [GenApi::Integer](#) & [EncoderValueAtReset](#)
Description: Reads the value of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.
- [GenApi::EnumerationT< 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::EnumerationT< ActionUnconditionalModeEnums >](#) & [ActionUnconditionalMode](#)
Description: Enables the unconditional action command mode where action commands are processed even when the primary control channel is closed.
- [GenApi::Integer](#) & [ActionDeviceKey](#)
Description: Provides the device key that allows the device to check the validity of action commands.
- [GenApi::Integer](#) & [ActionQueueSize](#)
Description: Indicates the size of the scheduled action commands queue.
- [GenApi::Integer](#) & [ActionSelector](#)
Description: Selects to which Action Signal further Action settings apply.
- [GenApi::Integer](#) & [ActionGroupMask](#)
Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer](#) & [ActionGroupKey](#)
Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer](#) & [EventAcquisitionTrigger](#)
Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTriggerTimestamp](#)
Description: Returns the Timestamp of the Acquisition Trigger [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTriggerFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionStart](#)
Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionStartTimestamp](#)
Description: Returns the Timestamp of the Acquisition Start [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionEnd](#)

- Description: Returns the unique Identifier of the Acquisition End type of [Event](#).*

 - [GenApi::Integer](#) & [EventAcquisitionEndTimestamp](#)

Description: Returns the Timestamp of the Acquisition End [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferStart](#)

Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferStartTimestamp](#)

Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferEnd](#)

Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferEndTimestamp](#)

Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionTransferEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionError](#)

Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionErrorTimestamp](#)

Description: Returns the Timestamp of the Acquisition Error [Event](#).
- [GenApi::Integer](#) & [EventAcquisitionErrorFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error [Event](#).
- [GenApi::Integer](#) & [EventFrameTrigger](#)

Description: Returns the unique Identifier of the FrameTrigger type of [Event](#).
- [GenApi::Integer](#) & [EventFrameTriggerTimestamp](#)

Description: Returns the Timestamp of the FrameTrigger [Event](#).
- [GenApi::Integer](#) & [EventFrameTriggerFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).
- [GenApi::Integer](#) & [EventFrameStart](#)

Description: Returns the unique Identifier of the Frame Start type of [Event](#).
- [GenApi::Integer](#) & [EventFrameStartTimestamp](#)

Description: Returns the Timestamp of the Frame Start [Event](#).
- [GenApi::Integer](#) & [EventFrameStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).
- [GenApi::Integer](#) & [EventFrameEnd](#)

Description: Returns the unique Identifier of the Frame End type of [Event](#).
- [GenApi::Integer](#) & [EventFrameEndTimestamp](#)

Description: Returns the Timestamp of the Frame End [Event](#).
- [GenApi::Integer](#) & [EventFrameEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).
- [GenApi::Integer](#) & [EventFrameBurstStart](#)

Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).
- [GenApi::Integer](#) & [EventFrameBurstStartTimestamp](#)

Description: Returns the Timestamp of the Frame Burst Start [Event](#).
- [GenApi::Integer](#) & [EventFrameBurstStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst Start [Event](#).
- [GenApi::Integer](#) & [EventFrameBurstEnd](#)

Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).
- [GenApi::Integer](#) & [EventFrameBurstEndTimestamp](#)

Description: Returns the Timestamp of the Frame Burst End [Event](#).

- [GenApi::Integer](#) & [EventFrameBurstEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Burst End [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStart](#)
Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStartTimestamp](#)
Description: Returns the Timestamp of the Frame Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEnd](#)
Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEndTimestamp](#)
Description: Returns the Timestamp of the Frame Transfer End [Event](#).
- [GenApi::Integer](#) & [EventFrameTransferEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).
- [GenApi::Integer](#) & [EventExposureStart](#)
Description: Returns the unique Identifier of the Exposure Start type of [Event](#).
- [GenApi::Integer](#) & [EventExposureStartTimestamp](#)
Description: Returns the Timestamp of the Exposure Start [Event](#).
- [GenApi::Integer](#) & [EventExposureStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferStartFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEnd](#)
Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEndTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferEndFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPause](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPauseTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferPauseFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResume](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResumeTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferResumeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStart](#)
Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStartTimestamp](#)
Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).
- [GenApi::Integer](#) & [EventStream0TransferBlockStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Start Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockEnd](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockEndTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Block End Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block End Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockTrigger](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger Event.

- [GenApi::Integer](#) & [EventStream0TransferBlockTriggerFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstStart](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstStartTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstStartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstEnd](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstEndTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Burst End Event.

- [GenApi::Integer](#) & [EventStream0TransferBurstEndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End Event.

- [GenApi::Integer](#) & [EventStream0TransferOverflow](#)

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of Event.

- [GenApi::Integer](#) & [EventStream0TransferOverflowTimestamp](#)

Description: Returns the Timestamp of the Stream 0 Transfer Overflow Event.

- [GenApi::Integer](#) & [EventStream0TransferOverflowFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow Event.

- [GenApi::Integer](#) & [EventSequencerSetChange](#)

Description: Returns the unique Identifier of the Sequencer Set Change type of Event.

- [GenApi::Integer](#) & [EventSequencerSetChangeTimestamp](#)

Description: Returns the Timestamp of the Sequencer Set Change Event.

- [GenApi::Integer](#) & [EventSequencerSetChangeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change Event.

- [GenApi::Integer](#) & [EventCounter0Start](#)

Description: Returns the unique Identifier of the Counter 0 Start type of Event.

- [GenApi::Integer](#) & [EventCounter0StartTimestamp](#)

Description: Returns the Timestamp of the Counter 0 Start Event.

- [GenApi::Integer](#) & [EventCounter0StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start Event.

- [GenApi::Integer](#) & [EventCounter1Start](#)

Description: Returns the unique Identifier of the Counter 1 Start type of Event.

- [GenApi::Integer](#) & [EventCounter1StartTimestamp](#)

- Description: Returns the Timestamp of the Counter 1 Start [Event](#).*

 - [GenApi::Integer](#) & [EventCounter1StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).

- [GenApi::Integer](#) & [EventCounter0End](#)

Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).

- [GenApi::Integer](#) & [EventCounter0EndTimestamp](#)

Description: Returns the Timestamp of the Counter 0 End [Event](#).

- [GenApi::Integer](#) & [EventCounter0EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).

- [GenApi::Integer](#) & [EventCounter1End](#)

Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).

- [GenApi::Integer](#) & [EventCounter1EndTimestamp](#)

Description: Returns the Timestamp of the Counter 1 End [Event](#).

- [GenApi::Integer](#) & [EventCounter1EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).

- [GenApi::Integer](#) & [EventTimer0Start](#)

Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).

- [GenApi::Integer](#) & [EventTimer0StartTimestamp](#)

Description: Returns the Timestamp of the Timer 0 Start [Event](#).

- [GenApi::Integer](#) & [EventTimer0StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).

- [GenApi::Integer](#) & [EventTimer1Start](#)

Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).

- [GenApi::Integer](#) & [EventTimer1StartTimestamp](#)

Description: Returns the Timestamp of the Timer 1 Start [Event](#).

- [GenApi::Integer](#) & [EventTimer1StartFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).

- [GenApi::Integer](#) & [EventTimer0End](#)

Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).

- [GenApi::Integer](#) & [EventTimer0EndTimestamp](#)

Description: Returns the Timestamp of the Timer 0 End [Event](#).

- [GenApi::Integer](#) & [EventTimer0EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).

- [GenApi::Integer](#) & [EventTimer1End](#)

Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).

- [GenApi::Integer](#) & [EventTimer1EndTimestamp](#)

Description: Returns the Timestamp of the Timer 1 End [Event](#).

- [GenApi::Integer](#) & [EventTimer1EndFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End [Event](#).

- [GenApi::Integer](#) & [EventEncoder0Stopped](#)

Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).

- [GenApi::Integer](#) & [EventEncoder0StoppedTimestamp](#)

Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).

- [GenApi::Integer](#) & [EventEncoder0StoppedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).

- [GenApi::Integer](#) & [EventEncoder1Stopped](#)

Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).

- [GenApi::Integer](#) & [EventEncoder1StoppedTimestamp](#)

Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).

- [GenApi::Integer](#) & [EventEncoder1StoppedFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).

- [GenApi::Integer](#) & [EventEncoder0Restarted](#)
Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).
- [GenApi::Integer](#) & [EventEncoder0RestartedTimestamp](#)
Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder0RestartedFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder1Restarted](#)
Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).
- [GenApi::Integer](#) & [EventEncoder1RestartedTimestamp](#)
Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).
- [GenApi::Integer](#) & [EventEncoder1RestartedFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdge](#)
Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0RisingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdge](#)
Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1RisingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdge](#)
Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0FallingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdge](#)
Description: Returns the unique Identifier of the Line 1 Falling Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1FallingEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdge](#)
Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine0AnyEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdge](#)
Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdgeTimestamp](#)
Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLine1AnyEdgeFrameID](#)
Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger0](#)
Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger0Timestamp](#)

- Description: Returns the Timestamp of the Link Trigger 0 [Event](#).*

 - [GenApi::Integer](#) & [EventLinkTrigger0FrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger1](#)

Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger1Timestamp](#)

Description: Returns the Timestamp of the Link Trigger 1 [Event](#).
- [GenApi::Integer](#) & [EventLinkTrigger1FrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).
- [GenApi::Integer](#) & [EventActionLate](#)

Description: Returns the unique Identifier of the Action Late type of [Event](#).
- [GenApi::Integer](#) & [EventActionLateTimestamp](#)

Description: Returns the Timestamp of the Action Late [Event](#).
- [GenApi::Integer](#) & [EventActionLateFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).
- [GenApi::Integer](#) & [EventLinkSpeedChange](#)

Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).
- [GenApi::Integer](#) & [EventLinkSpeedChangeTimestamp](#)

Description: Returns the Timestamp of the Link Speed Change [Event](#).
- [GenApi::Integer](#) & [EventLinkSpeedChangeFrameID](#)

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).
- [GenApi::IRegister](#) & [FileAccessBuffer](#)

Description: Defines the intermediate access buffer that allows the exchange of data between the device file storage and the application.
- [GenApi::Integer](#) & [SourceCount](#)

Description: Controls or returns the number of sources supported by the device.
- [GenApi::IEnumerationT](#) < [SourceSelectorEnums](#) > & [SourceSelector](#)

Description: Selects the source to control.
- [GenApi::IEnumerationT](#) < [TransferSelectorEnums](#) > & [TransferSelector](#)

Description: Selects which stream transfers are currently controlled by the selected Transfer features.
- [GenApi::Integer](#) & [TransferBurstCount](#)

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.
- [GenApi::ICommand](#) & [TransferAbort](#)

Description: Aborts immediately the streaming of data block(s).
- [GenApi::ICommand](#) & [TransferPause](#)

Description: Pauses the streaming of data Block(s).
- [GenApi::ICommand](#) & [TransferResume](#)

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.
- [GenApi::IEnumerationT](#) < [TransferTriggerSelectorEnums](#) > & [TransferTriggerSelector](#)

Description: Selects the type of transfer trigger to configure.
- [GenApi::IEnumerationT](#) < [TransferTriggerModeEnums](#) > & [TransferTriggerMode](#)

Description: Controls if the selected trigger is active.
- [GenApi::IEnumerationT](#) < [TransferTriggerSourceEnums](#) > & [TransferTriggerSource](#)

Description: Specifies the signal to use as the trigger source for transfers.
- [GenApi::IEnumerationT](#) < [TransferTriggerActivationEnums](#) > & [TransferTriggerActivation](#)

Description: Specifies the activation mode of the transfer control trigger.
- [GenApi::IEnumerationT](#) < [TransferStatusSelectorEnums](#) > & [TransferStatusSelector](#)

Description: Selects which status of the transfer module to read.
- [GenApi::IBoolean](#) & [TransferStatus](#)

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.
- [GenApi::IEnumerationT](#) < [TransferComponentSelectorEnums](#) > & [TransferComponentSelector](#)

- Description: Selects the color component for the control of the TransferStreamChannel feature.*

 - [GenApi::Integer](#) & [TransferStreamChannel](#)
- Description: Selects the streaming channel that will be used to transfer the selected stream of data.*

 - [GenApi::EnumerationT< Scan3dDistanceUnitEnums >](#) & [Scan3dDistanceUnit](#)
- Description: Specifies the unit used when delivering calibrated distance data.*

 - [GenApi::EnumerationT< Scan3dCoordinateSystemEnums >](#) & [Scan3dCoordinateSystem](#)
- Description: Specifies the Coordinate system to use for the device.*

 - [GenApi::EnumerationT< Scan3dOutputModeEnums >](#) & [Scan3dOutputMode](#)
- Description: Controls the Calibration and data organization of the device, naming the coordinates transmitted.*

 - [GenApi::EnumerationT< Scan3dCoordinateSystemReferenceEnums >](#) & [Scan3dCoordinateSystemReference](#)
- Description: Defines coordinate system reference location.*

 - [GenApi::EnumerationT< Scan3dCoordinateSelectorEnums >](#) & [Scan3dCoordinateSelector](#)
- Description: Selects the individual coordinates in the vectors for 3D information/transformation.*

 - [GenApi::IFloat](#) & [Scan3dCoordinateScale](#)
- Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.*

 - [GenApi::IFloat](#) & [Scan3dCoordinateOffset](#)
- Description: Offset when transforming a pixel from relative coordinates to world coordinates.*

 - [GenApi::IBoolean](#) & [Scan3dInvalidDataFlag](#)
- Description: Enables the definition of a non-valid flag value in the data stream.*

 - [GenApi::IFloat](#) & [Scan3dInvalidDataValue](#)
- Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.*

 - [GenApi::IFloat](#) & [Scan3dAxisMin](#)
- Description: Minimum valid transmitted coordinate value of the selected Axis.*

 - [GenApi::IFloat](#) & [Scan3dAxisMax](#)
- Description: Maximum valid transmitted coordinate value of the selected Axis.*

 - [GenApi::EnumerationT< Scan3dCoordinateTransformSelectorEnums >](#) & [Scan3dCoordinateTransformSelector](#)
- Description: Sets the index to read/write a coordinate transform value.*

 - [GenApi::IFloat](#) & [Scan3dTransformValue](#)
- Description: Specifies the transform value selected.*

 - [GenApi::EnumerationT< Scan3dCoordinateReferenceSelectorEnums >](#) & [Scan3dCoordinateReferenceSelector](#)
- Description: Sets the index to read a coordinate system reference value defining the transform of a point from the current (Anchor or Transformed) system to the reference system.*

 - [GenApi::IFloat](#) & [Scan3dCoordinateReferenceValue](#)
- Description: Returns the reference value selected.*

 - [GenApi::Integer](#) & [ChunkPartSelector](#)
- Description: Selects the part to access in chunk data in a multipart transmission.*

 - [GenApi::EnumerationT< ChunkImageComponentEnums >](#) & [ChunkImageComponent](#)
- Description: Returns the component of the payload image.*

 - [GenApi::Integer](#) & [ChunkPixelDynamicRangeMin](#)
- Description: Returns the minimum value of dynamic range of the image included in the payload.*

 - [GenApi::Integer](#) & [ChunkPixelDynamicRangeMax](#)
- Description: Returns the maximum value of dynamic range of the image included in the payload.*

 - [GenApi::Integer](#) & [ChunkTimestampLatchValue](#)
- Description: Returns the last Timestamp latched with the TimestampLatch command.*

 - [GenApi::Integer](#) & [ChunkLineStatusAll](#)
- Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.*

 - [GenApi::EnumerationT< ChunkCounterSelectorEnums >](#) & [ChunkCounterSelector](#)
- Description: Selects which counter to retrieve data from.*

- [GenApi::Integer](#) & [ChunkCounterValue](#)
Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.
- [GenApi::EnumerationT< 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::EnumerationT< ChunkEncoderSelectorEnums >](#) & [ChunkEncoderSelector](#)
Description: Selects which Encoder to retrieve data from.
- [GenApi::Integer](#) & [ChunkScanLineSelector](#)
Description: Index for vector representation of one chunk value per line in an image.
- [GenApi::Integer](#) & [ChunkEncoderValue](#)
Description: Returns the counter's value of the selected Encoder at the time of the FrameStart in area scan mode or the counter's value at the time of the LineStart selected by ChunkScanLineSelector in LineScan mode.
- [GenApi::EnumerationT< ChunkEncoderStatusEnums >](#) & [ChunkEncoderStatus](#)
Description: Returns the motion status of the selected encoder.
- [GenApi::EnumerationT< ChunkExposureTimeSelectorEnums >](#) & [ChunkExposureTimeSelector](#)
Description: Selects which exposure time is read by the ChunkExposureTime feature.
- [GenApi::Integer](#) & [ChunkLinePitch](#)
Description: Returns the LinePitch of the image included in the payload.
- [GenApi::EnumerationT< ChunkSourceIDEnums >](#) & [ChunkSourceID](#)
Description: Returns the identifier of Source that the image comes from.
- [GenApi::EnumerationT< ChunkRegionIDEnums >](#) & [ChunkRegionID](#)
Description: Returns the identifier of Region that the image comes from.
- [GenApi::Integer](#) & [ChunkTransferBlockID](#)
Description: Returns the unique identifier of the transfer block used to transport the payload.
- [GenApi::EnumerationT< ChunkTransferStreamIDEnums >](#) & [ChunkTransferStreamID](#)
Description: Returns identifier of the stream that generated this block.
- [GenApi::Integer](#) & [ChunkTransferQueueCurrentBlockCount](#)
Description: Returns the current number of blocks in the transfer queue.
- [GenApi::Integer](#) & [ChunkStreamChannelID](#)
Description: Returns identifier of the stream channel used to carry the block.
- [GenApi::EnumerationT< ChunkScan3dDistanceUnitEnums >](#) & [ChunkScan3dDistanceUnit](#)
Description: Returns the Distance Unit of the payload image.
- [GenApi::EnumerationT< ChunkScan3dOutputModeEnums >](#) & [ChunkScan3dOutputMode](#)
Description: Returns the Calibrated Mode of the payload image.
- [GenApi::EnumerationT< ChunkScan3dCoordinateSystemEnums >](#) & [ChunkScan3dCoordinateSystem](#)
Description: Returns the Coordinate System of the image included in the payload.
- [GenApi::EnumerationT< ChunkScan3dCoordinateSystemReferenceEnums >](#) & [ChunkScan3dCoordinateSystemReference](#)
Description: Returns the Coordinate System Position of the image included in the payload.
- [GenApi::EnumerationT< ChunkScan3dCoordinateSelectorEnums >](#) & [ChunkScan3dCoordinateSelector](#)
Description: Selects which Coordinate to retrieve data from.
- [GenApi::IFloat](#) & [ChunkScan3dCoordinateScale](#)
Description: Returns the Scale for the selected coordinate axis of the image included in the payload.
- [GenApi::IFloat](#) & [ChunkScan3dCoordinateOffset](#)
Description: Returns the Offset for the selected coordinate axis of the image included in the payload.
- [GenApi::Boolean](#) & [ChunkScan3dInvalidDataFlag](#)
Description: Returns if a specific non-valid data flag is used in the data stream.
- [GenApi::IFloat](#) & [ChunkScan3dInvalidDataValue](#)
Description: Returns the Invalid Data Value used for the image included in the payload.
- [GenApi::IFloat](#) & [ChunkScan3dAxisMin](#)

- Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.*

 - [GenApi::IFloat](#) & [ChunkScan3dAxisMax](#)

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

 - [GenApi::IEnumerationT](#) < [ChunkScan3dCoordinateTransformSelectorEnums](#) > & [ChunkScan3dCoordinateTransformSelector](#)

Description: Selector for transform values.

 - [GenApi::IFloat](#) & [ChunkScan3dTransformValue](#)

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](#) & [ChunkScan3dCoordinateReferenceValue](#)

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point.

 - [GenApi::Integer](#) & [TestPendingAck](#)

Description: Tests the device's pending acknowledge feature.

 - [GenApi::IEnumerationT](#) < [DeviceTapGeometryEnums](#) > & [DeviceTapGeometry](#)

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

 - [GenApi::IEnumerationT](#) < [GevPhysicalLinkConfigurationEnums](#) > & [GevPhysicalLinkConfiguration](#)

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

 - [GenApi::IEnumerationT](#) < [GevCurrentPhysicalLinkConfigurationEnums](#) > & [GevCurrentPhysicalLinkConfiguration](#)

Description: Indicates the current physical link configuration of the device.

 - [GenApi::Integer](#) & [GevActiveLinkCount](#)

Description: Indicates the current number of active logical links.

 - [GenApi::Boolean](#) & [GevPAUSEFrameReception](#)

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

 - [GenApi::Boolean](#) & [GevPAUSEFrameTransmission](#)

Description: Controls whether PAUSE Frames can be generated on the given logical link.

 - [GenApi::IEnumerationT](#) < [GevIPConfigurationStatusEnums](#) > & [GevIPConfigurationStatus](#)

Description: Reports the current IP configuration status.

 - [GenApi::Integer](#) & [GevDiscoveryAckDelay](#)

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

 - [GenApi::IEnumerationT](#) < [GevGVCPExtendedStatusCodesSelectorEnums](#) > & [GevGVCPExtendedStatusCodesSelector](#)

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

 - [GenApi::Boolean](#) & [GevGVCPExtendedStatusCodes](#)

Description: Enables the generation of extended status codes.

 - [GenApi::Integer](#) & [GevPrimaryApplicationSwitchoverKey](#)

Description: Controls the key to use to authenticate primary application switchover requests.

 - [GenApi::IEnumerationT](#) < [GevGVSPExtendedIDModeEnums](#) > & [GevGVSPExtendedIDMode](#)

Description: Enables the extended IDs mode.

 - [GenApi::Integer](#) & [GevPrimaryApplicationSocket](#)

Description: Returns the UDP source port of the primary application.

 - [GenApi::Integer](#) & [GevPrimaryApplicationIPAddress](#)

Description: Returns the address of the primary application.

 - [GenApi::Boolean](#) & [GevSCCFGPacketResendDestination](#)

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.

 - [GenApi::Boolean](#) & [GevSCCFGAllInTransmission](#)

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

- [GenApi::Integer](#) & [GevSCZoneCount](#)
Description: Reports the number of zones per block transmitted on the selected stream channel.
- [GenApi::Integer](#) & [GevSCZoneDirectionAll](#)
Description: Reports the transmission direction of each zone transmitted on the selected stream channel.
- [GenApi::Boolean](#) & [GevSCZoneConfigurationLock](#)
Description: Controls whether the selected stream channel multi-zone configuration is locked.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesTransmitted](#)
Description: Reports the number of transmitted PAUSE frames.
- [GenApi::Integer](#) & [aPAUSEMACCtrlFramesReceived](#)
Description: Reports the number of received PAUSE frames.
- [GenApi::EnumerationT< CIConfigurationEnums >](#) & [CIConfiguration](#)
Description: This [Camera](#) Link specific feature describes the configuration used by the camera.
- [GenApi::EnumerationT< 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::EnumerationT< CxpLinkConfigurationStatusEnums >](#) & [CxpLinkConfigurationStatus](#)
Description: This feature indicates the current and active Link configuration used by the Device.
- [GenApi::EnumerationT< CxpLinkConfigurationPreferredEnums >](#) & [CxpLinkConfigurationPreferred](#)
Description: Provides the Link configuration that allows the Transmitter Device to operate in its default mode.
- [GenApi::EnumerationT< CxpLinkConfigurationEnums >](#) & [CxpLinkConfiguration](#)
Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.
- [GenApi::Integer](#) & [CxpConnectionSelector](#)
Description: Selects the CoaXPress physical connection to control.
- [GenApi::EnumerationT< CxpConnectionTestModeEnums >](#) & [CxpConnectionTestMode](#)
Description: Enables the test mode for an individual physical connection of the Device.
- [GenApi::Integer](#) & [CxpConnectionTestErrorCount](#)
Description: Reports the current connection error count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).
- [GenApi::Integer](#) & [CxpConnectionTestPacketCount](#)
Description: Reports the current count for test packets recieved by the device on the connection selected by [CxpConnectionSelector](#).
- [GenApi::Command](#) & [CxpPoCxpAuto](#)
Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.
- [GenApi::Command](#) & [CxpPoCxpTurnOff](#)
Description: Disable Power over CoaXPress (PoCXP) for the Link.
- [GenApi::Command](#) & [CxpPoCxpTripReset](#)
Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).
- [GenApi::EnumerationT< CxpPoCxpStatusEnums >](#) & [CxpPoCxpStatus](#)
Description: Returns the Power over CoaXPress (PoCXP) status of the Device.
- [GenApi::Integer](#) & [ChunkInferenceResult](#)
Description: Visibility: Expert.
- [GenApi::Float](#) & [ChunkInferenceConfidence](#)
Description: Visibility: Expert.

Protected Member Functions

- [Camera](#) ()

Additional Inherited Members

10.10.1 Detailed Description

The camera object class.

10.10.2 Constructor & Destructor Documentation

10.10.2.1 `~Camera()`

```
~Camera ( )
```

10.10.2.2 `Camera()`

```
Camera ( ) [protected]
```

10.10.3 Member Function Documentation

10.10.3.1 `Init()`

```
void Init ( ) [virtual]
```

Implements [ICameraBase](#).

10.10.4 Member Data Documentation

10.10.4.1 `AasRoiEnable`

```
GenApi::IBoolean& AasRoiEnable
```

Description:

Controls whether a user-specified ROI is used for auto algorithm that is currently selected by the `AutoAlgorithm`↔ Selector feature.

Visibility:

10.10.4.2 AasRoiHeight

`GenApi::Integer& AasRoiHeight`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

10.10.4.3 AasRoiOffsetX

`GenApi::Integer& AasRoiOffsetX`

Description:

Controls the x-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

10.10.4.4 AasRoiOffsetY

`GenApi::Integer& AasRoiOffsetY`

Description:

Controls the y-offset of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

10.10.4.5 AasRoiWidth

`GenApi::Integer& AasRoiWidth`

Description:

Controls the width of the ROI used by the auto algorithm that is currently selected by the AutoAlgorithmSelector feature.

Visibility:

10.10.4.6 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

10.10.4.7 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

10.10.4.8 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:

10.10.4.9 AcquisitionFrameCount

`GenApi::IInteger& AcquisitionFrameCount`

Description:

Number of images to acquire during a multi frame acquisition.

Visibility:

10.10.4.10 AcquisitionFrameRate

`GenApi::IFloat& AcquisitionFrameRate`

Description: User controlled acquisition frame rate in Hertz Visibility:

10.10.4.11 AcquisitionFrameRateEnable

`GenApi::IBoolen& AcquisitionFrameRateEnable`

Description: If enabled, AcquisitionFrameRate can be used to manually control the frame rate.

Visibility:

10.10.4.12 AcquisitionLineRate

`GenApi::IFloat& AcquisitionLineRate`

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

Visibility:

10.10.4.13 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:

10.10.4.14 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:

10.10.4.15 AcquisitionStart

`GenApi::ICommand& AcquisitionStart`

Description: This command starts the acquisition of images.

Visibility:

10.10.4.16 AcquisitionStatus

`GenApi::IBoolen& AcquisitionStatus`

Description: Reads the state of the internal acquisition signal selected using AcquisitionStatusSelector.

Visibility: Expert

10.10.4.17 AcquisitionStatusSelector

`GenApi::IEnumerationT<AcquisitionStatusSelectorEnums>& AcquisitionStatusSelector`

Description: Selects the internal acquisition signal to read using AcquisitionStatus.

Visibility: Expert

10.10.4.18 AcquisitionStop

`GenApi::ICommand& AcquisitionStop`

Description: This command stops the acquisition of images.

Visibility:

10.10.4.19 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

10.10.4.20 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

10.10.4.21 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

10.10.4.22 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

10.10.4.23 ActionSelector

`GenApi::Integer& ActionSelector`

Description: Selects to which Action Signal further Action settings apply.

Visibility: Guru

10.10.4.24 ActionUnconditionalMode

`GenApi::EnumerationT<ActionUnconditionalModeEnums>& ActionUnconditionalMode`

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

Visibility: Guru

10.10.4.25 AdaptiveCompressionEnable

`GenApi::Boolean& AdaptiveCompressionEnable`

Description: Controls whether lossless compression adapts to the image content.

If disabled, a fixed encoding table is used. Visibility:

10.10.4.26 AdcBitDepth

`GenApi::EnumerationT<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:

10.10.4.27 aPAUSEMACtrlFramesReceived

`GenApi::Integer& aPAUSEMACtrlFramesReceived`

Description: Reports the number of received PAUSE frames.

Visibility: Guru

10.10.4.28 aPAUSEMACtrlFramesTransmitted

`GenApi::Integer& aPAUSEMACtrlFramesTransmitted`

Description: Reports the number of transmitted PAUSE frames.

Visibility: Guru

10.10.4.29 AutoAlgorithmSelector

`GenApi::EnumerationT<AutoAlgorithmSelectorEnums>& AutoAlgorithmSelector`

Description:

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

Visibility:

10.10.4.30 AutoExposureControlLoopDamping

`GenApi::Float& 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:

10.10.4.31 AutoExposureControlPriority

`GenApi::EnumerationT<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:

10.10.4.32 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:

10.10.4.33 AutoExposureExposureTimeLowerLimit

`GenApi::IFloat& AutoExposureExposureTimeLowerLimit`

Description:

The smallest exposure time that auto exposure can set.

Visibility:

10.10.4.34 AutoExposureExposureTimeUpperLimit

`GenApi::IFloat& AutoExposureExposureTimeUpperLimit`

Description:

The largest exposure time that auto exposure can set.

Visibility:

10.10.4.35 AutoExposureGainLowerLimit

`GenApi::IFloat& AutoExposureGainLowerLimit`

Description:

The smallest gain that auto exposure can set.

Visibility:

10.10.4.36 AutoExposureGainUpperLimit

`GenApi::IFloat& AutoExposureGainUpperLimit`

Description:

The largest gain that auto exposure can set.

Visibility:

10.10.4.37 AutoExposureGreyValueLowerLimit

`GenApi::IFloat& AutoExposureGreyValueLowerLimit`

Description:

The lowest value in percentage that the target mean may reach.

Visibility:

10.10.4.38 AutoExposureGreyValueUpperLimit

`GenApi::IFloat& AutoExposureGreyValueUpperLimit`

Description:

The highest value in percentage that the target mean may reach.

Visibility:

10.10.4.39 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:

10.10.4.40 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:

10.10.4.41 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:

10.10.4.42 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:

10.10.4.43 BalanceRatio

`GenApi::IFloat& BalanceRatio`

Description:

Controls the balance ratio of the selected color relative to green.

Used for white balancing.

Visibility:

10.10.4.44 BalanceRatioSelector

`GenApi::IEnumerationT<BalanceRatioSelectorEnums>& BalanceRatioSelector`

Description:

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

Visibility:

10.10.4.45 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:

10.10.4.46 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:

10.10.4.47 BalanceWhiteAutoLowerLimit

`GenApi::IFloat& BalanceWhiteAutoLowerLimit`

Description:

Controls the minimum value Auto White Balance can set for the Red/Blue BalanceRatio.

Visibility:

10.10.4.48 BalanceWhiteAutoProfile

`GenApi::IEnumerationT<BalanceWhiteAutoProfileEnums>& BalanceWhiteAutoProfile`

Description: Selects the profile used by BalanceWhiteAuto.

Visibility:

10.10.4.49 BalanceWhiteAutoUpperLimit

`GenApi::IFloat& BalanceWhiteAutoUpperLimit`

Description:

Controls the maximum value Auto White Balance can set the Red/Blue BalanceRatio.

Visibility:

10.10.4.50 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:

10.10.4.51 BinningHorizontalMode

`GenApi::IEnumerationT<BinningHorizontalModeEnums>& BinningHorizontalMode`

Description: Visibility:

10.10.4.52 BinningSelector

`GenApi::IEnumerationT<BinningSelectorEnums>& BinningSelector`

Description:

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

Visibility:

10.10.4.53 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:

10.10.4.54 BinningVerticalMode

`GenApi::IEnumerationT<BinningVerticalModeEnums>& BinningVerticalMode`

Description: Visibility:

10.10.4.55 BlackLevel

`GenApi::IFloat& BlackLevel`

Description:

Controls the offset of the video signal in percent.

Visibility:

10.10.4.56 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

10.10.4.57 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

10.10.4.58 BlackLevelClampingEnable

```
GenApi::IBoolean& BlackLevelClampingEnable
```

Description:

Enable the black level auto clamping feature which performing dark current compensation.

Visibility:

10.10.4.59 BlackLevelRaw

```
GenApi::IInteger& BlackLevelRaw
```

Description:

Controls the offset of the video signal in camera specific units.

Visibility:

10.10.4.60 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:

10.10.4.61 ChunkBlackLevel

`GenApi::IFloat& ChunkBlackLevel`

Description: Returns the black level used to capture the image.

Visibility:

10.10.4.62 ChunkBlackLevelSelector

`GenApi::IEnumerationT<ChunkBlackLevelSelectorEnums>& ChunkBlackLevelSelector`

Description: Selects which black level to retrieve Visibility:

10.10.4.63 ChunkCounterSelector

`GenApi::IEnumerationT<ChunkCounterSelectorEnums>& ChunkCounterSelector`

Description: Selects which counter to retrieve data from.

Visibility: Expert

10.10.4.64 ChunkCounterValue

`GenApi::IInteger& ChunkCounterValue`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event.

Visibility: Expert

10.10.4.65 ChunkCRC

`GenApi::IInteger& ChunkCRC`

Description: Returns the CRC of the image payload.

Visibility:

10.10.4.66 ChunkEnable

`GenApi::IBoolean& ChunkEnable`

Description: Enables the inclusion of the selected Chunk data in the payload of the image.

Visibility:

10.10.4.67 ChunkEncoderSelector

`GenApi::IEnumerationT<ChunkEncoderSelectorEnums>& ChunkEncoderSelector`

Description: Selects which Encoder to retrieve data from.

Visibility: Expert

10.10.4.68 ChunkEncoderStatus

`GenApi::IEnumerationT<ChunkEncoderStatusEnums>& ChunkEncoderStatus`

Description: Returns the motion status of the selected encoder.

Visibility: Expert

10.10.4.69 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

10.10.4.70 ChunkExposureEndLineStatusAll

`GenApi::IInteger& ChunkExposureEndLineStatusAll`

Description: Returns the status of all the I/O lines at the end of exposure event.

Visibility:

10.10.4.71 ChunkExposureTime

`GenApi::IFloat& ChunkExposureTime`

Description: Returns the exposure time used to capture the image.

Visibility:

10.10.4.72 ChunkExposureTimeSelector

`GenApi::IEnumerationT<ChunkExposureTimeSelectorEnums>& ChunkExposureTimeSelector`

Description: Selects which exposure time is read by the ChunkExposureTime feature.

Visibility: Expert

10.10.4.73 ChunkFrameID

`GenApi::Integer& ChunkFrameID`

Description: Returns the image frame ID.

Visibility:

10.10.4.74 ChunkGain

`GenApi::IFloat& ChunkGain`

Description: Returns the gain used to capture the image.

Visibility:

10.10.4.75 ChunkGainSelector

`GenApi::EnumerationT<ChunkGainSelectorEnums>& ChunkGainSelector`

Description: Selects which gain to retrieve Visibility:

10.10.4.76 ChunkHeight

`GenApi::Integer& ChunkHeight`

Description: Returns the height of the image included in the payload.

Visibility:

10.10.4.77 ChunkImage

`GenApi::Integer& ChunkImage`

Description: Returns the image payload.

Visibility:

10.10.4.78 ChunkImageComponent

`GenApi::EnumerationT<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

10.10.4.79 ChunkInferenceConfidence

`GenApi::IFloat& ChunkInferenceConfidence`

Description: Visibility: Expert.

10.10.4.80 ChunkInferenceResult

`GenApi::IInteger& ChunkInferenceResult`

Description: Visibility: Expert.

10.10.4.81 ChunkLinePitch

`GenApi::IInteger& ChunkLinePitch`

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

10.10.4.82 ChunkLineStatusAll

`GenApi::IInteger& ChunkLineStatusAll`

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event.

Visibility: Expert

10.10.4.83 ChunkModeActive

`GenApi::IBoolean& ChunkModeActive`

Description: Activates the inclusion of Chunk data in the payload of the image.

Visibility:

10.10.4.84 ChunkOffsetX

`GenApi::IInteger& ChunkOffsetX`

Description: Returns the Offset X of the image included in the payload.

Visibility:

10.10.4.85 ChunkOffsetY

`GenApi::Integer& ChunkOffsetY`

Description: Returns the Offset Y of the image included in the payload.

Visibility:

10.10.4.86 ChunkPartSelector

`GenApi::Integer& ChunkPartSelector`

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

10.10.4.87 ChunkPixelDynamicRangeMax

`GenApi::Integer& ChunkPixelDynamicRangeMax`

Description: Returns the maximum value of dynamic range of the image included in the payload.

Visibility: Expert

10.10.4.88 ChunkPixelDynamicRangeMin

`GenApi::Integer& ChunkPixelDynamicRangeMin`

Description: Returns the minimum value of dynamic range of the image included in the payload.

Visibility: Expert

10.10.4.89 ChunkPixelFormat

`GenApi::EnumerationT<ChunkPixelFormatEnums>& ChunkPixelFormat`

Description: Format of the pixel provided by the camera Visibility:

10.10.4.90 ChunkRegionID

`GenApi::EnumerationT<ChunkRegionIDEnums>& ChunkRegionID`

Description: Returns the identifier of Region that the image comes from.

Visibility: Expert

10.10.4.91 ChunkScan3dAxisMax

`GenApi::IFloat& ChunkScan3dAxisMax`

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.10.4.92 ChunkScan3dAxisMin

`GenApi::IFloat& ChunkScan3dAxisMin`

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.10.4.93 ChunkScan3dCoordinateOffset

`GenApi::IFloat& ChunkScan3dCoordinateOffset`

Description: Returns the Offset for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.10.4.94 ChunkScan3dCoordinateReferenceSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateReferenceSelectorEnums>& ChunkScan3dCoordinate↔
ReferenceSelector`

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

Visibility: Expert

10.10.4.95 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

10.10.4.96 ChunkScan3dCoordinateScale

`GenApi::IFloat& ChunkScan3dCoordinateScale`

Description: Returns the Scale for the selected coordinate axis of the image included in the payload.

Visibility: Expert

10.10.4.97 ChunkScan3dCoordinateSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateSelectorEnums>& ChunkScan3dCoordinateSelector`

Description: Selects which Coordinate to retrieve data from.

Visibility: Expert

10.10.4.98 ChunkScan3dCoordinateSystem

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemEnums>& ChunkScan3dCoordinateSystem`

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

Visibility: Expert

10.10.4.99 ChunkScan3dCoordinateSystemReference

`GenApi::IEnumerationT<ChunkScan3dCoordinateSystemReferenceEnums>& ChunkScan3dCoordinate↵
SystemReference`

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

Visibility: Expert

10.10.4.100 ChunkScan3dCoordinateTransformSelector

`GenApi::IEnumerationT<ChunkScan3dCoordinateTransformSelectorEnums>& ChunkScan3dCoordinate↵
TransformSelector`

Description: Selector for transform values.

Visibility: Expert

10.10.4.101 ChunkScan3dDistanceUnit

`GenApi::IEnumerationT<ChunkScan3dDistanceUnitEnums>& ChunkScan3dDistanceUnit`

Description: Returns the Distance Unit of the payload image.

Visibility: Expert

10.10.4.102 ChunkScan3dInvalidDataFlag

`GenApi::IBoolean& ChunkScan3dInvalidDataFlag`

Description: Returns if a specific non-valid data flag is used in the data stream.

Visibility: Expert

10.10.4.103 ChunkScan3dInvalidDataValue

`GenApi::IFloat& ChunkScan3dInvalidDataValue`

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

10.10.4.104 ChunkScan3dOutputMode

`GenApi::IEnumerationT<ChunkScan3dOutputModeEnums>& ChunkScan3dOutputMode`

Description: Returns the Calibrated Mode of the payload image.

Visibility: Expert

10.10.4.105 ChunkScan3dTransformValue

`GenApi::IFloat& ChunkScan3dTransformValue`

Description: Returns the transform value.

Visibility: Expert

10.10.4.106 ChunkScanLineSelector

`GenApi::IInteger& ChunkScanLineSelector`

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

10.10.4.107 ChunkSelector

`GenApi::IEnumerationT<ChunkSelectorEnums>& ChunkSelector`

Description: Selects which chunk data to enable or disable.

Visibility:

10.10.4.108 ChunkSequencerSetActive

`GenApi::IInteger& ChunkSequencerSetActive`

Description: Returns the index of the active set of the running sequencer included in the payload.

Visibility:

10.10.4.109 ChunkSerialData

`GenApi::IString& ChunkSerialData`

Description: Returns the serial data that was received.

Visibility:

10.10.4.110 ChunkSerialDataLength

`GenApi::IInteger& ChunkSerialDataLength`

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

10.10.4.111 ChunkSerialReceiveOverflow

`GenApi::IBoolean& ChunkSerialReceiveOverflow`

Description: Returns the status of the chunk serial receive overflow.

Visibility:

10.10.4.112 ChunkSourceID

`GenApi::IEnumerationT<ChunkSourceIDEnums>& ChunkSourceID`

Description: Returns the identifier of Source that the image comes from.

Visibility: Expert

10.10.4.113 ChunkStreamChannelID

`GenApi::IInteger& ChunkStreamChannelID`

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

10.10.4.114 ChunkTimerSelector

`GenApi::IEnumerationT<ChunkTimerSelectorEnums>& ChunkTimerSelector`

Description: Selects which Timer to retrieve data from.

Visibility: Expert

10.10.4.115 ChunkTimerValue

`GenApi::IFloat& ChunkTimerValue`

Description: Returns the value of the selected Timer at the time of the FrameStart internal event.

Visibility: Expert

10.10.4.116 ChunkTimestamp

`GenApi::IInteger& ChunkTimestamp`

Description: Returns the Timestamp of the image.

Visibility:

10.10.4.117 ChunkTimestampLatchValue

`GenApi::IInteger& ChunkTimestampLatchValue`

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

10.10.4.118 ChunkTransferBlockID

`GenApi::IInteger& ChunkTransferBlockID`

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

10.10.4.119 ChunkTransferQueueCurrentBlockCount

`GenApi::IInteger& ChunkTransferQueueCurrentBlockCount`

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

10.10.4.120 ChunkTransferStreamID

`GenApi::IEnumerationT<ChunkTransferStreamIDEnums>& ChunkTransferStreamID`

Description: Returns identifier of the stream that generated this block.

Visibility: Expert

10.10.4.121 ChunkWidth

`GenApi::Integer& ChunkWidth`

Description: Returns the width of the image included in the payload.

Visibility:

10.10.4.122 ClConfiguration

`GenApi::EnumerationT<ClConfigurationEnums>& ClConfiguration`

Description: This [Camera](#) Link specific feature describes the configuration used by the camera.

It helps especially when a camera is capable of operation in a non-standard configuration, and when the features PixelSize, SensorDigitizationTaps, and DeviceTapGeometry do not provide enough information for interpretation of the image data provided by the camera. Visibility: Beginner

10.10.4.123 ClTimeSlotsCount

`GenApi::EnumerationT<ClTimeSlotsCountEnums>& ClTimeSlotsCount`

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

Visibility: Expert

10.10.4.124 ColorTransformationEnable

`GenApi::Boolean& 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:

10.10.4.125 ColorTransformationSelector

`GenApi::EnumerationT<ColorTransformationSelectorEnums>& ColorTransformationSelector`

Description: Selects which Color Transformation module is controlled by the various Color Transformation features.

Visibility:

10.10.4.126 ColorTransformationValue

```
GenApi::IFloat& ColorTransformationValue
```

Description:

Represents the value of the selected Gain factor or Offset inside the Transformation matrix in floating point precision.

Visibility:

10.10.4.127 ColorTransformationValueSelector

```
GenApi::IEnumerationT<ColorTransformationValueSelectorEnums>& ColorTransformationValueSelector
```

Description:

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

Visibility:

10.10.4.128 CompressionRatio

```
GenApi::IFloat& CompressionRatio
```

Description: Reports the ratio between the uncompressed image size and compressed image size.

Visibility:

10.10.4.129 CounterDelay

```
GenApi::IInteger& CounterDelay
```

Description: Sets the delay (or number of events) before the CounterStart event is generated.

Visibility:

10.10.4.130 CounterDuration

```
GenApi::IInteger& CounterDuration
```

Description: Sets the duration (or number of events) before the CounterEnd event is generated.

Visibility:

10.10.4.131 CounterEventActivation

`GenApi::IEnumerationT<CounterEventActivationEnums>& CounterEventActivation`

Description: Selects the activation mode of the event to increment the Counter.

Visibility:

10.10.4.132 CounterEventSource

`GenApi::IEnumerationT<CounterEventSourceEnums>& CounterEventSource`

Description: Selects the event that will increment the counter Visibility:

10.10.4.133 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

10.10.4.134 CounterResetActivation

`GenApi::IEnumerationT<CounterResetActivationEnums>& CounterResetActivation`

Description: Selects the Activation mode of the Counter Reset Source signal.

Visibility:

10.10.4.135 CounterResetSource

`GenApi::IEnumerationT<CounterResetSourceEnums>& CounterResetSource`

Description: Selects the signal that will be the source to reset the Counter.

Visibility:

10.10.4.136 CounterSelector

`GenApi::IEnumerationT<CounterSelectorEnums>& CounterSelector`

Description: Selects which counter to configure Visibility:

10.10.4.137 CounterStatus

`GenApi::IEnumerationT<CounterStatusEnums>& CounterStatus`

Description: Returns the current status of the Counter.

Visibility:

10.10.4.138 CounterTriggerActivation

`GenApi::IEnumerationT<CounterTriggerActivationEnums>& CounterTriggerActivation`

Description: Selects the activation mode of the trigger to start the Counter.

Visibility:

10.10.4.139 CounterTriggerSource

`GenApi::IEnumerationT<CounterTriggerSourceEnums>& CounterTriggerSource`

Description: Selects the source of the trigger to start the counter Visibility:

10.10.4.140 CounterValue

`GenApi::IInteger& CounterValue`

Description: Current counter value Visibility:

10.10.4.141 CounterValueAtReset

`GenApi::IInteger& CounterValueAtReset`

Description: Value of the selected Counter when it was reset by a trigger.

Visibility:

10.10.4.142 CxpConnectionSelector

`GenApi::IInteger& CxpConnectionSelector`

Description: Selects the CoaXPress physical connection to control.

Visibility: Expert

10.10.4.143 CxpConnectionTestErrorCount

`GenApi::Integer& CxpConnectionTestErrorCount`

Description: Reports the current connection error count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

10.10.4.144 CxpConnectionTestMode

`GenApi::EnumerationT<CxpConnectionTestModeEnums>& CxpConnectionTestMode`

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

Visibility: Expert

10.10.4.145 CxpConnectionTestPacketCount

`GenApi::Integer& CxpConnectionTestPacketCount`

Description: Reports the current count for test packets recieved by the device on the connection selected by CxpConnectionSelector.

Visibility: Expert

10.10.4.146 CxpLinkConfiguration

`GenApi::EnumerationT<CxpLinkConfigurationEnums>& CxpLinkConfiguration`

Description: This feature allows specifying the Link configuration for the communication between the Receiver and Transmitter Device.

In most cases this feature does not need to be written because automatic discovery will set configuration correctly to the value returned by CxpLinkConfigurationPreferred. Note that the currently active configuration of the Link can be read using CxpLinkConfigurationStatus. Visibility: Beginner

10.10.4.147 CxpLinkConfigurationPreferred

`GenApi::EnumerationT<CxpLinkConfigurationPreferredEnums>& CxpLinkConfigurationPreferred`

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

Visibility: Expert

10.10.4.148 CxpLinkConfigurationStatus

`GenApi::EnumerationT<CxpLinkConfigurationStatusEnums>& CxpLinkConfigurationStatus`

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

Visibility: Beginner

10.10.4.149 CxpPoCxpAuto

```
GenApi::ICommand& CxpPoCxpAuto
```

Description: Activate automatic control of the Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

10.10.4.150 CxpPoCxpStatus

```
GenApi::IEnumerationT<CxpPoCxpStatusEnums>& CxpPoCxpStatus
```

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

Visibility: Expert

10.10.4.151 CxpPoCxpTripReset

```
GenApi::ICommand& CxpPoCxpTripReset
```

Description: Reset the Power over CoaXPress (PoCXP) Link after an over-current trip on the Device connection(s).

Visibility: Expert

10.10.4.152 CxpPoCxpTurnOff

```
GenApi::ICommand& CxpPoCxpTurnOff
```

Description: Disable Power over CoaXPress (PoCXP) for the Link.

Visibility: Expert

10.10.4.153 DecimationHorizontal

```
GenApi::IInteger& 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:

10.10.4.154 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:

10.10.4.155 DecimationSelector

`GenApi::IEnumerationT<DecimationSelectorEnums>& DecimationSelector`

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

Visibility:

10.10.4.156 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:

10.10.4.157 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:

10.10.4.158 DefectCorrectionMode

`GenApi::IEnumerationT<DefectCorrectionModeEnums>& DefectCorrectionMode`

Description: Controls the method used for replacing defective pixels.

Visibility:

10.10.4.159 DefectCorrectStaticEnable

`GenApi::IBoolean& DefectCorrectStaticEnable`

Description: Enables/Disables table-based defective pixel correction.

Visibility:

10.10.4.160 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:

10.10.4.161 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:

10.10.4.162 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:

10.10.4.163 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:

10.10.4.164 DefectTableIndex

`GenApi::IInteger& DefectTableIndex`

Description:

Controls the offset of the element to access in the defective pixel location table.

Visibility:

10.10.4.165 DefectTablePixelCount

`GenApi::IInteger& DefectTablePixelCount`

Description:

The number of defective pixel locations in the current table.

Visibility:

10.10.4.166 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:

10.10.4.167 Deinterlacing

`GenApi::IEnumerationT<DeinterlacingEnums>& Deinterlacing`

Description: Controls how the device performs de-interlacing.

Visibility: Beginner

10.10.4.168 DeviceCharacterSet

```
GenApi::IEnumerationT<DeviceCharacterSetEnums>& DeviceCharacterSet
```

Description:

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

Visibility:

10.10.4.169 DeviceClockFrequency

```
GenApi::IFloat& DeviceClockFrequency
```

Description: Returns the frequency of the selected Clock.

Visibility: Expert

10.10.4.170 DeviceClockSelector

```
GenApi::IEnumerationT<DeviceClockSelectorEnums>& DeviceClockSelector
```

Description: Selects the clock frequency to access from the device.

Visibility: Expert

10.10.4.171 DeviceConnectionSelector

```
GenApi::IInteger& DeviceConnectionSelector
```

Description: Selects which Connection of the device to control.

Visibility: Beginner

10.10.4.172 DeviceConnectionSpeed

```
GenApi::IInteger& DeviceConnectionSpeed
```

Description: Indicates the speed of transmission of the specified Connection Visibility: Expert.

10.10.4.173 DeviceConnectionStatus

```
GenApi::IEnumerationT<DeviceConnectionStatusEnums>& DeviceConnectionStatus
```

Description: Indicates the status of the specified Connection.

Visibility: Expert

10.10.4.174 DeviceEventChannelCount

`GenApi::Integer& DeviceEventChannelCount`

Description:

Indicates the number of event channels supported by the device.

Visibility:

10.10.4.175 DeviceFamilyName

`GenApi::IString& DeviceFamilyName`

Description: Identifier of the product family of the device.

Visibility: Beginner

10.10.4.176 DeviceFeaturePersistenceEnd

`GenApi::ICommand& DeviceFeaturePersistenceEnd`

Description: Indicate to the device the end of feature persistence.

Visibility: Guru

10.10.4.177 DeviceFeaturePersistenceStart

`GenApi::ICommand& DeviceFeaturePersistenceStart`

Description: Indicate to the device and [GenICam](#) XML to get ready for persisting of all streamable features.

Visibility: Guru

10.10.4.178 DeviceFirmwareVersion

`GenApi::IString& DeviceFirmwareVersion`

Description: Version of the firmware on the device.

Visibility:

10.10.4.179 DeviceGenCPVersionMajor

`GenApi::Integer& DeviceGenCPVersionMajor`

Description: Major version of the GenCP protocol supported by the device.

Visibility: Beginner

10.10.4.180 DeviceGenCPVersionMinor

`GenApi::Integer& DeviceGenCPVersionMinor`

Description: Minor version of the GenCP protocol supported by the device.

Visibility: Beginner

10.10.4.181 DeviceID

`GenApi::String& DeviceID`

Description: Device identifier (serial number).

Visibility:

10.10.4.182 DeviceIndicatorMode

`GenApi::EnumerationT<DeviceIndicatorModeEnums>& DeviceIndicatorMode`

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

Visibility:

10.10.4.183 DeviceLinkBandwidthReserve

`GenApi::Float& DeviceLinkBandwidthReserve`

Description:

Percentage of streamed data bandwidth reserved for packet resend.

Visibility:

10.10.4.184 DeviceLinkCommandTimeout

`GenApi::Float& 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

10.10.4.185 DeviceLinkConnectionCount

`GenApi::Integer& DeviceLinkConnectionCount`

Description: Returns the number of physical connection of the device used by a particular Link.

Visibility: Beginner

10.10.4.186 DeviceLinkCurrentThroughput

`GenApi::Integer& DeviceLinkCurrentThroughput`

Description: Current bandwidth of streamed data.

Visibility:

10.10.4.187 DeviceLinkHeartbeatMode

`GenApi::EnumerationT<DeviceLinkHeartbeatModeEnums>& DeviceLinkHeartbeatMode`

Description: Activate or deactivate the Link's heartbeat.

Visibility: Expert

10.10.4.188 DeviceLinkHeartbeatTimeout

`GenApi::Float& DeviceLinkHeartbeatTimeout`

Description: Controls the current heartbeat timeout of the specific Link.

Visibility: Guru

10.10.4.189 DeviceLinkSelector

`GenApi::Integer& DeviceLinkSelector`

Description: Selects which Link of the device to control.

Visibility: Beginner

10.10.4.190 DeviceLinkSpeed

`GenApi::Integer& DeviceLinkSpeed`

Description:

Indicates the speed of transmission negotiated on the specified Link.

(Bps)

Visibility:

10.10.4.191 DeviceLinkThroughputLimit

`GenApi::Integer& DeviceLinkThroughputLimit`

Description:

Limits the maximum bandwidth of the data that will be streamed out by the device on the selected Link.

If necessary, delays will be uniformly inserted between transport layer packets in order to control the peak bandwidth.

Visibility:

10.10.4.192 DeviceLinkThroughputLimitMode

`GenApi::EnumerationT<DeviceLinkThroughputLimitModeEnums>& DeviceLinkThroughputLimitMode`

Description: Controls if the DeviceLinkThroughputLimit is active.

When disabled, lower level TL specific features are expected to control the throughput. When enabled, DeviceLinkThroughputLimit controls the overall throughput. Visibility: Expert

10.10.4.193 DeviceManifestEntrySelector

`GenApi::Integer& DeviceManifestEntrySelector`

Description: Selects the manifest entry to reference.

Visibility: Guru

10.10.4.194 DeviceManifestPrimaryURL

`GenApi::String& DeviceManifestPrimaryURL`

Description: Indicates the first URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

10.10.4.195 DeviceManifestSchemaMajorVersion

`GenApi::Integer& DeviceManifestSchemaMajorVersion`

Description: Indicates the major version number of the schema file of the selected manifest entry.

Visibility: Guru

10.10.4.196 DeviceManifestSchemaMinorVersion

`GenApi::Integer& DeviceManifestSchemaMinorVersion`

Description: Indicates the minor version number of the schema file of the selected manifest entry.

Visibility: Guru

10.10.4.197 DeviceManifestSecondaryURL

`GenApi::String& DeviceManifestSecondaryURL`

Description: Indicates the second URL to the [GenICam](#) XML device description file of the selected manifest entry.

Visibility: Guru

10.10.4.198 DeviceManifestXMLMajorVersion

`GenApi::Integer& DeviceManifestXMLMajorVersion`

Description: Indicates the major version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.10.4.199 DeviceManifestXMLMinorVersion

`GenApi::Integer& DeviceManifestXMLMinorVersion`

Description: Indicates the minor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.10.4.200 DeviceManifestXMLSubMinorVersion

`GenApi::Integer& DeviceManifestXMLSubMinorVersion`

Description: Indicates the subminor version number of the [GenICam](#) XML file of the selected manifest entry.

Visibility: Guru

10.10.4.201 DeviceManufacturerInfo

`GenApi::String& DeviceManufacturerInfo`

Description: Manufacturer information about the device.

Visibility:

10.10.4.202 DeviceMaxThroughput

`GenApi::Integer& 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:

10.10.4.203 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Model of the device.

Visibility:

10.10.4.204 DevicePowerSupplySelector

`GenApi::IEnumerationT<DevicePowerSupplySelectorEnums>& DevicePowerSupplySelector`

Description:

Selects the power supply source to control or read.

Visibility:

10.10.4.205 DeviceRegistersCheck

`GenApi::ICommand& DeviceRegistersCheck`

Description: Perform the validation of the current register set for consistency.

This will update the DeviceRegistersValid flag. Visibility: Expert

10.10.4.206 DeviceRegistersEndianness

`GenApi::IEnumerationT<DeviceRegistersEndiannessEnums>& DeviceRegistersEndianness`

Description: Endianness of the registers of the device.

Visibility:

10.10.4.207 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

10.10.4.208 DeviceRegistersStreamingStart

`GenApi::ICommand& DeviceRegistersStreamingStart`

Description: Prepare the device for registers streaming without checking for consistency.

Visibility: Guru

10.10.4.209 DeviceRegistersValid

`GenApi::IBoolean& DeviceRegistersValid`

Description: Returns if the current register set is valid and consistent.

Visibility: Expert

10.10.4.210 DeviceReset

`GenApi::ICommand& DeviceReset`

Description: This is a command that immediately resets and reboots the device.

Visibility:

10.10.4.211 DeviceScanType

`GenApi::IEnumerationT<DeviceScanTypeEnums>& DeviceScanType`

Description: Scan type of the sensor of the device.

Visibility:

10.10.4.212 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description:

Device's serial number.

This string is a unique identifier of the device.

Visibility:

10.10.4.213 DeviceSerialPortBaudRate

`GenApi::IEnumerationT<DeviceSerialPortBaudRateEnums>& DeviceSerialPortBaudRate`

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

Visibility: Expert

10.10.4.214 DeviceSerialPortSelector

`GenApi::IEnumerationT<DeviceSerialPortSelectorEnums>& DeviceSerialPortSelector`

Description: Selects which serial port of the device to control.

Visibility: Expert

10.10.4.215 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

10.10.4.216 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

10.10.4.217 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

10.10.4.218 DeviceStreamChannelCount

`GenApi::IInteger& DeviceStreamChannelCount`

Description:

Indicates the number of streaming channels supported by the device.

Visibility:

10.10.4.219 DeviceStreamChannelEndianness

`GenApi::IEnumerationT<DeviceStreamChannelEndiannessEnums>& DeviceStreamChannelEndianness`

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

Visibility: Guru

10.10.4.220 DeviceStreamChannelLink

`GenApi::IInteger& DeviceStreamChannelLink`

Description: Index of device's Link to use for streaming the specified stream channel.

Visibility: Guru

10.10.4.221 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

10.10.4.222 DeviceStreamChannelSelector

`GenApi::IInteger& DeviceStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility: Expert

10.10.4.223 DeviceStreamChannelType

`GenApi::IEnumerationT<DeviceStreamChannelTypeEnums>& DeviceStreamChannelType`

Description: Reports the type of the stream channel.

Visibility: Guru

10.10.4.224 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

10.10.4.225 DeviceTemperature

`GenApi::IFloat& DeviceTemperature`

Description: Device temperature in degrees Celsius (C).

Visibility:

10.10.4.226 DeviceTemperatureSelector

`GenApi::IEnumerationT<DeviceTemperatureSelectorEnums>& DeviceTemperatureSelector`

Description:

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

Visibility:

10.10.4.227 DeviceTLType

`GenApi::IEnumerationT<DeviceTLTypeEnums>& DeviceTLType`

Description: Transport Layer type of the device.

Visibility:

10.10.4.228 DeviceTLVersionMajor

`GenApi::IInteger& DeviceTLVersionMajor`

Description:

Major version of the Transport Layer of the device.

Visibility:

10.10.4.229 DeviceTLVersionMinor

`GenApi::IInteger& DeviceTLVersionMinor`

Description:

Minor version of the Transport Layer of the device.

Visibility:

10.10.4.230 DeviceTLVersionSubMinor

`GenApi::Integer& DeviceTLVersionSubMinor`

Description: Sub minor version of the Transport Layer of the device.

Visibility: Beginner

10.10.4.231 DeviceType

`GenApi::EnumerationT<DeviceTypeEnum>& DeviceType`

Description: Returns the device type.

Visibility: Guru

10.10.4.232 DeviceUptime

`GenApi::Integer& DeviceUptime`

Description: Total time since the device was powered up in seconds.

Visibility:

10.10.4.233 DeviceUserID

`GenApi::String& DeviceUserID`

Description: User-programmable device identifier.

Visibility:

10.10.4.234 DeviceVendorName

`GenApi::String& DeviceVendorName`

Description: Name of the manufacturer of the device.

Visibility:

10.10.4.235 DeviceVersion

`GenApi::String& DeviceVersion`

Description: Version of the device.

Visibility:

10.10.4.236 EncoderDivider

`GenApi::Integer& EncoderDivider`

Description: Sets how many Encoder increment/decrements that are needed generate an Encoder output pulse signal.

Visibility: Expert

10.10.4.237 EncoderMode

`GenApi::EnumerationT<EncoderModeEnums>& EncoderMode`

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

Visibility: Expert

10.10.4.238 EncoderOutputMode

`GenApi::EnumerationT<EncoderOutputModeEnums>& EncoderOutputMode`

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

Visibility: Expert

10.10.4.239 EncoderReset

`GenApi::Command& 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

10.10.4.240 EncoderResetActivation

`GenApi::EnumerationT<EncoderResetActivationEnums>& EncoderResetActivation`

Description: Selects the Activation mode of the Encoder Reset Source signal.

Visibility: Expert

10.10.4.241 EncoderResetSource

`GenApi::EnumerationT<EncoderResetSourceEnums>& EncoderResetSource`

Description: Selects the signals that will be the source to reset the Encoder.

Visibility: Expert

10.10.4.242 EncoderSelector

```
GenApi::IEnumerationT<EncoderSelectorEnums>& EncoderSelector
```

Description: Selects which Encoder to configure.

Visibility: Expert

10.10.4.243 EncoderSourceA

```
GenApi::IEnumerationT<EncoderSourceAEnums>& EncoderSourceA
```

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

Visibility: Expert

10.10.4.244 EncoderSourceB

```
GenApi::IEnumerationT<EncoderSourceBEnums>& EncoderSourceB
```

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

Visibility: Expert

10.10.4.245 EncoderStatus

```
GenApi::IEnumerationT<EncoderStatusEnums>& EncoderStatus
```

Description: Returns the motion status of the encoder.

Visibility: Expert

10.10.4.246 EncoderTimeout

```
GenApi::IFloat& EncoderTimeout
```

Description: Sets the maximum time interval between encoder counter increments before the status turns to static.

Visibility: Expert

10.10.4.247 EncoderValue

```
GenApi::IInteger& EncoderValue
```

Description: Reads or writes the current value of the position counter of the selected Encoder.

Visibility: Expert

10.10.4.248 EncoderValueAtReset

`GenApi::Integer& EncoderValueAtReset`

Description: Reads the value of the of the position counter of the selected Encoder when it was reset by a signal or by an explicit EncoderReset command.

Visibility: Expert

10.10.4.249 EnumerationCount

`GenApi::Integer& EnumerationCount`

Description: Number of enumerations since uptime.

Visibility:

10.10.4.250 EventAcquisitionEnd

`GenApi::Integer& EventAcquisitionEnd`

Description: Returns the unique Identifier of the Acquisition End type of [Event](#).

Visibility: Expert

10.10.4.251 EventAcquisitionEndFrameID

`GenApi::Integer& EventAcquisitionEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition End [Event](#).

Visibility: Expert

10.10.4.252 EventAcquisitionEndTimestamp

`GenApi::Integer& EventAcquisitionEndTimestamp`

Description: Returns the Timestamp of the Acquisition End [Event](#).

Visibility: Expert

10.10.4.253 EventAcquisitionError

`GenApi::Integer& EventAcquisitionError`

Description: Returns the unique Identifier of the Acquisition Error type of [Event](#).

Visibility: Expert

10.10.4.254 EventAcquisitionErrorFrameID

`GenApi::Integer& EventAcquisitionErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Error [Event](#).

Visibility: Expert

10.10.4.255 EventAcquisitionErrorTimestamp

`GenApi::Integer& EventAcquisitionErrorTimestamp`

Description: Returns the Timestamp of the Acquisition Error [Event](#).

Visibility: Expert

10.10.4.256 EventAcquisitionStart

`GenApi::Integer& EventAcquisitionStart`

Description: Returns the unique Identifier of the Acquisition Start type of [Event](#).

Visibility: Expert

10.10.4.257 EventAcquisitionStartFrameID

`GenApi::Integer& EventAcquisitionStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Start [Event](#).

Visibility: Expert

10.10.4.258 EventAcquisitionStartTimestamp

`GenApi::Integer& EventAcquisitionStartTimestamp`

Description: Returns the Timestamp of the Acquisition Start [Event](#).

Visibility: Expert

10.10.4.259 EventAcquisitionTransferEnd

`GenApi::Integer& EventAcquisitionTransferEnd`

Description: Returns the unique Identifier of the Acquisition Transfer End type of [Event](#).

Visibility: Expert

10.10.4.260 EventAcquisitionTransferEndFrameID

`GenApi::Integer& EventAcquisitionTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer End [Event](#).

Visibility: Expert

10.10.4.261 EventAcquisitionTransferEndTimestamp

`GenApi::Integer& EventAcquisitionTransferEndTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer End [Event](#).

Visibility: Expert

10.10.4.262 EventAcquisitionTransferStart

`GenApi::Integer& EventAcquisitionTransferStart`

Description: Returns the unique Identifier of the Acquisition Transfer Start type of [Event](#).

Visibility: Expert

10.10.4.263 EventAcquisitionTransferStartFrameID

`GenApi::Integer& EventAcquisitionTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Transfer Start [Event](#).

Visibility: Expert

10.10.4.264 EventAcquisitionTransferStartTimestamp

`GenApi::Integer& EventAcquisitionTransferStartTimestamp`

Description: Returns the Timestamp of the Acquisition Transfer Start [Event](#).

Visibility: Expert

10.10.4.265 EventAcquisitionTrigger

`GenApi::Integer& EventAcquisitionTrigger`

Description: Returns the unique Identifier of the Acquisition Trigger type of [Event](#).

Visibility: Expert

10.10.4.266 EventAcquisitionTriggerFrameID

`GenApi::Integer& EventAcquisitionTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Acquisition Trigger [Event](#).

Visibility: Expert

10.10.4.267 EventAcquisitionTriggerTimestamp

`GenApi::Integer& EventAcquisitionTriggerTimestamp`

Description: Returns the Timestamp of the Acquisition Trigger [Event](#).

Visibility: Expert

10.10.4.268 EventActionLate

`GenApi::Integer& EventActionLate`

Description: Returns the unique Identifier of the Action Late type of [Event](#).

Visibility: Expert

10.10.4.269 EventActionLateFrameID

`GenApi::Integer& EventActionLateFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Action Late [Event](#).

Visibility: Expert

10.10.4.270 EventActionLateTimestamp

`GenApi::Integer& EventActionLateTimestamp`

Description: Returns the Timestamp of the Action Late [Event](#).

Visibility: Expert

10.10.4.271 EventCounter0End

`GenApi::Integer& EventCounter0End`

Description: Returns the unique Identifier of the Counter 0 End type of [Event](#).

Visibility: Expert

10.10.4.272 EventCounter0EndFrameID

`GenApi::Integer& EventCounter0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 End [Event](#).

Visibility: Expert

10.10.4.273 EventCounter0EndTimestamp

`GenApi::Integer& EventCounter0EndTimestamp`

Description: Returns the Timestamp of the Counter 0 End [Event](#).

Visibility: Expert

10.10.4.274 EventCounter0Start

`GenApi::Integer& EventCounter0Start`

Description: Returns the unique Identifier of the Counter 0 Start type of [Event](#).

Visibility: Expert

10.10.4.275 EventCounter0StartFrameID

`GenApi::Integer& EventCounter0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 0 Start [Event](#).

Visibility: Expert

10.10.4.276 EventCounter0StartTimestamp

`GenApi::Integer& EventCounter0StartTimestamp`

Description: Returns the Timestamp of the Counter 0 Start [Event](#).

Visibility: Expert

10.10.4.277 EventCounter1End

`GenApi::Integer& EventCounter1End`

Description: Returns the unique Identifier of the Counter 1 End type of [Event](#).

Visibility: Expert

10.10.4.278 EventCounter1EndFrameID

`GenApi::Integer& EventCounter1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 End [Event](#).

Visibility: Expert

10.10.4.279 EventCounter1EndTimestamp

`GenApi::Integer& EventCounter1EndTimestamp`

Description: Returns the Timestamp of the Counter 1 End [Event](#).

Visibility: Expert

10.10.4.280 EventCounter1Start

`GenApi::Integer& EventCounter1Start`

Description: Returns the unique Identifier of the Counter 1 Start type of [Event](#).

Visibility: Expert

10.10.4.281 EventCounter1StartFrameID

`GenApi::Integer& EventCounter1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Counter 1 Start [Event](#).

Visibility: Expert

10.10.4.282 EventCounter1StartTimestamp

`GenApi::Integer& EventCounter1StartTimestamp`

Description: Returns the Timestamp of the Counter 1 Start [Event](#).

Visibility: Expert

10.10.4.283 EventEncoder0Restarted

`GenApi::Integer& EventEncoder0Restarted`

Description: Returns the unique Identifier of the Encoder 0 Restarted type of [Event](#).

Visibility: Expert

10.10.4.284 EventEncoder0RestartedFrameID

`GenApi::Integer& EventEncoder0RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Restarted [Event](#).

Visibility: Expert

10.10.4.285 EventEncoder0RestartedTimestamp

`GenApi::Integer& EventEncoder0RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Restarted [Event](#).

Visibility: Expert

10.10.4.286 EventEncoder0Stopped

`GenApi::Integer& EventEncoder0Stopped`

Description: Returns the unique Identifier of the Encoder 0 Stopped type of [Event](#).

Visibility: Expert

10.10.4.287 EventEncoder0StoppedFrameID

`GenApi::Integer& EventEncoder0StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 0 Stopped [Event](#).

Visibility: Expert

10.10.4.288 EventEncoder0StoppedTimestamp

`GenApi::Integer& EventEncoder0StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 0 Stopped [Event](#).

Visibility: Expert

10.10.4.289 EventEncoder1Restarted

`GenApi::Integer& EventEncoder1Restarted`

Description: Returns the unique Identifier of the Encoder 1 Restarted type of [Event](#).

Visibility: Expert

10.10.4.290 EventEncoder1RestartedFrameID

`GenApi::Integer& EventEncoder1RestartedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Restarted [Event](#).

Visibility: Expert

10.10.4.291 EventEncoder1RestartedTimestamp

`GenApi::Integer& EventEncoder1RestartedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Restarted [Event](#).

Visibility: Expert

10.10.4.292 EventEncoder1Stopped

`GenApi::Integer& EventEncoder1Stopped`

Description: Returns the unique Identifier of the Encoder 1 Stopped type of [Event](#).

Visibility: Expert

10.10.4.293 EventEncoder1StoppedFrameID

`GenApi::Integer& EventEncoder1StoppedFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Encoder 1 Stopped [Event](#).

Visibility: Expert

10.10.4.294 EventEncoder1StoppedTimestamp

`GenApi::Integer& EventEncoder1StoppedTimestamp`

Description: Returns the Timestamp of the Encoder 1 Stopped [Event](#).

Visibility: Expert

10.10.4.295 EventError

`GenApi::Integer& EventError`

Description: Returns the unique identifier of the Error type of [Event](#).

Visibility:

10.10.4.296 EventErrorCode

`GenApi::Integer& EventErrorCode`

Description: Returns the error code for the error that happened Visibility:

10.10.4.297 EventErrorFrameID

`GenApi::Integer& EventErrorFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Error [Event](#).

Visibility:

10.10.4.298 EventErrorTimestamp

`GenApi::Integer& EventErrorTimestamp`

Description: Returns the Timestamp of the Error [Event](#).

Visibility:

10.10.4.299 EventExposureEnd

`GenApi::Integer& EventExposureEnd`

Description: Returns the unique identifier of the Exposure End type of [Event](#).

Visibility:

10.10.4.300 EventExposureEndFrameID

`GenApi::Integer& EventExposureEndFrameID`

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

Visibility:

10.10.4.301 EventExposureEndTimestamp

`GenApi::Integer& EventExposureEndTimestamp`

Description: Returns the Timestamp of the Exposure End [Event](#).

Visibility:

10.10.4.302 EventExposureStart

`GenApi::Integer& EventExposureStart`

Description: Returns the unique Identifier of the Exposure Start type of [Event](#).

Visibility: Expert

10.10.4.303 EventExposureStartFrameID

`GenApi::Integer& EventExposureStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Exposure Start [Event](#).

Visibility: Expert

10.10.4.304 EventExposureStartTimestamp

`GenApi::Integer& EventExposureStartTimestamp`

Description: Returns the Timestamp of the Exposure Start [Event](#).

Visibility: Expert

10.10.4.305 EventFrameBurstEnd

`GenApi::Integer& EventFrameBurstEnd`

Description: Returns the unique Identifier of the Frame Burst End type of [Event](#).

Visibility: Expert

10.10.4.306 EventFrameBurstEndFrameID

`GenApi::Integer& EventFrameBurstEndFrameID`

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

Visibility: Expert

10.10.4.307 EventFrameBurstEndTimestamp

`GenApi::Integer& EventFrameBurstEndTimestamp`

Description: Returns the Timestamp of the Frame Burst End [Event](#).

Visibility: Expert

10.10.4.308 EventFrameBurstStart

`GenApi::Integer& EventFrameBurstStart`

Description: Returns the unique Identifier of the Frame Burst Start type of [Event](#).

Visibility: Expert

10.10.4.309 EventFrameBurstStartFrameID

`GenApi::Integer& EventFrameBurstStartFrameID`

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

Visibility: Expert

10.10.4.310 EventFrameBurstStartTimestamp

`GenApi::Integer& EventFrameBurstStartTimestamp`

Description: Returns the Timestamp of the Frame Burst Start [Event](#).

Visibility: Expert

10.10.4.311 EventFrameEnd

`GenApi::Integer& EventFrameEnd`

Description: Returns the unique Identifier of the Frame End type of [Event](#).

Visibility: Expert

10.10.4.312 EventFrameEndFrameID

`GenApi::Integer& EventFrameEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame End [Event](#).

Visibility: Expert

10.10.4.313 EventFrameEndTimestamp

`GenApi::Integer& EventFrameEndTimestamp`

Description: Returns the Timestamp of the Frame End [Event](#).

Visibility: Expert

10.10.4.314 EventFrameStart

`GenApi::Integer& EventFrameStart`

Description: Returns the unique Identifier of the Frame Start type of [Event](#).

Visibility: Expert

10.10.4.315 EventFrameStartFrameID

`GenApi::Integer& EventFrameStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Start [Event](#).

Visibility: Expert

10.10.4.316 EventFrameStartTimestamp

`GenApi::Integer& EventFrameStartTimestamp`

Description: Returns the Timestamp of the Frame Start [Event](#).

Visibility: Expert

10.10.4.317 EventFrameTransferEnd

`GenApi::Integer& EventFrameTransferEnd`

Description: Returns the unique Identifier of the Frame Transfer End type of [Event](#).

Visibility: Expert

10.10.4.318 EventFrameTransferEndFrameID

`GenApi::Integer& EventFrameTransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer End [Event](#).

Visibility: Expert

10.10.4.319 EventFrameTransferEndTimestamp

`GenApi::Integer& EventFrameTransferEndTimestamp`

Description: Returns the Timestamp of the Frame Transfer End [Event](#).

Visibility: Expert

10.10.4.320 EventFrameTransferStart

`GenApi::Integer& EventFrameTransferStart`

Description: Returns the unique Identifier of the Frame Transfer Start type of [Event](#).

Visibility: Expert

10.10.4.321 EventFrameTransferStartFrameID

`GenApi::Integer& EventFrameTransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Frame Transfer Start [Event](#).

Visibility: Expert

10.10.4.322 EventFrameTransferStartTimestamp

`GenApi::Integer& EventFrameTransferStartTimestamp`

Description: Returns the Timestamp of the Frame Transfer Start [Event](#).

Visibility: Expert

10.10.4.323 EventFrameTrigger

`GenApi::Integer& EventFrameTrigger`

Description: Returns the unique Identifier of the FrameTrigger type of [Event](#).

It can be used to register a callback function to be notified of the event occurrence. Its value uniquely identifies the type event received. Visibility: Expert

10.10.4.324 EventFrameTriggerFrameID

`GenApi::Integer& EventFrameTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the FrameTrigger [Event](#).

Visibility: Expert

10.10.4.325 EventFrameTriggerTimestamp

`GenApi::Integer& EventFrameTriggerTimestamp`

Description: Returns the Timestamp of the FrameTrigger [Event](#).

It can be used to determine precisely when the event occurred. Visibility: Expert

10.10.4.326 EventLine0AnyEdge

`GenApi::Integer& EventLine0AnyEdge`

Description: Returns the unique Identifier of the Line 0 Any Edge type of [Event](#).

Visibility: Expert

10.10.4.327 EventLine0AnyEdgeFrameID

`GenApi::Integer& EventLine0AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Any Edge [Event](#).

Visibility: Expert

10.10.4.328 EventLine0AnyEdgeTimestamp

`GenApi::Integer& EventLine0AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Any Edge [Event](#).

Visibility: Expert

10.10.4.329 EventLine0FallingEdge

`GenApi::Integer& EventLine0FallingEdge`

Description: Returns the unique Identifier of the Line 0 Falling Edge type of [Event](#).

Visibility: Expert

10.10.4.330 EventLine0FallingEdgeFrameID

`GenApi::Integer& EventLine0FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Falling Edge [Event](#).

Visibility: Expert

10.10.4.331 EventLine0FallingEdgeTimestamp

`GenApi::Integer& EventLine0FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Falling Edge [Event](#).

Visibility: Expert

10.10.4.332 EventLine0RisingEdge

`GenApi::Integer& EventLine0RisingEdge`

Description: Returns the unique Identifier of the Line 0 Rising Edge type of [Event](#).

Visibility: Expert

10.10.4.333 EventLine0RisingEdgeFrameID

`GenApi::Integer& EventLine0RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 0 Rising Edge [Event](#).

Visibility: Expert

10.10.4.334 EventLine0RisingEdgeTimestamp

`GenApi::Integer& EventLine0RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 0 Rising Edge [Event](#).

Visibility: Expert

10.10.4.335 EventLine1AnyEdge

`GenApi::Integer& EventLine1AnyEdge`

Description: Returns the unique Identifier of the Line 1 Any Edge type of [Event](#).

Visibility: Expert

10.10.4.336 EventLine1AnyEdgeFrameID

`GenApi::Integer& EventLine1AnyEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Any Edge [Event](#).

Visibility: Expert

10.10.4.337 EventLine1AnyEdgeTimestamp

`GenApi::Integer& EventLine1AnyEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Any Edge [Event](#).

Visibility: Expert

10.10.4.338 EventLine1FallingEdge

`GenApi::Integer& EventLine1FallingEdge`

Description: Returns the unique Identifier of the Line 1 Falling Edge type of [Event](#).

Visibility: Expert

10.10.4.339 EventLine1FallingEdgeFrameID

`GenApi::Integer& EventLine1FallingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Falling Edge [Event](#).

Visibility: Expert

10.10.4.340 EventLine1FallingEdgeTimestamp

`GenApi::Integer& EventLine1FallingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Falling Edge [Event](#).

Visibility: Expert

10.10.4.341 EventLine1RisingEdge

`GenApi::Integer& EventLine1RisingEdge`

Description: Returns the unique Identifier of the Line 1 Rising Edge type of [Event](#).

Visibility: Expert

10.10.4.342 EventLine1RisingEdgeFrameID

`GenApi::Integer& EventLine1RisingEdgeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Line 1 Rising Edge [Event](#).

Visibility: Expert

10.10.4.343 EventLine1RisingEdgeTimestamp

`GenApi::Integer& EventLine1RisingEdgeTimestamp`

Description: Returns the Timestamp of the Line 1 Rising Edge [Event](#).

Visibility: Expert

10.10.4.344 EventLinkSpeedChange

`GenApi::Integer& EventLinkSpeedChange`

Description: Returns the unique Identifier of the Link Speed Change type of [Event](#).

Visibility: Expert

10.10.4.345 EventLinkSpeedChangeFrameID

`GenApi::Integer& EventLinkSpeedChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Speed Change [Event](#).

Visibility: Expert

10.10.4.346 EventLinkSpeedChangeTimestamp

`GenApi::Integer& EventLinkSpeedChangeTimestamp`

Description: Returns the Timestamp of the Link Speed Change [Event](#).

Visibility: Expert

10.10.4.347 EventLinkTrigger0

`GenApi::Integer& EventLinkTrigger0`

Description: Returns the unique Identifier of the Link Trigger 0 type of [Event](#).

Visibility: Expert

10.10.4.348 EventLinkTrigger0FrameID

`GenApi::Integer& EventLinkTrigger0FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 0 [Event](#).

Visibility: Expert

10.10.4.349 EventLinkTrigger0Timestamp

`GenApi::Integer& EventLinkTrigger0Timestamp`

Description: Returns the Timestamp of the Link Trigger 0 [Event](#).

Visibility: Expert

10.10.4.350 EventLinkTrigger1

`GenApi::Integer& EventLinkTrigger1`

Description: Returns the unique Identifier of the Link Trigger 1 type of [Event](#).

Visibility: Expert

10.10.4.351 EventLinkTrigger1FrameID

`GenApi::Integer& EventLinkTrigger1FrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Link Trigger 1 [Event](#).

Visibility: Expert

10.10.4.352 EventLinkTrigger1Timestamp

`GenApi::Integer& EventLinkTrigger1Timestamp`

Description: Returns the Timestamp of the Link Trigger 1 [Event](#).

Visibility: Expert

10.10.4.353 EventNotification

`GenApi::EnumerationT<EventNotificationEnums>& EventNotification`

Description: Enables/Disables the selected event.

Visibility:

10.10.4.354 EventSelector

`GenApi::EnumerationT<EventSelectorEnums>& EventSelector`

Description: Selects which [Event](#) to enable or disable.

Visibility:

10.10.4.355 EventSequencerSetChange

`GenApi::Integer& EventSequencerSetChange`

Description: Returns the unique Identifier of the Sequencer Set Change type of [Event](#).

Visibility: Expert

10.10.4.356 EventSequencerSetChangeFrameID

`GenApi::Integer& EventSequencerSetChangeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Sequencer Set Change [Event](#).

Visibility: Expert

10.10.4.357 EventSequencerSetChangeTimestamp

`GenApi::Integer& EventSequencerSetChangeTimestamp`

Description: Returns the Timestamp of the Sequencer Set Change [Event](#).

Visibility: Expert

10.10.4.358 EventSerialData

`GenApi::IString& EventSerialData`

Description: Returns the serial data that was received.

Visibility:

10.10.4.359 EventSerialDataLength

`GenApi::Integer& EventSerialDataLength`

Description: Returns the length of the received serial data that was included in the event payload.

Visibility:

10.10.4.360 EventSerialPortReceive

`GenApi::Integer& EventSerialPortReceive`

Description: Returns the unique identifier of the Serial Port Receive type of [Event](#).

Visibility:

10.10.4.361 EventSerialPortReceiveTimestamp

`GenApi::Integer& EventSerialPortReceiveTimestamp`

Description: Returns the Timestamp of the Serial Port Receive [Event](#).

Visibility:

10.10.4.362 EventSerialReceiveOverflow

`GenApi::IBoolean& EventSerialReceiveOverflow`

Description: Returns the status of the event serial receive overflow.

Visibility:

10.10.4.363 EventStream0TransferBlockEnd

`GenApi::IInteger& EventStream0TransferBlockEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Block End type of [Event](#).

Visibility: Expert

10.10.4.364 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

10.10.4.365 EventStream0TransferBlockEndTimestamp

`GenApi::IInteger& EventStream0TransferBlockEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block End [Event](#).

Visibility: Expert

10.10.4.366 EventStream0TransferBlockStart

`GenApi::IInteger& EventStream0TransferBlockStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Start type of [Event](#).

Visibility: Expert

10.10.4.367 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

10.10.4.368 EventStream0TransferBlockStartTimestamp

`GenApi::Integer& EventStream0TransferBlockStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Start [Event](#).

Visibility: Expert

10.10.4.369 EventStream0TransferBlockTrigger

`GenApi::Integer& EventStream0TransferBlockTrigger`

Description: Returns the unique Identifier of the Stream 0 Transfer Block Trigger type of [Event](#).

Visibility: Expert

10.10.4.370 EventStream0TransferBlockTriggerFrameID

`GenApi::Integer& EventStream0TransferBlockTriggerFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

10.10.4.371 EventStream0TransferBlockTriggerTimestamp

`GenApi::Integer& EventStream0TransferBlockTriggerTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Block Trigger [Event](#).

Visibility: Expert

10.10.4.372 EventStream0TransferBurstEnd

`GenApi::Integer& EventStream0TransferBurstEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst End type of [Event](#).

Visibility: Expert

10.10.4.373 EventStream0TransferBurstEndFrameID

`GenApi::Integer& EventStream0TransferBurstEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

10.10.4.374 EventStream0TransferBurstEndTimestamp

`GenApi::Integer& EventStream0TransferBurstEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst End [Event](#).

Visibility: Expert

10.10.4.375 EventStream0TransferBurstStart

`GenApi::Integer& EventStream0TransferBurstStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Burst Start type of [Event](#).

Visibility: Expert

10.10.4.376 EventStream0TransferBurstStartFrameID

`GenApi::Integer& EventStream0TransferBurstStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

10.10.4.377 EventStream0TransferBurstStartTimestamp

`GenApi::Integer& EventStream0TransferBurstStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Burst Start [Event](#).

Visibility: Expert

10.10.4.378 EventStream0TransferEnd

`GenApi::Integer& EventStream0TransferEnd`

Description: Returns the unique Identifier of the Stream 0 Transfer End type of [Event](#).

Visibility: Expert

10.10.4.379 EventStream0TransferEndFrameID

`GenApi::Integer& EventStream0TransferEndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer End [Event](#).

Visibility: Expert

10.10.4.380 EventStream0TransferEndTimestamp

`GenApi::Integer& EventStream0TransferEndTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer End [Event](#).

Visibility: Expert

10.10.4.381 EventStream0TransferOverflow

`GenApi::Integer& EventStream0TransferOverflow`

Description: Returns the unique Identifier of the Stream 0 Transfer Overflow type of [Event](#).

Visibility: Expert

10.10.4.382 EventStream0TransferOverflowFrameID

`GenApi::Integer& EventStream0TransferOverflowFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

10.10.4.383 EventStream0TransferOverflowTimestamp

`GenApi::Integer& EventStream0TransferOverflowTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Overflow [Event](#).

Visibility: Expert

10.10.4.384 EventStream0TransferPause

`GenApi::Integer& EventStream0TransferPause`

Description: Returns the unique Identifier of the Stream 0 Transfer Pause type of [Event](#).

Visibility: Expert

10.10.4.385 EventStream0TransferPauseFrameID

`GenApi::Integer& EventStream0TransferPauseFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

10.10.4.386 EventStream0TransferPauseTimestamp

`GenApi::Integer& EventStream0TransferPauseTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Pause [Event](#).

Visibility: Expert

10.10.4.387 EventStream0TransferResume

`GenApi::Integer& EventStream0TransferResume`

Description: Returns the unique Identifier of the Stream 0 Transfer Resume type of [Event](#).

Visibility: Expert

10.10.4.388 EventStream0TransferResumeFrameID

`GenApi::Integer& EventStream0TransferResumeFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

10.10.4.389 EventStream0TransferResumeTimestamp

`GenApi::Integer& EventStream0TransferResumeTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Resume [Event](#).

Visibility: Expert

10.10.4.390 EventStream0TransferStart

`GenApi::Integer& EventStream0TransferStart`

Description: Returns the unique Identifier of the Stream 0 Transfer Start type of [Event](#).

Visibility: Expert

10.10.4.391 EventStream0TransferStartFrameID

`GenApi::Integer& EventStream0TransferStartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Stream 0 Transfer Start [Event](#).

Visibility: Expert

10.10.4.392 EventStream0TransferStartTimestamp

`GenApi::Integer& EventStream0TransferStartTimestamp`

Description: Returns the Timestamp of the Stream 0 Transfer Start [Event](#).

Visibility: Expert

10.10.4.393 EventTest

`GenApi::Integer& EventTest`

Description: Returns the unique identifier of the Test type of [Event](#).

Visibility:

10.10.4.394 EventTestTimestamp

`GenApi::Integer& EventTestTimestamp`

Description: Returns the Timestamp of the Test [Event](#).

Visibility:

10.10.4.395 EventTimer0End

`GenApi::Integer& EventTimer0End`

Description: Returns the unique Identifier of the Timer 0 End type of [Event](#).

Visibility: Expert

10.10.4.396 EventTimer0EndFrameID

`GenApi::Integer& EventTimer0EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 End [Event](#).

Visibility: Expert

10.10.4.397 EventTimer0EndTimestamp

`GenApi::Integer& EventTimer0EndTimestamp`

Description: Returns the Timestamp of the Timer 0 End [Event](#).

Visibility: Expert

10.10.4.398 EventTimer0Start

`GenApi::Integer& EventTimer0Start`

Description: Returns the unique Identifier of the Timer 0 Start type of [Event](#).

Visibility: Expert

10.10.4.399 EventTimer0StartFrameID

`GenApi::Integer& EventTimer0StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 0 Start [Event](#).

Visibility: Expert

10.10.4.400 EventTimer0StartTimestamp

`GenApi::Integer& EventTimer0StartTimestamp`

Description: Returns the Timestamp of the Timer 0 Start [Event](#).

Visibility: Expert

10.10.4.401 EventTimer1End

`GenApi::Integer& EventTimer1End`

Description: Returns the unique Identifier of the Timer 1 End type of [Event](#).

Visibility: Expert

10.10.4.402 EventTimer1EndFrameID

`GenApi::Integer& EventTimer1EndFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 End [Event](#).

Visibility: Expert

10.10.4.403 EventTimer1EndTimestamp

`GenApi::Integer& EventTimer1EndTimestamp`

Description: Returns the Timestamp of the Timer 1 End [Event](#).

Visibility: Expert

10.10.4.404 EventTimer1Start

`GenApi::Integer& EventTimer1Start`

Description: Returns the unique Identifier of the Timer 1 Start type of [Event](#).

Visibility: Expert

10.10.4.405 EventTimer1StartFrameID

`GenApi::Integer& EventTimer1StartFrameID`

Description: Returns the unique Identifier of the Frame (or image) that generated the Timer 1 Start [Event](#).

Visibility: Expert

10.10.4.406 EventTimer1StartTimestamp

`GenApi::Integer& EventTimer1StartTimestamp`

Description: Returns the Timestamp of the Timer 1 Start [Event](#).

Visibility: Expert

10.10.4.407 ExposureActiveMode

`GenApi::EnumerationT<ExposureActiveModeEnums>& ExposureActiveMode`

Description: Control sensor active exposure mode.

Visibility:

10.10.4.408 ExposureAuto

`GenApi::EnumerationT<ExposureAutoEnums>& ExposureAuto`

Description: Sets the automatic exposure mode Visibility:

10.10.4.409 ExposureMode

`GenApi::EnumerationT<ExposureModeEnums>& ExposureMode`

Description:

Sets the operation mode of the Exposure.

Visibility:

10.10.4.410 ExposureTime

`GenApi::IFloat& ExposureTime`

Description:

Exposure time in microseconds when Exposure Mode is Timed.

Visibility:

10.10.4.411 ExposureTimeMode

`GenApi::IEnumerationT<ExposureTimeModeEnums>& ExposureTimeMode`

Description: Sets the configuration mode of the ExposureTime feature.

Visibility: Beginner

10.10.4.412 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

10.10.4.413 FactoryReset

`GenApi::ICommand& FactoryReset`

Description: Returns all user tables to factory default Visibility:

10.10.4.414 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

10.10.4.415 FileAccessLength

`GenApi::IInteger& FileAccessLength`

Description: Controls the Length of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

10.10.4.416 FileAccessOffset

`GenApi::Integer& FileAccessOffset`

Description: Controls the Offset of the mapping between the device file storage and the FileAccessBuffer.

Visibility:

10.10.4.417 FileOpenMode

`GenApi::EnumerationT<FileOpenModeEnums>& FileOpenMode`

Description:

The mode of the file when it is opened.

The file can be opened for reading, writing or both. This must be set before opening the file.

Visibility:

10.10.4.418 FileOperationExecute

`GenApi::Command& FileOperationExecute`

Description:

This is a command that executes the selected file operation on the selected file.

Visibility:

10.10.4.419 FileOperationResult

`GenApi::Integer& FileOperationResult`

Description: Represents the file operation result.

For Read or Write operations, the number of successfully read/written bytes is returned. Visibility:

10.10.4.420 FileOperationSelector

`GenApi::EnumerationT<FileOperationSelectorEnums>& FileOperationSelector`

Description:

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

Visibility:

10.10.4.421 FileOperationStatus

`GenApi::IEnumerationT<FileOperationStatusEnums>& FileOperationStatus`

Description: Represents the file operation execution status.

Visibility:

10.10.4.422 FileSelector

`GenApi::IEnumerationT<FileSelectorEnums>& FileSelector`

Description:

Selects which file is being operated on.

This must be set before performing any file operations.

Visibility:

10.10.4.423 FileSize

`GenApi::IInteger& FileSize`

Description: Represents the size of the selected file in bytes.

Visibility:

10.10.4.424 Gain

`GenApi::IFloat& Gain`

Description:

Controls the amplification of the video signal in dB.

Visibility:

10.10.4.425 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:

10.10.4.426 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

10.10.4.427 GainSelector

`GenApi::IEnumerationT<GainSelectorEnums>& GainSelector`

Description: Selects which gain to control.

The All selection is a total amplification across all channels (or taps).

Visibility:

10.10.4.428 Gamma

`GenApi::IFloat& Gamma`

Description: Controls the gamma correction of pixel intensity.

Visibility:

10.10.4.429 GammaEnable

`GenApi::IBoolean& GammaEnable`

Description: Enables/disables gamma correction.

Visibility:

10.10.4.430 GevActiveLinkCount

`GenApi::IInteger& GevActiveLinkCount`

Description: Indicates the current number of active logical links.

Visibility: Expert

10.10.4.431 GevCCP

`GenApi::IEnumerationT<GevCCPEnums>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility:

10.10.4.432 GevCurrentDefaultGateway

`GenApi::Integer& GevCurrentDefaultGateway`

Description: Reports the default gateway IP address to be used on the given logical link.

Visibility:

10.10.4.433 GevCurrentIPAddress

`GenApi::Integer& GevCurrentIPAddress`

Description: Reports the IP address for the given logical link.

Visibility:

10.10.4.434 GevCurrentIPConfigurationDHCP

`GenApi::Boolean& GevCurrentIPConfigurationDHCP`

Description: Controls whether the DHCP IP configuration scheme is activated on the given logical link.

Visibility:

10.10.4.435 GevCurrentIPConfigurationLLA

`GenApi::Boolean& GevCurrentIPConfigurationLLA`

Description: Controls whether the Link Local Address IP configuration scheme is activated on the given logical link.

Visibility:

10.10.4.436 GevCurrentIPConfigurationPersistentIP

`GenApi::Boolean& GevCurrentIPConfigurationPersistentIP`

Description: Controls whether the PersistentIP configuration scheme is activated on the given logical link.

Visibility:

10.10.4.437 GevCurrentPhysicalLinkConfiguration

`GenApi::EnumerationT<GevCurrentPhysicalLinkConfigurationEnums>& GevCurrentPhysicalLinkConfiguration`

Description: Indicates the current physical link configuration of the device.

Visibility: Expert

10.10.4.438 GevCurrentSubnetMask

`GenApi::Integer& GevCurrentSubnetMask`

Description: Reports the subnet mask of the given logical link.

Visibility:

10.10.4.439 GevDiscoveryAckDelay

`GenApi::Integer& GevDiscoveryAckDelay`

Description: Indicates the maximum randomized delay the device will wait to acknowledge a discovery command.

Visibility: Expert

10.10.4.440 GevFirstURL

`GenApi::String& GevFirstURL`

Description: The first choice of URL for the XML device description file.

Visibility:

10.10.4.441 GevGVCPExtendedStatusCodes

`GenApi::Boolean& GevGVCPExtendedStatusCodes`

Description: Enables the generation of extended status codes.

Visibility: Guru

10.10.4.442 GevGVCPExtendedStatusCodesSelector

`GenApi::EnumerationT<GevGVCPExtendedStatusCodesSelectorEnums>& GevGVCPExtendedStatusCodesSelector`

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

Visibility: Guru

10.10.4.443 GevGVCPHeartbeatDisable

`GenApi::Boolean& GevGVCPHeartbeatDisable`

Description: Disables the GVCP heartbeat.

Visibility:

10.10.4.444 GevGVCPPendingAck

`GenApi::IBoolean& GevGVCPPendingAck`

Description: Enables the generation of PENDING_ACK.

Visibility:

10.10.4.445 GevGVCPPendingTimeout

`GenApi::IInteger& GevGVCPPendingTimeout`

Description: Indicates the longest GVCP command execution time before the device returns a PENDING_ACK in milliseconds.

Visibility:

10.10.4.446 GevGVSPExtendedIDMode

`GenApi::IEnumerationT<GevGVSPExtendedIDModeEnums>& GevGVSPExtendedIDMode`

Description: Enables the extended IDs mode.

Visibility: Expert

10.10.4.447 GevHeartbeatTimeout

`GenApi::IInteger& GevHeartbeatTimeout`

Description: Indicates the current heartbeat timeout in milliseconds.

Visibility:

10.10.4.448 GevIEEE1588

`GenApi::IBoolean& GevIEEE1588`

Description: Enables the IEEE 1588 Precision Time Protocol to control the timestamp register.

Visibility:

10.10.4.449 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:

10.10.4.450 GevIEEE1588Mode

`GenApi::IEnumerationT<GevIEEE1588ModeEnums>& GevIEEE1588Mode`

Description: Provides the mode of the IEEE 1588 clock.

Visibility:

10.10.4.451 GevIEEE1588Status

`GenApi::IEnumerationT<GevIEEE1588StatusEnums>& GevIEEE1588Status`

Description: Provides the status of the IEEE 1588 clock.

Visibility:

10.10.4.452 GevInterfaceSelector

`GenApi::IInteger& GevInterfaceSelector`

Description: Selects which logical link to control.

Visibility:

10.10.4.453 GevIPConfigurationStatus

`GenApi::IEnumerationT<GevIPConfigurationStatusEnums>& GevIPConfigurationStatus`

Description: Reports the current IP configuration status.

Visibility: Beginner

10.10.4.454 GevMACAddress

`GenApi::IInteger& GevMACAddress`

Description: MAC address of the logical link.

Visibility:

10.10.4.455 GevMCDA

`GenApi::IInteger& GevMCDA`

Description: Controls the destination IP address of the message channel Visibility:

10.10.4.456 GevMCPHostPort

`GenApi::Integer& GevMCPHostPort`

Description: The port to which the device must send messages Visibility:

10.10.4.457 GevMCRC

`GenApi::Integer& GevMCRC`

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

Visibility:

10.10.4.458 GevMCSP

`GenApi::Integer& GevMCSP`

Description: Indicates the source port of the message channel.

Visibility:

10.10.4.459 GevMCTT

`GenApi::Integer& GevMCTT`

Description: Indicates the transmission timeout of the message channel.

Visibility:

10.10.4.460 GevNumberOfInterfaces

`GenApi::Integer& GevNumberOfInterfaces`

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

Visibility:

10.10.4.461 GevPAUSEFrameReception

`GenApi::Boolean& GevPAUSEFrameReception`

Description: Controls whether incoming PAUSE Frames are handled on the given logical link.

Visibility: Expert

10.10.4.462 GevPAUSEFrameTransmission

`GenApi::IBoolean& GevPAUSEFrameTransmission`

Description: Controls whether PAUSE Frames can be generated on the given logical link.

Visibility: Expert

10.10.4.463 GevPersistentDefaultGateway

`GenApi::IInteger& GevPersistentDefaultGateway`

Description: Controls the persistent default gateway for this logical link.

Visibility:

10.10.4.464 GevPersistentIPAddress

`GenApi::IInteger& GevPersistentIPAddress`

Description: Controls the Persistent IP address for this logical link.

Visibility:

10.10.4.465 GevPersistentSubnetMask

`GenApi::IInteger& GevPersistentSubnetMask`

Description: Controls the Persistent subnet mask associated with the Persistent IP address on this logical link.

Visibility:

10.10.4.466 GevPhysicalLinkConfiguration

`GenApi::IEnumerationT<GevPhysicalLinkConfigurationEnums>& GevPhysicalLinkConfiguration`

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

Visibility: Expert

10.10.4.467 GevPrimaryApplicationIPAddress

`GenApi::IInteger& GevPrimaryApplicationIPAddress`

Description: Returns the address of the primary application.

Visibility: Guru

10.10.4.468 GevPrimaryApplicationSocket

`GenApi::Integer& GevPrimaryApplicationSocket`

Description: Returns the UDP source port of the primary application.

Visibility: Guru

10.10.4.469 GevPrimaryApplicationSwitchoverKey

`GenApi::Integer& GevPrimaryApplicationSwitchoverKey`

Description: Controls the key to use to authenticate primary application switchover requests.

Visibility: Guru

10.10.4.470 GevSCCFGAllInTransmission

`GenApi::Boolean& GevSCCFGAllInTransmission`

Description: Enables the selected GVSP transmitter to use the single packet per data block All-in Transmission mode.

Visibility: Guru

10.10.4.471 GevSCCFGExtendedChunkData

`GenApi::Boolean& GevSCCFGExtendedChunkData`

Description: Enables cameras to use the extended chunk data payload type for this stream channel.

Visibility:

10.10.4.472 GevSCCFGPacketResendDestination

`GenApi::Boolean& GevSCCFGPacketResendDestination`

Description: Enables the alternate IP destination for stream packets resent due to a packet resend request.

When True, the source IP address provided in the packet resend command packet is used. When False, the value set in the `GevSCDA[GevStreamChannelSelector]` feature is used. Visibility: Guru

10.10.4.473 GevSCCFGUnconditionalStreaming

`GenApi::Boolean& GevSCCFGUnconditionalStreaming`

Description: Enables the camera to continue to stream, for this stream channel, if its control channel is closed or regardless of the reception of any ICMP messages (such as destination unreachable messages).

Visibility:

10.10.4.474 GevSCDA

`GenApi::Integer & GevSCDA`

Description: Controls the destination IP address of the selected stream channel to which a GVSP transmitter must send data stream or the destination IP address from which a GVSP receiver may receive data stream.

Visibility:

10.10.4.475 GevSCPD

`GenApi::Integer & GevSCPD`

Description: Controls the delay (in GEV timestamp counter unit) to insert between each packet for this stream channel.

This can be used as a crude flow-control mechanism if the application or the network infrastructure cannot keep up with the packets coming from the device. Visibility:

10.10.4.476 GevSCPDirection

`GenApi::Integer & GevSCPDirection`

Description: Transmit or Receive of the channel Visibility:

10.10.4.477 GevSCPHostPort

`GenApi::Integer & GevSCPHostPort`

Description: Controls the port of the selected channel to which a GVSP transmitter must send data stream or the port from which a GVSP receiver may receive data stream.

Visibility:

10.10.4.478 GevSCPInterfaceIndex

`GenApi::Integer & GevSCPInterfaceIndex`

Description: Index of the logical link to use.

Visibility:

10.10.4.479 GevSCPSBigEndian

`GenApi::Boolean & GevSCPSBigEndian`

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

Visibility:

10.10.4.480 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:

10.10.4.481 GevSCPSFireTestPacket

`GenApi::IBoolean& GevSCPSFireTestPacket`

Description: Sends a test packet.

Visibility:

10.10.4.482 GevSCSPacketSize

`GenApi::IInteger& GevSCSPacketSize`

Description: Specifies the stream packet size (in bytes) to send on this channel.

Visibility:

10.10.4.483 GevSCSP

`GenApi::IInteger& GevSCSP`

Description: Indicates the source port of the stream channel.

Visibility:

10.10.4.484 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

10.10.4.485 GevSCZoneCount

`GenApi::IInteger& GevSCZoneCount`

Description: Reports the number of zones per block transmitted on the selected stream channel.

Visibility: Guru

10.10.4.486 GevSCZoneDirectionAll

`GenApi::Integer& GevSCZoneDirectionAll`

Description: Reports the transmission direction of each zone transmitted on the selected stream channel.

Visibility: Guru

10.10.4.487 GevSecondURL

`GenApi::String& GevSecondURL`

Description: The second choice of URL to the XML device description file.

Visibility:

10.10.4.488 GevStreamChannelSelector

`GenApi::Integer& GevStreamChannelSelector`

Description: Selects the stream channel to control.

Visibility:

10.10.4.489 GevSupportedOption

`GenApi::Boolean& GevSupportedOption`

Description: Returns if the selected GEV option is supported.

Visibility:

10.10.4.490 GevSupportedOptionSelector

`GenApi::EnumerationT<GevSupportedOptionSelectorEnums>& GevSupportedOptionSelector`

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

Visibility:

10.10.4.491 GevTimestampTickFrequency

`GenApi::Integer& GevTimestampTickFrequency`

Description: Indicates the number of timestamp ticks in 1 second (frequency in Hz).

Visibility:

10.10.4.492 GuiXmlManifestAddress

`GenApi::Integer& GuiXmlManifestAddress`

Description: Location of the GUI XML manifest table.

Visibility:

10.10.4.493 Height

`GenApi::Integer& Height`

Description:

Height of the image provided by the device (in pixels).

Visibility:

10.10.4.494 HeightMax

`GenApi::Integer& HeightMax`

Description: Maximum height of the image (in pixels).

This dimension is calculated after vertical binning. HeightMax does not take into account the current Region of interest (Height or OffsetY). Visibility:

10.10.4.495 ImageComponentEnable

`GenApi::Boolean& ImageComponentEnable`

Description: Controls if the selected component streaming is active.

Visibility: Beginner

10.10.4.496 ImageComponentSelector

`GenApi::EnumerationT<ImageComponentSelectorEnums>& ImageComponentSelector`

Description: Selects a component to activate data streaming from.

Visibility: Beginner

10.10.4.497 ImageCompressionBitrate

`GenApi::Float& ImageCompressionBitrate`

Description: Control the rate of the produced compressed stream.

Visibility: Expert

10.10.4.498 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

10.10.4.499 ImageCompressionMode

```
GenApi::IEnumerationT<ImageCompressionModeEnums>& ImageCompressionMode
```

Description: Visibility:

10.10.4.500 ImageCompressionQuality

```
GenApi::IInteger& ImageCompressionQuality
```

Description: Control the quality of the produced compressed stream.

Visibility: Expert

10.10.4.501 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

10.10.4.502 IspEnable

```
GenApi::IBoolean& IspEnable
```

Description:

Controls whether the image processing core is used for optional pixel format mode (i.e. mono).

Visibility:

10.10.4.503 LineFilterWidth

`GenApi::IFloat& LineFilterWidth`

Description: Filter width in microseconds for the selected line and filter combination Visibility:

10.10.4.504 LineFormat

`GenApi::IEnumerationT<LineFormatEnums>& LineFormat`

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

Visibility:

10.10.4.505 LineInputFilterSelector

`GenApi::IEnumerationT<LineInputFilterSelectorEnums>& LineInputFilterSelector`

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

Visibility:

10.10.4.506 LineInverter

`GenApi::IBoolean& LineInverter`

Description: Controls the inversion of the signal of the selected input or output line.

Visibility:

10.10.4.507 LineMode

`GenApi::IEnumerationT<LineModeEnums>& LineMode`

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

Visibility:

10.10.4.508 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

10.10.4.509 LineSelector

`GenApi::IEnumerationT<LineSelectorEnums>& LineSelector`

Description: Selects the physical line (or pin) of the external device connector to configure Visibility:

10.10.4.510 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:

10.10.4.511 LineStatus

`GenApi::IBoolean& LineStatus`

Description: Returns the current status of the selected input or output Line Visibility:

10.10.4.512 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:

10.10.4.513 LinkErrorCount

`GenApi::IInteger& LinkErrorCount`

Description: Counts the number of error on the link.

Visibility:

10.10.4.514 LinkUptime

`GenApi::IInteger& LinkUptime`

Description: Time since the last phy negotiation (enumeration).

Visibility:

10.10.4.515 LogicBlockLUTInputActivation

`GenApi::IEnumerationT<LogicBlockLUTInputActivationEnums>& LogicBlockLUTInputActivation`

Description: Selects the activation mode of the Logic Input Source signal.

Visibility:

10.10.4.516 LogicBlockLUTInputSelector

`GenApi::IEnumerationT<LogicBlockLUTInputSelectorEnums>& LogicBlockLUTInputSelector`

Description: Controls which LogicBlockLUT Input Source & Activation to access.

Visibility:

10.10.4.517 LogicBlockLUTInputSource

`GenApi::IEnumerationT<LogicBlockLUTInputSourceEnums>& LogicBlockLUTInputSource`

Description: Selects the source for the input into the Logic LUT.

Visibility:

10.10.4.518 LogicBlockLUTOutputValue

`GenApi::IBoolean& LogicBlockLUTOutputValue`

Description: Controls the output column of the truth table for the selected LogicBlockLUTRowIndex.

Visibility:

10.10.4.519 LogicBlockLUTOutputValueAll

`GenApi::IInteger& LogicBlockLUTOutputValueAll`

Description: Sets the value of all the output bits in the selected LUT.

Visibility:

10.10.4.520 LogicBlockLUTRowIndex

`GenApi::IInteger& LogicBlockLUTRowIndex`

Description: Controls the row of the truth table to access in the selected LUT.

Visibility:

10.10.4.521 LogicBlockLUTSelector

`GenApi::IEnumerationT<LogicBlockLUTSelectorEnums>& LogicBlockLUTSelector`

Description: Selects which LogicBlock LUT to configure Visibility:

10.10.4.522 LogicBlockSelector

`GenApi::IEnumerationT<LogicBlockSelectorEnums>& LogicBlockSelector`

Description: Selects which LogicBlock to configure Visibility:

10.10.4.523 LUTEnable

`GenApi::IBoolean& LUTEnable`

Description:

Activates the selected LUT.

Visibility:

10.10.4.524 LUTIndex

`GenApi::IInteger& LUTIndex`

Description:

Control the index (offset) of the coefficient to access in the selected LUT.

Visibility:

10.10.4.525 LUTSelector

`GenApi::IEnumerationT<LUTSelectorEnums>& LUTSelector`

Description:

Selects which LUT to control.

Visibility:

10.10.4.526 LUTValue

`GenApi::Integer& LUTValue`

Description:

Returns the Value at entry LUTIndex of the LUT selected by LUTSelector.

Visibility:

10.10.4.527 LUTValueAll

`GenApi::IRegister& LUTValueAll`

Description: Accesses all the LUT coefficients in a single access without using individual LUTIndex.

Visibility: Guru

10.10.4.528 MaxDeviceResetTime

`GenApi::Integer& MaxDeviceResetTime`

Description: Time to wait until device reset complete (ms).

Visibility:

10.10.4.529 OffsetX

`GenApi::Integer& OffsetX`

Description:

Horizontal offset from the origin to the ROI (in pixels).

Visibility:

10.10.4.530 OffsetY

`GenApi::Integer& OffsetY`

Description:

Vertical offset from the origin to the ROI (in pixels).

Visibility:

10.10.4.531 PacketResendRequestCount

`GenApi::Integer& PacketResendRequestCount`

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

Visibility:

10.10.4.532 PayloadSize

`GenApi::Integer& PayloadSize`

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

Visibility:

10.10.4.533 PixelColorFilter

`GenApi::EnumerationT<PixelColorFilterEnums>& PixelColorFilter`

Description: Type of color filter that is applied to the image.

Only applies to Bayer pixel formats. All others have no color filter.

Visibility:

10.10.4.534 PixelDynamicRangeMax

`GenApi::Integer& PixelDynamicRangeMax`

Description: Maximum value that can be returned during the digitization process.

This corresponds to the brightest value of the camera. For color cameras, this returns the biggest value that each color component can take.

Visibility:

10.10.4.535 PixelDynamicRangeMin

`GenApi::Integer& PixelDynamicRangeMin`

Description: Minimum value that can be returned during the digitization process.

This corresponds to the darkest value of the camera. For color cameras, this returns the smallest value that each color component can take.

Visibility:

10.10.4.536 PixelFormat

`GenApi::IEnumerationT<PixelFormatEnums>& PixelFormat`

Description: Format of the pixel provided by the camera.

Visibility:

10.10.4.537 PixelFormatInfoID

`GenApi::IInteger& PixelFormatInfoID`

Description: Returns the value used by the streaming channels to identify the selected pixel format.

Visibility: Guru

10.10.4.538 PixelFormatInfoSelector

`GenApi::IEnumerationT<PixelFormatInfoSelectorEnums>& PixelFormatInfoSelector`

Description: Select the pixel format for which the information will be returned.

Visibility: Guru

10.10.4.539 PixelSize

`GenApi::IEnumerationT<PixelSizeEnums>& PixelSize`

Description: Total size in bits of a pixel of the image.

Visibility:

10.10.4.540 PowerSupplyCurrent

`GenApi::IFloat& PowerSupplyCurrent`

Description:

Indicates the output current of the selected power supply (A).

Visibility:

10.10.4.541 PowerSupplyVoltage

`GenApi::IFloat& PowerSupplyVoltage`

Description:

Indicates the current voltage of the selected power supply (V).

Visibility:

10.10.4.542 RegionDestination

`GenApi::IEnumerationT<RegionDestinationEnums>& RegionDestination`

Description: Control the destination of the selected region.

Visibility: Expert

10.10.4.543 RegionMode

`GenApi::IEnumerationT<RegionModeEnums>& RegionMode`

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

Visibility: Beginner

10.10.4.544 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

10.10.4.545 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:

10.10.4.546 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:

10.10.4.547 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:

10.10.4.548 Saturation

`GenApi::IFloat& Saturation`

Description: Controls the color saturation.

Visibility:

10.10.4.549 SaturationEnable

`GenApi::IBoolean& SaturationEnable`

Description: Enables/disables Saturation adjustment.

Visibility:

10.10.4.550 Scan3dAxisMax

`GenApi::IFloat& Scan3dAxisMax`

Description: Maximum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

10.10.4.551 Scan3dAxisMin

`GenApi::IFloat& Scan3dAxisMin`

Description: Minimum valid transmitted coordinate value of the selected Axis.

Visibility: Expert

10.10.4.552 Scan3dCoordinateOffset

```
GenApi::IFloat& Scan3dCoordinateOffset
```

Description: Offset when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

10.10.4.553 Scan3dCoordinateReferenceSelector

```
GenApi::IEnumerationT<Scan3dCoordinateReferenceSelectorEnums>& Scan3dCoordinateReferenceSelector
```

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

Visibility: Expert

10.10.4.554 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

10.10.4.555 Scan3dCoordinateScale

```
GenApi::IFloat& Scan3dCoordinateScale
```

Description: Scale factor when transforming a pixel from relative coordinates to world coordinates.

Visibility: Expert

10.10.4.556 Scan3dCoordinateSelector

```
GenApi::IEnumerationT<Scan3dCoordinateSelectorEnums>& Scan3dCoordinateSelector
```

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

Visibility: Expert

10.10.4.557 Scan3dCoordinateSystem

```
GenApi::IEnumerationT<Scan3dCoordinateSystemEnums>& Scan3dCoordinateSystem
```

Description: Specifies the Coordinate system to use for the device.

Visibility: Beginner

10.10.4.558 Scan3dCoordinateSystemReference

```
GenApi::IEnumerationT<Scan3dCoordinateSystemReferenceEnums>& Scan3dCoordinateSystemReference
```

Description: Defines coordinate system reference location.

Visibility: Expert

10.10.4.559 Scan3dCoordinateTransformSelector

```
GenApi::IEnumerationT<Scan3dCoordinateTransformSelectorEnums>& Scan3dCoordinateTransformSelector
```

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

Visibility: Expert

10.10.4.560 Scan3dDistanceUnit

```
GenApi::IEnumerationT<Scan3dDistanceUnitEnums>& Scan3dDistanceUnit
```

Description: Specifies the unit used when delivering calibrated distance data.

Visibility: Beginner

10.10.4.561 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

10.10.4.562 Scan3dInvalidDataValue

```
GenApi::IFloat& Scan3dInvalidDataValue
```

Description: Value which identifies a non-valid pixel if Scan3dInvalidDataFlag is enabled.

Visibility: Expert

10.10.4.563 Scan3dOutputMode

```
GenApi::IEnumerationT<Scan3dOutputModeEnums>& Scan3dOutputMode
```

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

Visibility: Expert

10.10.4.564 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

10.10.4.565 SensorDescription

```
GenApi::IString& SensorDescription
```

Description: Returns Sensor Description Visibility:

10.10.4.566 SensorDigitizationTaps

```
GenApi::IEnumerationT<SensorDigitizationTapsEnums>& SensorDigitizationTaps
```

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

Visibility: Expert

10.10.4.567 SensorHeight

```
GenApi::IInteger& SensorHeight
```

Description: Effective height of the sensor in pixels.

Visibility:

10.10.4.568 SensorShutterMode

```
GenApi::IEnumerationT<SensorShutterModeEnums>& SensorShutterMode
```

Description: Sets the shutter mode of the device.

Visibility:

10.10.4.569 SensorTaps

```
GenApi::IEnumerationT<SensorTapsEnums>& SensorTaps
```

Description: Number of taps of the camera sensor.

Visibility: Expert

10.10.4.570 SensorWidth

`GenApi::Integer& SensorWidth`

Description: Effective width of the sensor in pixels.

Visibility:

10.10.4.571 SequencerConfigurationMode

`GenApi::EnumerationT<SequencerConfigurationModeEnums>& SequencerConfigurationMode`

Description:

Controls whether or not a sequencer is in configuration mode.

Visibility:

10.10.4.572 SequencerConfigurationValid

`GenApi::EnumerationT<SequencerConfigurationValidEnums>& SequencerConfigurationValid`

Description:

Display whether the current sequencer configuration is valid to run.

Visibility:

10.10.4.573 SequencerFeatureEnable

`GenApi::Boolean& SequencerFeatureEnable`

Description:

Enables the selected feature and makes it active in all sequencer sets.

Visibility:

10.10.4.574 SequencerMode

`GenApi::EnumerationT<SequencerModeEnums>& SequencerMode`

Description: Controls whether or not a sequencer is active.

Visibility:

10.10.4.575 SequencerPathSelector

`GenApi::Integer& SequencerPathSelector`

Description:

Selects branching path to be used for subsequent settings.

Visibility:

10.10.4.576 SequencerSetActive

`GenApi::Integer& SequencerSetActive`

Description: Displays the currently active sequencer set.

Visibility:

10.10.4.577 SequencerSetLoad

`GenApi::Command& SequencerSetLoad`

Description:

Loads currently selected sequencer to the current device configuration.

Visibility:

10.10.4.578 SequencerSetNext

`GenApi::Integer& SequencerSetNext`

Description: Specifies the next sequencer set.

Visibility:

10.10.4.579 SequencerSetSave

`GenApi::Command& SequencerSetSave`

Description:

Saves the current device configuration to the currently selected sequencer set.

Visibility:

10.10.4.580 SequencerSetSelector

`GenApi::Integer& SequencerSetSelector`

Description:

Selects the sequencer set to which subsequent settings apply.

Visibility:

10.10.4.581 SequencerSetStart

`GenApi::Integer& SequencerSetStart`

Description: Sets the first sequencer set to be used.

Visibility:

10.10.4.582 SequencerSetValid

`GenApi::EnumerationT<SequencerSetValidEnums>& SequencerSetValid`

Description:

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

Visibility:

10.10.4.583 SequencerTriggerActivation

`GenApi::EnumerationT<SequencerTriggerActivationEnums>& SequencerTriggerActivation`

Description:

Specifies the activation mode of the sequencer trigger.

Visibility:

10.10.4.584 SequencerTriggerSource

`GenApi::EnumerationT<SequencerTriggerSourceEnums>& SequencerTriggerSource`

Description:

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

Visibility:

10.10.4.585 SerialPortBaudRate

`GenApi::IEnumerationT<SerialPortBaudRateEnums>& SerialPortBaudRate`

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

Visibility:

10.10.4.586 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:

10.10.4.587 SerialPortParity

`GenApi::IEnumerationT<SerialPortParityEnums>& SerialPortParity`

Description: This feature controls the parity used by the selected serial port.

Visibility:

10.10.4.588 SerialPortSelector

`GenApi::IEnumerationT<SerialPortSelectorEnums>& SerialPortSelector`

Description: Selects which serial port of the device to control.

Visibility:

10.10.4.589 SerialPortSource

`GenApi::IEnumerationT<SerialPortSourceEnums>& SerialPortSource`

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

Visibility:

10.10.4.590 SerialPortStopBits

`GenApi::IEnumerationT<SerialPortStopBitsEnums>& SerialPortStopBits`

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

Visibility:

10.10.4.591 SerialReceiveFramingErrorCount

`GenApi::Integer& SerialReceiveFramingErrorCount`

Description: Returns the number of framing errors that have occurred on the serial port.

Visibility:

10.10.4.592 SerialReceiveParityErrorCount

`GenApi::Integer& SerialReceiveParityErrorCount`

Description: Returns the number of parity errors that have occurred on the serial port.

Visibility:

10.10.4.593 SerialReceiveQueueClear

`GenApi::Command& SerialReceiveQueueClear`

Description: This is a command that clears the device serial port receive queue.

Visibility:

10.10.4.594 SerialReceiveQueueCurrentCharacterCount

`GenApi::Integer& SerialReceiveQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port receive queue.

Visibility:

10.10.4.595 SerialReceiveQueueMaxCharacterCount

`GenApi::Integer& SerialReceiveQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port receive queue.

Visibility:

10.10.4.596 SerialTransmitQueueCurrentCharacterCount

`GenApi::Integer& SerialTransmitQueueCurrentCharacterCount`

Description: Returns the number of characters currently in the serial port transmit queue.

Visibility:

10.10.4.597 SerialTransmitQueueMaxCharacterCount

`GenApi::Integer& SerialTransmitQueueMaxCharacterCount`

Description: >Returns the maximum number of characters in the serial port transmit queue.

Visibility:

10.10.4.598 Sharpening

`GenApi::Float& Sharpening`

Description:

Controls the amount to sharpen a signal.

The sharpened amount is proportional to the difference between a pixel and its neighbors. A negative value smooths out the difference, while a positive value amplifies the difference. You can boost by a maximum of 8x, but smoothing is limited to 1x (in float). Default value: 2.0

Visibility:

10.10.4.599 SharpeningAuto

`GenApi::Boolean& 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:

10.10.4.600 SharpeningEnable

`GenApi::Boolean& SharpeningEnable`

Description:

Enables/disables the sharpening feature.

Sharpening is disabled by default.

Visibility:

10.10.4.601 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:

10.10.4.602 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

10.10.4.603 SoftwareSignalSelector

`GenApi::IEnumerationT<SoftwareSignalSelectorEnums>& SoftwareSignalSelector`

Description: Selects which Software Signal features to control.

Visibility: Beginner

10.10.4.604 SourceCount

`GenApi::IInteger& SourceCount`

Description: Controls or returns the number of sources supported by the device.

Visibility: Beginner

10.10.4.605 SourceSelector

`GenApi::IEnumerationT<SourceSelectorEnums>& SourceSelector`

Description: Selects the source to control.

Visibility: Beginner

10.10.4.606 Test0001

```
GenApi::Integer& Test0001
```

Description: For testing only.

Visibility:

10.10.4.607 TestEventGenerate

```
GenApi::ICommand& TestEventGenerate
```

Description: This command generates a test event and sends it to the host.

Visibility:

10.10.4.608 TestPattern

```
GenApi::IEnumerationT<TestPatternEnums>& TestPattern
```

Description:

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

Visibility:

10.10.4.609 TestPatternGeneratorSelector

```
GenApi::IEnumerationT<TestPatternGeneratorSelectorEnums>& TestPatternGeneratorSelector
```

Description:

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

Visibility:

10.10.4.610 TestPendingAck

```
GenApi::Integer& 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

10.10.4.611 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

10.10.4.612 TimerDuration

`GenApi::IFloat& TimerDuration`

Description: Sets the duration (in microseconds) of the Timer pulse.

Visibility: Expert

10.10.4.613 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

10.10.4.614 TimerSelector

`GenApi::IEnumerationT<TimerSelectorEnums>& TimerSelector`

Description: Selects which Timer to configure.

Visibility: Expert

10.10.4.615 TimerStatus

`GenApi::IEnumerationT<TimerStatusEnums>& TimerStatus`

Description: Returns the current status of the Timer.

Visibility: Expert

10.10.4.616 TimerTriggerActivation

`GenApi::IEnumerationT<TimerTriggerActivationEnums>& TimerTriggerActivation`

Description: Selects the activation mode of the trigger to start the Timer.

Visibility: Expert

10.10.4.617 TimerTriggerSource

`GenApi::IEnumerationT<TimerTriggerSourceEnums>& TimerTriggerSource`

Description: Selects the source of the trigger to start the Timer.

Visibility: Expert

10.10.4.618 TimerValue

`GenApi::IFloat& TimerValue`

Description: Reads or writes the current value (in microseconds) of the selected Timer.

Visibility: Expert

10.10.4.619 Timestamp

`GenApi::IInteger& Timestamp`

Description: Reports the current value of the device timestamp counter.

Visibility: Expert

10.10.4.620 TimestampLatch

`GenApi::ICommand& TimestampLatch`

Description: Latches the current timestamp counter into TimestampLatchValue.

Visibility:

10.10.4.621 TimestampLatchValue

`GenApi::IInteger& TimestampLatchValue`

Description: Returns the latched value of the timestamp counter.

Visibility:

10.10.4.622 TimestampReset

`GenApi::ICommand& TimestampReset`

Description: Resets the current value of the device timestamp counter.

Visibility:

10.10.4.623 TLParamsLocked

`GenApi::Integer& TLParamsLocked`

Description: Visibility:

10.10.4.624 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

10.10.4.625 TransferBlockCount

`GenApi::Integer& 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:

10.10.4.626 TransferBurstCount

`GenApi::Integer& TransferBurstCount`

Description: Number of Block(s) to transfer for each TransferBurstStart trigger.

Visibility: Expert

10.10.4.627 TransferComponentSelector

`GenApi::IEnumerationT<TransferComponentSelectorEnums>& TransferComponentSelector`

Description: Selects the color component for the control of the TransferStreamChannel feature.

Visibility: Guru

10.10.4.628 TransferControlMode

`GenApi::IEnumerationT<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:

10.10.4.629 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:

10.10.4.630 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

10.10.4.631 TransferQueueCurrentBlockCount

```
GenApi::IInteger& TransferQueueCurrentBlockCount
```

Description: Returns number of data blocks (images) currently in the transfer queue.

Visibility:

10.10.4.632 TransferQueueMaxBlockCount

```
GenApi::IInteger& TransferQueueMaxBlockCount
```

Description: Returns the maximum number of data blocks (images) in the transfer queue Visibility:

10.10.4.633 TransferQueueMode

```
GenApi::IEnumerationT<TransferQueueModeEnums>& TransferQueueMode
```

Description: Specifies the operation mode of the transfer queue.

Visibility:

10.10.4.634 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:

10.10.4.635 TransferResume

`GenApi::ICommand& TransferResume`

Description: Resumes a data Blocks streaming that was previously paused by a TransferPause command.

Visibility: Guru

10.10.4.636 TransferSelector

`GenApi::IEnumerationT<TransferSelectorEnums>& TransferSelector`

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

Visibility: Expert

10.10.4.637 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:

10.10.4.638 TransferStatus

`GenApi::IBoolean& TransferStatus`

Description: Reads the status of the Transfer module signal selected by TransferStatusSelector.

Visibility: Guru

10.10.4.639 TransferStatusSelector

`GenApi::IEnumerationT<TransferStatusSelectorEnums>& TransferStatusSelector`

Description: Selects which status of the transfer module to read.

Visibility: Guru

10.10.4.640 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:

10.10.4.641 TransferStreamChannel

```
GenApi::Integer& 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

10.10.4.642 TransferTriggerActivation

```
GenApi::EnumerationT<TransferTriggerActivationEnums>& TransferTriggerActivation
```

Description: Specifies the activation mode of the transfer control trigger.

Visibility: Guru

10.10.4.643 TransferTriggerMode

```
GenApi::EnumerationT<TransferTriggerModeEnums>& TransferTriggerMode
```

Description: Controls if the selected trigger is active.

Visibility: Guru

10.10.4.644 TransferTriggerSelector

```
GenApi::EnumerationT<TransferTriggerSelectorEnums>& TransferTriggerSelector
```

Description: Selects the type of transfer trigger to configure.

Visibility: Guru

10.10.4.645 TransferTriggerSource

```
GenApi::EnumerationT<TransferTriggerSourceEnums>& TransferTriggerSource
```

Description: Specifies the signal to use as the trigger source for transfers.

Visibility: Guru

10.10.4.646 TriggerActivation

```
GenApi::EnumerationT<TriggerActivationEnums>& TriggerActivation
```

Description: Specifies the activation mode of the trigger.

Visibility:

10.10.4.647 TriggerDelay

`GenApi::IFloat& TriggerDelay`

Description:

Specifies the delay in microseconds (us) to apply after the trigger reception before activating it.

Visibility:

10.10.4.648 TriggerDivider

`GenApi::IInteger& TriggerDivider`

Description: Specifies a division factor for the incoming trigger pulses.

Visibility: Expert

10.10.4.649 TriggerEventTest

`GenApi::ICommand& TriggerEventTest`

Description: This command generates a test event and sends it to the host.

Visibility:

10.10.4.650 TriggerMode

`GenApi::IEnumerationT<TriggerModeEnums>& TriggerMode`

Description:

Controls whether or not trigger is active.

Visibility:

10.10.4.651 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

10.10.4.652 TriggerOverlap

`GenApi::IEnumerationT<TriggerOverlapEnums>& TriggerOverlap`

Description: Specifies the overlap mode of the trigger.

Visibility:

10.10.4.653 TriggerSelector

`GenApi::IEnumerationT<TriggerSelectorEnums>& TriggerSelector`

Description: Selects the type of trigger to configure.

Visibility:

10.10.4.654 TriggerSoftware

`GenApi::ICommand& TriggerSoftware`

Description:

Generates an internal trigger if Trigger Source is set to Software.

Visibility:

10.10.4.655 TriggerSource

`GenApi::IEnumerationT<TriggerSourceEnums>& TriggerSource`

Description:

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

Visibility:

10.10.4.656 UserOutputSelector

`GenApi::IEnumerationT<UserOutputSelectorEnums>& UserOutputSelector`

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

Visibility:

10.10.4.657 UserOutputValue

`GenApi::IBoolean& UserOutputValue`

Description: Value of the selected user output, either logic high (enabled) or logic low (disabled).

Visibility:

10.10.4.658 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:

10.10.4.659 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

10.10.4.660 UserSetDefault

`GenApi::IEnumerationT<UserSetDefaultEnums>& UserSetDefault`

Description:

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

Visibility:

10.10.4.661 UserSetFeatureEnable

`GenApi::IBoolean& UserSetFeatureEnable`

Description: Whether or not the selected feature is saved to user sets.

Visibility:

10.10.4.662 UserSetLoad

`GenApi::ICommand& UserSetLoad`

Description:

Loads the User Set specified by UserSetSelector to the device and makes it active.

Visibility:

10.10.4.663 UserSetSave

`GenApi::ICommand& UserSetSave`

Description:

Saves the User Set specified by UserSetSelector to the non-volatile memory of the device.

Visibility:

10.10.4.664 UserSetSelector

`GenApi::IEnumerationT<UserSetSelectorEnums>& UserSetSelector`

Description:

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

Visibility:

10.10.4.665 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:

10.10.4.666 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

10.10.4.667 WhiteClipSelector

`GenApi::IEnumerationT<WhiteClipSelectorEnums>& WhiteClipSelector`

Description: Selects which White Clip to control.

Visibility: Expert

10.10.4.668 Width

`GenApi::IInteger& Width`

Description:

Width of the image provided by the device (in pixels).

Visibility:

10.10.4.669 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:

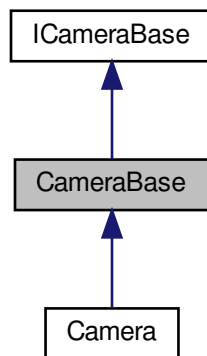
The documentation for this class was generated from the following file:

- `include/Camera.h`

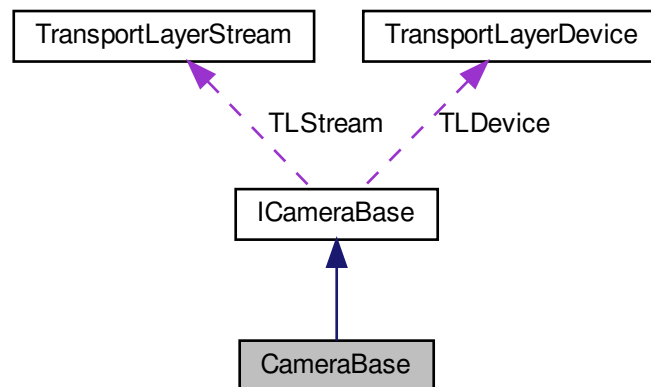
10.11 CameraBase Class Reference

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)
- void **WritePort** (uint64_t iAddress, const void *pBuffer, size_t iSize)
- void **BeginAcquisition** ()
BeginAcquisition Starts the image acquisition engine.
- void **EndAcquisition** ()
EndAcquisition Stops the image acquisition engine.
- **BufferOwnership** **GetBufferOwnership** () const
GetBufferOwnership Gets data buffer ownership.
- void **SetBufferOwnership** (const **BufferOwnership** mode)
SetBufferOwnership Sets data buffer ownership.
- uint64_t **GetUserBufferCount** () const
GetUserBufferCount Gets the number of user memory buffers.
- uint64_t **GetUserBufferSize** () const
GetUserBufferSize Gets the size of one user memory buffer (in bytes).
- uint64_t **GetUserBufferTotalSize** () const
GetUserBufferTotalSize Gets the total size of all the user memory buffers (in bytes).
- void **SetUserBuffers** (void *const pMemBuffers, uint64_t totalSize)
SetUserBuffers Specify contiguous user allocated memory to use as data buffers.
- void **SetUserBuffers** (void **const ppMemBuffers, const uint64_t bufferCount, const uint64_t bufferSize)
SetUserBuffers Specify non-contiguous user allocated memory to use as data buffers.
- **ImagePtr** **GetNextImage** (uint64_t grabTimeout=**EVENT_TIMEOUT_INFINITE**, uint64_t 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 **RegisterEvent** (**Event** &evtToRegister)
RegisterEvent(Event &) Registers a specific event for the camera.
- void **RegisterEvent** (**Event** &evtToRegister, const **GenICam::gcstring** &eventName)
RegisterEvent(Event &, const GenICam::gcstring&) Registers a specific event for the camera.
- void **UnregisterEvent** (**Event** &evtToUnregister)
UnregisterEvent Unregisters an event for the camera Events should be unregistered first before calling camera [DeInit](#)().

- 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 [InterfacelImpl](#)

Additional Inherited Members

10.11.1 Detailed Description

The base class for the camera object.

10.11.2 Constructor & Destructor Documentation

10.11.2.1 [~CameraBase\(\)](#)

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

Virtual Destructor.

10.11.2.2 CameraBase() [1/2]

```
CameraBase (
    void ) [protected]
```

Default constructor.

10.11.2.3 CameraBase() [2/2]

```
CameraBase (
    const CameraBase & ) [protected]
```

Copy constructor.

10.11.3 Member Function Documentation

10.11.3.1 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](#).

10.11.3.2 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\(\)](#)
[UnregisterEvent\(Event & evtToUnregister\)](#)

Implements [ICameraBase](#).

10.11.3.3 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](#).

10.11.3.4 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](#).

10.11.3.5 ForceIP()

```
void ForceIP ( ) [virtual]
```

ForceIP Forces the camera to be on the same subnet as its corresponding interface.

Implements [ICameraBase](#).

10.11.3.6 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](#).

10.11.3.7 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](#).

10.11.3.8 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](#).

10.11.3.9 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](#).

10.11.3.10 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](#).

10.11.3.11 GetNumDataStreams()

```
unsigned int GetNumDataStreams ( ) [virtual]
```

GetNumDataStreams Returns the number of streams that a device supports.

Returns

The number of data streams

Implements [ICameraBase](#).

10.11.3.12 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](#).

10.11.3.13 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](#).

10.11.3.14 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](#).

10.11.3.15 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](#).

10.11.3.16 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](#).

10.11.3.17 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](#).

10.11.3.18 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](#).

10.11.3.19 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](#).

10.11.3.20 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](#).

10.11.3.21 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](#).

10.11.3.22 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](#).

10.11.3.23 operator=()

```
CameraBase& operator= (
    const CameraBase & ) [protected]
```

Assignment operator.

10.11.3.24 ReadPort()

```
void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

10.11.3.25 RegisterEvent() [1/2]

```
void RegisterEvent (
    Event & evtToRegister ) [virtual]
```

[RegisterEvent\(Event &\)](#) Registers a specific event for the camera.

The camera has to be initialized first with a call to [Init\(\)](#) before registering for events.

See also

[Init\(\)](#)

Parameters

| | |
|----------------------|--------------------------------------|
| <i>evtToRegister</i> | The event to register for the camera |
|----------------------|--------------------------------------|

Implements [ICameraBase](#).

10.11.3.26 RegisterEvent() [2/2]

```
void RegisterEvent (
    Event & evtToRegister,
    const GenICam::gcstring & eventName ) [virtual]
```

[RegisterEvent\(Event &, const GenICam::gcstring&\)](#) Registers a specific event for the camera.

See also

[Init\(\)](#)

Parameters

| | |
|----------------------|--------------------------------------|
| <i>evtToRegister</i> | The event to register for the camera |
| <i>eventName</i> | The event name to register |

Implements [ICameraBase](#).

10.11.3.27 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

| | |
|-------------|--|
| <i>mode</i> | System owned or User owned buffers |
|-------------|--|

Implements [ICameraBase](#).

10.11.3.28 SetUserBuffers() [1/2]

```
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](#).

10.11.3.29 SetUserBuffers() [2/2]

```
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](#).

10.11.3.30 UnregisterEvent()

```
void UnregisterEvent (
    Event & evtToUnregister ) [virtual]
```

UnregisterEvent Unregisters an event for the camera Events should be unregistered first 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

[DeInit\(\)](#)

Parameters

| | |
|------------------------|---|
| <i>evtToUnregister</i> | The event to unregister from the camera |
|------------------------|---|

Implements [ICameraBase](#).

10.11.3.31 WritePort()

```
void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [virtual]
```

Implements [ICameraBase](#).

10.11.4 Friends And Related Function Documentation

10.11.4.1 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

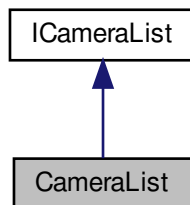
The documentation for this class was generated from the following file:

- [include/CameraBase.h](#)

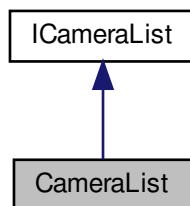
10.12 CameraList Class Reference

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.
- 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 [Append](#) ([CameraList](#) &otherList)
Appends a camera list to the current list.

Additional Inherited Members

10.12.1 Detailed Description

Used to hold a list of camera objects.

10.12.2 Constructor & Destructor Documentation

10.12.2.1 [CameraList\(\)](#) [1/2]

```
CameraList (
    void )
```

Default constructor.

10.12.2.2 [~CameraList\(\)](#)

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

Virtual destructor.

10.12.2.3 [CameraList\(\)](#) [2/2]

```
CameraList (
    const CameraList & iface )
```

Copy constructor.

10.12.3 Member Function Documentation

10.12.3.1 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](#).

10.12.3.2 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](#).

10.12.3.3 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 an camera object.

Implements [ICameraList](#).

10.12.3.4 GetBySerial()

```
CameraPtr GetBySerial (
    std::string serialNumber ) const [virtual]
```

Returns a pointer to a camera object with the specified serial number.

This function will return a NULL [CameraPtr](#) if no matching camera serial is found.

Parameters

| | |
|---------------------|--|
| <i>serialNumber</i> | The serial number of the camera object to retrieve |
|---------------------|--|

Returns

A pointer to an camera object.

Implements [ICameraList](#).

10.12.3.5 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](#).

10.12.3.6 operator=()

```
CameraList& operator= (
    const CameraList & iface )
```

Assignment operator.

10.12.3.7 operator[]()

```
CameraPtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [ICameraList](#).

10.12.3.8 RemoveByIndex()

```
void RemoveByIndex (
    unsigned int index ) [virtual]
```

Removes a camera at "index" and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_INVALID_PARAMETER error if the input index is out of range.

Parameters

| | |
|--------------|--|
| <i>index</i> | The index at which to remove the Camera object |
|--------------|--|

Implements [ICameraList](#).

10.12.3.9 RemoveBySerial()

```
void RemoveBySerial (
    std::string serialNumber ) [virtual]
```

Removes a camera using its serial number and destroys its corresponding reference counted object.

This function will throw a [Spinnaker](#) exception with SPINNAKER_ERR_NOT_AVAILABLE error if no matching camera serial is found.

Parameters

| | |
|---------------------|--|
| <i>serialNumber</i> | The serial number of the Camera object to remove |
|---------------------|--|

Implements [ICameraList](#).

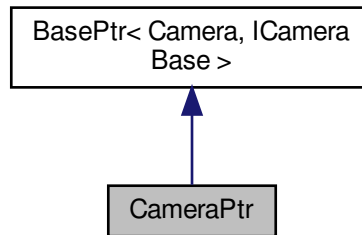
The documentation for this class was generated from the following file:

- include/[CameraList.h](#)

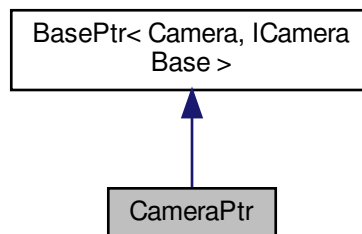
10.13 CameraPtr Class Reference

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

10.13.1 Detailed Description

A reference tracked pointer to a camera object.

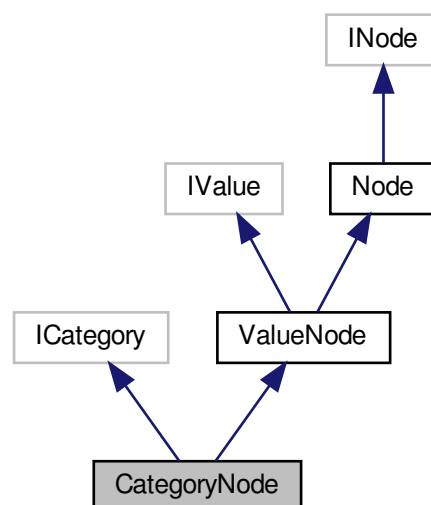
The documentation for this class was generated from the following file:

- [include/CameraPtr.h](#)

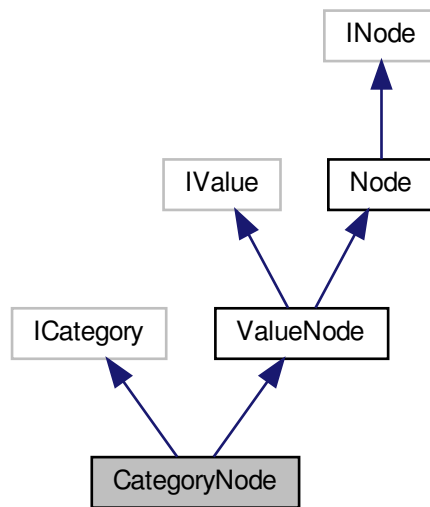
10.14 CategoryNode Class Reference

[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

10.14.1 Detailed Description

[Interface](#) for string properties.

10.14.2 Constructor & Destructor Documentation

10.14.2.1 CategoryNode() [1/2]

[CategoryNode](#) ()

10.14.2.2 CategoryNode() [2/2]

```
CategoryNode (
    std::shared_ptr< Node::NodeImpl > pCategory )
```

10.14.2.3 ~CategoryNode()

```
virtual ~CategoryNode ( ) [virtual]
```

10.14.3 Member Function Documentation

10.14.3.1 GetFeatures()

```
virtual void GetFeatures (
    FeatureList_t & Features ) const [virtual]
```

Get all features of the category (including sub-categories)

10.14.3.2 SetReference()

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

overload SetReference for Value

Reimplemented from [ValueNode](#).

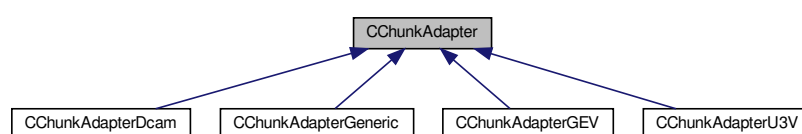
The documentation for this class was generated from the following file:

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

10.15 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](#)

10.15.1 Detailed Description

Connects a chunked buffer to a node map.

10.15.2 Constructor & Destructor Documentation

10.15.2.1 ~CChunkAdapter()

```
virtual ~CChunkAdapter ( ) [virtual]
```

Destructor.

10.15.2.2 CChunkAdapter()

```
CChunkAdapter (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 ) [protected]
```

Serves as default constructor.

10.15.3 Member Function Documentation

10.15.3.1 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](#).

10.15.3.2 AttachNodeMap()

```
void AttachNodeMap (
    INodeMap * pNodeMap )
```

Attaches to a node map and retrieves the chunk ports.

10.15.3.3 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](#).

10.15.3.4 ClearCaches()

```
void ClearCaches ( )
```

Clears the chunk caches.

10.15.3.5 DetachBuffer()

```
void DetachBuffer ( )
```

Detaches a buffer.

10.15.3.6 DetachNodeMap()

```
void DetachNodeMap ( )
```

Detaches from the node map.

10.15.3.7 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the buffer.

10.15.4 Member Data Documentation

10.15.4.1 m_pChunkAdapter

```
void* m_pChunkAdapter [protected]
```

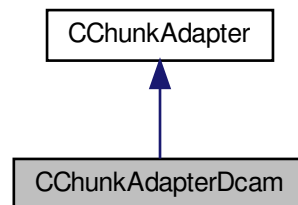
The documentation for this class was generated from the following file:

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

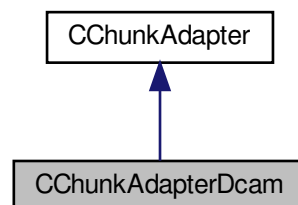
10.16 CChunkAdapterDcam Class Reference

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

10.16.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

10.16.2 Constructor & Destructor Documentation

10.16.2.1 CChunkAdapterDcam()

```
CChunkAdapterDcam (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.16.2.2 ~CChunkAdapterDcam()

```
virtual ~CChunkAdapterDcam ( ) [virtual]
```

Destructor.

10.16.3 Member Function Documentation

10.16.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.16.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

10.16.3.3 CheckCRC()

```
bool CheckCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

Checks CRC sum of buffer.

10.16.3.4 HasCRC()

```
bool HasCRC (
    uint8_t * pBuffer,
    int64_t BufferLength )
```

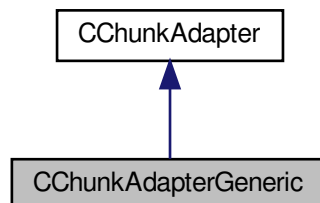
Checks if buffer has a CRC attached.

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

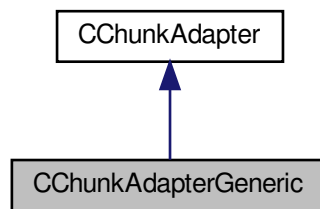
- include/SpinGenApi/[ChunkAdapterDcam.h](#)

10.17 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

10.17.1 Constructor & Destructor Documentation

10.17.1.1 CChunkAdapterGeneric()

```

CChunkAdapterGeneric (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
  
```

10.17.1.2 ~CChunkAdapterGeneric()

```

virtual ~CChunkAdapterGeneric ( ) [virtual]
  
```

10.17.2 Member Function Documentation

10.17.2.1 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](#).

10.17.2.2 AttachBuffer() [2/3]

```
virtual void AttachBuffer (  
    uint8_t * pBuffer,  
    SingleChunkData_t * ChunkData,  
    int64_t NumChunks,  
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

10.17.2.3 AttachBuffer() [3/3]

```
virtual void AttachBuffer (  
    uint8_t * pBuffer,  
    SingleChunkDataStr_t * ChunkData,  
    int64_t NumChunks,  
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

10.17.2.4 CheckBufferLayout()

```
virtual bool CheckBufferLayout (  
    uint8_t * pBuffer,  
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

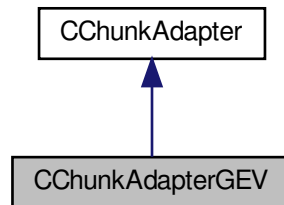
The documentation for this class was generated from the following file:

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

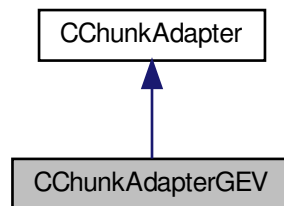
10.18 CChunkAdapterGEV Class Reference

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

10.18.1 Detailed Description

Connects a chunked DCAM buffer to a node map.

10.18.2 Constructor & Destructor Documentation

10.18.2.1 CChunkAdapterGEV()

```
CChunkAdapterGEV (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.18.2.2 ~CChunkAdapterGEV()

```
virtual ~CChunkAdapterGEV ( ) [virtual]
```

Destructor.

10.18.3 Member Function Documentation

10.18.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.18.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

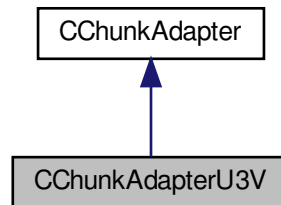
The documentation for this class was generated from the following file:

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

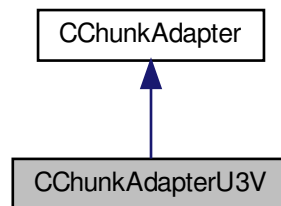
10.19 CChunkAdapterU3V Class Reference

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

10.19.1 Detailed Description

Connects a chunked U3V buffer to a node map.

10.19.2 Constructor & Destructor Documentation

10.19.2.1 CChunkAdapterU3V()

```
CChunkAdapterU3V (
    INodeMap * pNodeMap = NULL,
    int64_t MaxChunkCacheSize = -1 )
```

Constructor.

10.19.2.2 ~CChunkAdapterU3V()

```
virtual ~CChunkAdapterU3V ( ) [virtual]
```

Destructor.

10.19.3 Member Function Documentation

10.19.3.1 AttachBuffer()

```
virtual void AttachBuffer (
    uint8_t * pBuffer,
    int64_t BufferLength,
    AttachStatistics_t * pAttachStatistics = NULL ) [virtual]
```

Attaches a buffer to the matching ChunkPort.

Implements [CChunkAdapter](#).

10.19.3.2 CheckBufferLayout()

```
virtual bool CheckBufferLayout (
    uint8_t * pBuffer,
    int64_t BufferLength ) [virtual]
```

Checks if a buffer contains chunks in a known format.

Implements [CChunkAdapter](#).

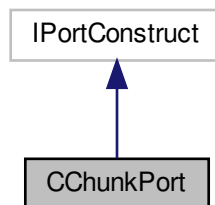
The documentation for this class was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

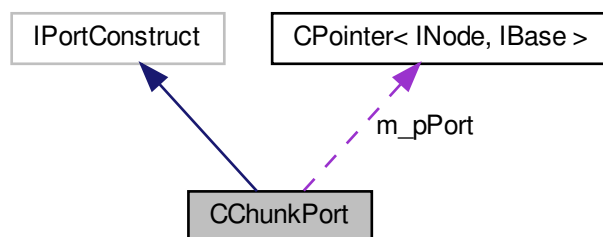
10.20 CChunkPort Class Reference

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](#)

10.20.1 Detailed Description

Port attachable to a chunk in a buffer.

10.20.2 Constructor & Destructor Documentation

10.20.2.1 CChunkPort()

```
CChunkPort (
    IPort * pPort = NULL )
```

Constructor; can attach to a port.

10.20.2.2 ~CChunkPort()

`~CChunkPort ()`

Destructor; detaches from the port.

10.20.3 Member Function Documentation

10.20.3.1 AttachChunk()

```
void AttachChunk (
    uint8_t * pBaseAddress,
    int64_t ChunkOffset,
    int64_t Length,
    bool Cache )
```

Attaches the Chunk to the ChunkPort.

10.20.3.2 AttachPort()

```
bool AttachPort (
    ::Spinnaker::GenApi::IPort * pPort )
```

Attaches the ChunkPort to the Port.

10.20.3.3 CheckChunkID() [1/2]

```
bool CheckChunkID (
    uint8_t * pChunkIDBuffer,
    int ChunkIDLength )
```

Checks if a ChunkID matches.

10.20.3.4 CheckChunkID() [2/2]

```
bool CheckChunkID (
    uint64_t ChunkID )
```

Checks if a ChunkID matches, version using uint64_t ID representation.

10.20.3.5 ClearCache()

```
void ClearCache ( )
```

Clears the chunk cache.

10.20.3.6 DetachChunk()

```
void DetachChunk ( )
```

Detaches the Chunk from the ChunkPort.

10.20.3.7 DetachPort()

```
void DetachPort ( )
```

Detaches the ChunkPort to the Port.

10.20.3.8 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

10.20.3.9 GetChunkIDLength()

```
int GetChunkIDLength ( )
```

Gets the ChunkID length.

10.20.3.10 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.20.3.11 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.20.3.12 InvalidateNode()

```
void InvalidateNode ( )
```

10.20.3.13 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.20.3.14 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [virtual]
```

Called from the port node to give the chunk port a pointer to itself.

10.20.3.15 UpdateBuffer()

```
void UpdateBuffer (
    uint8_t * pBaseAddress )
```

Updates the base address of the chunk.

10.20.3.16 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.20.4 Member Data Documentation

10.20.4.1 m_pChunkPort

`void* m_pChunkPort` [protected]

10.20.4.2 m_pPort

`CNodePtr m_pPort` [protected]

10.20.4.3 m_pPortAdapter

`std::shared_ptr<PortAdapter> m_pPortAdapter` [protected]

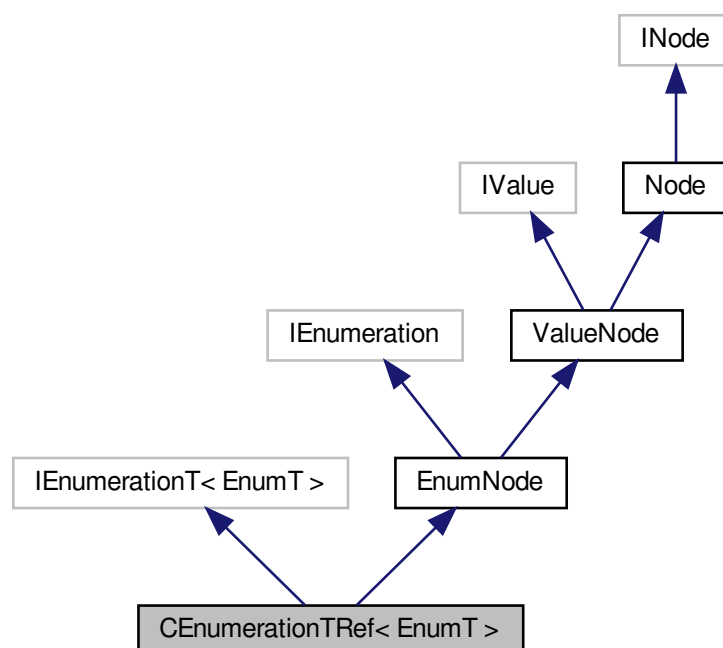
The documentation for this class was generated from the following file:

- `include/SpinGenApi/ChunkPort.h`

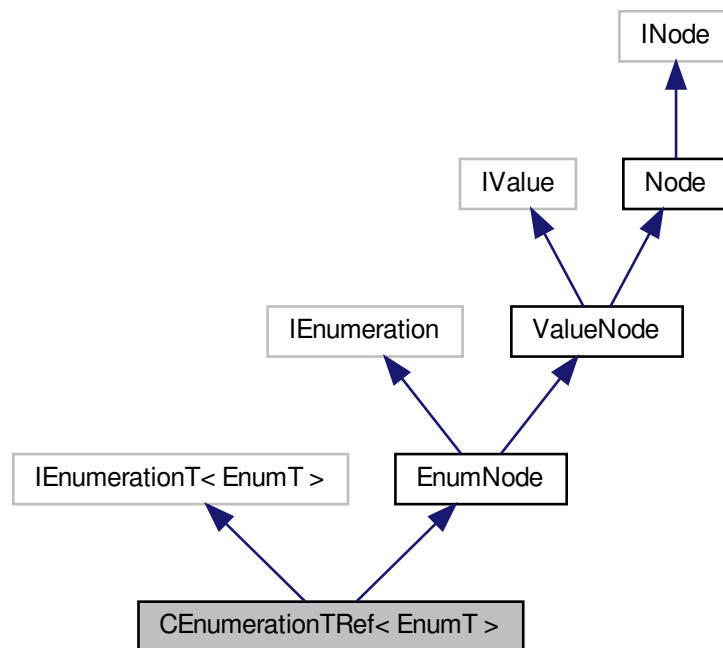
10.21 CEnumerationTRef< EnumT > Class Template Reference

[Interface](#) for string properties.

Inheritance diagram for CEnumerationTRef< EnumT >:



Collaboration diagram for CEnumerationTRef< EnumT >:



Public Member Functions

- `CEnumerationTRef` ()
- `CEnumerationTRef` (std::shared_ptr< Node::NodeImpl > pEnumeration)
- virtual `~CEnumerationTRef` ()
- virtual void `SetValue` (EnumT Value, bool `Verify`=true)
Set node value.
- virtual `IEnumeration` & `operator=` (EnumT Value)
Set node value.
- virtual EnumT `GetValue` (bool `Verify`=false, bool IgnoreCache=false)
Get node value.
- virtual EnumT `operator()` ()
Get node value.
- virtual `IEnumeration` & `operator=` (const `GenlCam::gcstring` &ValueStr)
Set node value.
- virtual `IEnumEntry` * `GetEntry` (const EnumT Value)
returns the `EnumEntry` object belonging to the Value
- virtual `IEnumEntry` * `GetEntry` (const int64_t IntValue)
Get an entry node by its IntValue.
- virtual `IEnumEntry` * `GetCurrentEntry` (bool `Verify`=false, bool IgnoreCache=false)
Get the current entry.
- virtual void `SetReference` (`INode` *pBase)
overload `SetReference` for `EnumerationT`

- virtual void [SetEnumReference](#) (int Index, [GenICam::gcstring](#) Name)
sets the Enum value corresponding to a value
- virtual void [SetNumEnums](#) (int NumEnums)
sets the number of enum values

Additional Inherited Members

10.21.1 Detailed Description

```
template<class EnumT>
class Spinnaker::GenApi::CEnumerationTRef< EnumT >
```

[Interface](#) for string properties.

10.21.2 Constructor & Destructor Documentation

10.21.2.1 CEnumerationTRef() [1/2]

```
CEnumerationTRef ( )
```

10.21.2.2 CEnumerationTRef() [2/2]

```
CEnumerationTRef (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

10.21.2.3 ~CEnumerationTRef()

```
virtual ~CEnumerationTRef ( ) [virtual]
```

10.21.3 Member Function Documentation

10.21.3.1 GetCurrentEntry()

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

Get the current entry.

Reimplemented from [EnumNode](#).

10.21.3.2 GetEntry() [1/2]

```
virtual IEnumEntry* GetEntry (
    const EnumT Value ) [virtual]
```

returns the EnumEntry object belonging to the Value

10.21.3.3 GetEntry() [2/2]

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

Get an entry node by its IntValue.

Reimplemented from [EnumNode](#).

10.21.3.4 GetValue()

```
virtual EnumT GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

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

Returns

The value read

10.21.3.5 operator()

```
virtual EnumT operator() ( ) [virtual]
```

Get node value.

10.21.3.6 operator=() [1/2]

```
virtual IEnumeration& operator= (
    EnumT Value ) [virtual]
```

Set node value.

10.21.3.7 operator=() [2/2]

```
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](#).

10.21.3.8 SetEnumReference()

```
virtual void SetEnumReference (
    int Index,
    GenICam::gcstring Name ) [virtual]
```

sets the Enum value corresponding to a value

10.21.3.9 SetNumEnums()

```
virtual void SetNumEnums (
    int NumEnums ) [virtual]
```

sets the number of enum values

10.21.3.10 SetReference()

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

overload SetReference for EnumerationT

Reimplemented from [EnumNode](#).

10.21.3.11 SetValue()

```
virtual void SetValue (
    EnumT Value,
    bool Verify = true ) [virtual]
```

Set node value.

Parameters

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

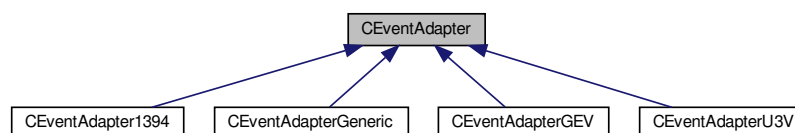
The documentation for this class was generated from the following file:

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

10.22 CEventAdapter Class Reference

Delivers Events to ports.

Inheritance diagram for CEventAdapter:



Public Member Functions

- [CEventAdapter](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapter](#) ()

Destructor.

- virtual void [AttachNodeMap](#) (INodeMap *pNodeMap)
Attaches to a node map and retrieves the chunk ports.
- virtual void [DetachNodeMap](#) ()
Detaches from the node map.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)=0
Deliver message.

Protected Attributes

- void * [m_pEventAdapter](#)

10.22.1 Detailed Description

Delivers Events to ports.

10.22.2 Constructor & Destructor Documentation

10.22.2.1 CEventAdapter()

```
CEventAdapter (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.22.2.2 ~CEventAdapter()

```
virtual ~CEventAdapter ( ) [virtual]
```

Destructor.

10.22.3 Member Function Documentation

10.22.3.1 AttachNodeMap()

```
virtual void AttachNodeMap (
    INodeMap * pNodeMap ) [virtual]
```

Attaches to a node map and retrieves the chunk ports.

10.22.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [pure virtual]
```

Deliver message.

Implemented in [CEventAdapterGEV](#), [CEventAdapterU3V](#), [CEventAdapter1394](#), and [CEventAdapterGeneric](#).

10.22.3.3 DetachNodeMap()

```
virtual void DetachNodeMap ( ) [virtual]
```

Detaches from the node emap.

10.22.4 Member Data Documentation

10.22.4.1 m_pEventAdapter

```
void* m_pEventAdapter [protected]
```

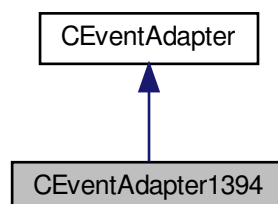
The documentation for this class was generated from the following file:

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

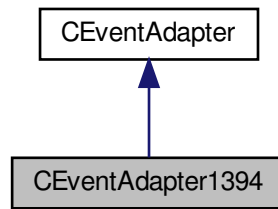
10.23 CEventAdapter1394 Class Reference

Distribute the events to the node map.

Inheritance diagram for CEventAdapter1394:



Collaboration diagram for CEventAdapter1394:



Public Member Functions

- [CEventAdapter1394](#) ([INodeMap](#) *pNodeMap=NULL)
constructor
- virtual [~CEventAdapter1394](#) ()
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) ([EventData1394](#) &[Event](#), uint32_t numBytes)
distributes events to node map

Additional Inherited Members

10.23.1 Detailed Description

Distribute the events to the node map.

10.23.2 Constructor & Destructor Documentation

10.23.2.1 CEventAdapter1394()

```

CEventAdapter1394 (
    INodeMap * pNodeMap = NULL ) [explicit]

```

constructor

10.23.2.2 ~CEventAdapter1394()

```

virtual ~CEventAdapter1394 ( ) [virtual]

```

10.23.3 Member Function Documentation

10.23.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    EventData1394 & Event,
    uint32_t numBytes )
```

distributes events to node map

10.23.3.2 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

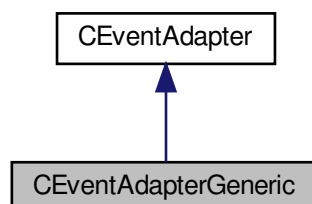
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapter1394.h](#)

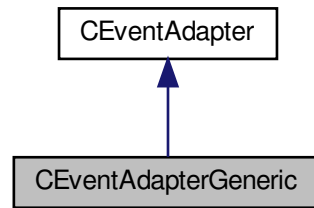
10.24 CEventAdapterGeneric Class Reference

Connects a generic event to a node map.

Inheritance diagram for CEventAdapterGeneric:



Collaboration diagram for CEventAdapterGeneric:



Public Member Functions

- [CEventAdapterGeneric](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGeneric](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, const [GenICam::gcstring](#) &EventID)
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes, uint64_t EventID)

Additional Inherited Members

10.24.1 Detailed Description

Connects a generic event to a node map.

10.24.2 Constructor & Destructor Documentation

10.24.2.1 CEventAdapterGeneric()

```

CEventAdapterGeneric (
    INodeMap * pNodeMap = NULL )
  
```

Constructor.

10.24.2.2 ~CEventAdapterGeneric()

```
virtual ~CEventAdapterGeneric ( ) [virtual]
```

Destructor.

10.24.3 Member Function Documentation

10.24.3.1 DeliverMessage() [1/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

10.24.3.2 DeliverMessage() [2/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    const GenICam::gcstring & EventID ) [virtual]
```

10.24.3.3 DeliverMessage() [3/3]

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes,
    uint64_t EventID ) [virtual]
```

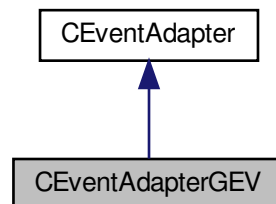
The documentation for this class was generated from the following file:

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

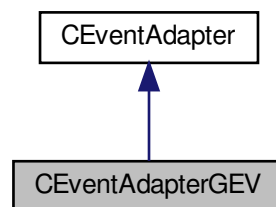
10.25 CEventAdapterGEV Class Reference

Connects a GigE [Event](#) to a node map.

Inheritance diagram for CEventAdapterGEV:



Collaboration diagram for CEventAdapterGEV:



Public Member Functions

- [CEventAdapterGEV](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterGEV](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [GVCP_EVENT_REQUEST](#) *pEvent)
Delivers the Events listed in the [Event](#) packet.
- void [DeliverEventMessage](#) (const [GVCP_EVENTDATA_REQUEST](#) *pEventData)
Delivers the [Event](#) + Data listed in the EventData packet.

Additional Inherited Members

10.25.1 Detailed Description

Connects a GigE [Event](#) to a node map.

10.25.2 Constructor & Destructor Documentation

10.25.2.1 CEventAdapterGEV()

```
CEventAdapterGEV (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.25.2.2 ~CEventAdapterGEV()

```
virtual ~CEventAdapterGEV ( ) [virtual]
```

Destructor.

10.25.3 Member Function Documentation

10.25.3.1 DeliverEventMessage() [1/2]

```
void DeliverEventMessage (
    const GVCP_EVENT_REQUEST * pEvent )
```

Delivers the Events listed in the [Event](#) packet.

10.25.3.2 DeliverEventMessage() [2/2]

```
void DeliverEventMessage (
    const GVCP_EVENTDATA_REQUEST * pEventData )
```

Delivers the [Event](#) + Data listed in the EventData packet.

10.25.3.3 DeliverMessage()

```
virtual void DeliverMessage (
    const uint8_t msg[],
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

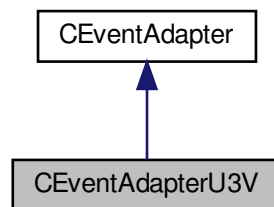
The documentation for this class was generated from the following file:

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

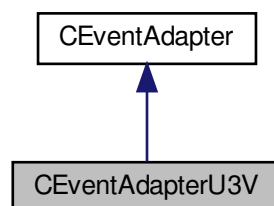
10.26 CEventAdapterU3V Class Reference

Connects a U3V [Event](#) to a node map.

Inheritance diagram for CEventAdapterU3V:



Collaboration diagram for CEventAdapterU3V:



Public Member Functions

- [CEventAdapterU3V](#) ([INodeMap](#) *pNodeMap=NULL)
Constructor.
- virtual [~CEventAdapterU3V](#) ()
Destructor.
- virtual void [DeliverMessage](#) (const uint8_t msg[], uint32_t numBytes)
Deliver message.
- void [DeliverEventMessage](#) (const [U3V_EVENT_MESSAGE](#) *pEventMessage)
Delivers the [Event](#) + Data listed in the packet.

Additional Inherited Members

10.26.1 Detailed Description

Connects a U3V [Event](#) to a node map.

10.26.2 Constructor & Destructor Documentation

10.26.2.1 CEventAdapterU3V()

```
CEventAdapterU3V (
    INodeMap * pNodeMap = NULL )
```

Constructor.

10.26.2.2 ~CEventAdapterU3V()

```
virtual ~CEventAdapterU3V ( ) [virtual]
```

Destructor.

10.26.3 Member Function Documentation

10.26.3.1 DeliverEventMessage()

```
void DeliverEventMessage (
    const U3V\_EVENT\_MESSAGE * pEventMessage )
```

Delivers the [Event](#) + Data listed in the packet.

10.26.3.2 DeliverMessage()

```
virtual void DeliverMessage (  
    const uint8_t msg[],  
    uint32_t numBytes ) [virtual]
```

Deliver message.

Implements [CEventAdapter](#).

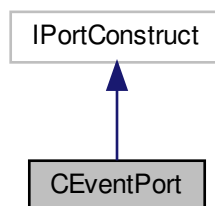
The documentation for this class was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

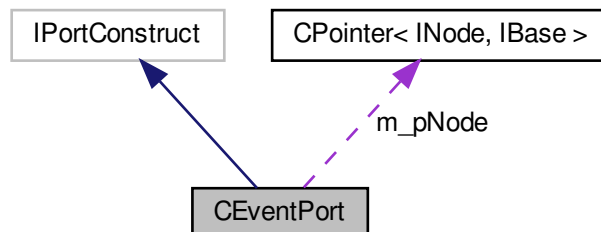
10.27 CEventPort Class Reference

Port attachable to an event.

Inheritance diagram for CEventPort:



Collaboration diagram for CEventPort:



Public Member Functions

- [CEventPort](#) ([INode](#) *pNode=NULL)
Constructor; can attach to a node.
- [~CEventPort](#) ()
Destructor; detaches from the port.
- virtual [EAccessMode](#) [GetAccessMode](#) () const
Get the access mode of the node.
- virtual [EInterfaceType](#) [GetPrincipalInterfaceType](#) () const
Get the type of the main interface of a node.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (::Spinnaker::GenApi::IPort *pPort)
Called from the port node to give the chunk port a pointer to itself.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- void [InvalidateNode](#) ()
- bool [AttachNode](#) (::Spinnaker::GenApi::INode *pNode)
Attaches to the [Node](#).
- void [DetachNode](#) ()
Detaches from the [Node](#).
- int [GetEventIDLength](#) ()
Gets the EventID length.
- bool [CheckEventID](#) (uint8_t *pEventIDBuffer, int EventIDLength)
Checks if a EventID matches.
- bool [CheckEventID](#) (uint64_t EventID)
Checks if a EventID matches, version using uint64_t ID representation.
- void [AttachEvent](#) (uint8_t *pBaseAddress, int64_t [Length](#))
Attaches the an [Event](#) to the EventPort.
- void [DetachEvent](#) ()
Detaches the [Event](#) from the EventPort.

Protected Attributes

- [CNodePtr](#) [m_pNode](#)
- std::shared_ptr< PortAdapter > [m_pPortAdapter](#)
- void * [m_pEventPort](#)

10.27.1 Detailed Description

Port attachable to an event.

10.27.2 Constructor & Destructor Documentation

10.27.2.1 CEventPort()

```
CEventPort (
    INode * pNode = NULL )
```

Constructor; can attach to a node.

10.27.2.2 ~CEventPort()

```
~CEventPort ( )
```

Destructor; detaches from the port.

10.27.3 Member Function Documentation

10.27.3.1 AttachEvent()

```
void AttachEvent (
    uint8_t * pBaseAddress,
    int64_t Length )
```

Attaches the an [Event](#) to the EventPort.

10.27.3.2 AttachNode()

```
bool AttachNode (
    ::Spinnaker::GenApi::INode * pNode )
```

Attaches to the [Node](#).

10.27.3.3 CheckEventID() [1/2]

```
bool CheckEventID (
    uint8_t * pEventIDBuffer,
    int EventIDLength )
```

Checks if a EventID matches.

10.27.3.4 CheckEventID() [2/2]

```
bool CheckEventID (
    uint64_t EventID )
```

Checks if a EventID matches, version using uint64_t ID representation.

10.27.3.5 DetachEvent()

```
void DetachEvent ( )
```

Detaches the [Event](#) from the EventPort.

10.27.3.6 DetachNode()

```
void DetachNode ( )
```

Detaches from the [Node](#).

10.27.3.7 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

10.27.3.8 GetEventIDLength()

```
int GetEventIDLength ( )
```

Gets the EventID length.

10.27.3.9 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.27.3.10 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.27.3.11 InvalidateNode()

```
void InvalidateNode ( )
```

10.27.3.12 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.27.3.13 SetPortImpl()

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

Called from the port node to give the chunk port a pointer to itself.

10.27.3.14 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.27.4 Member Data Documentation

10.27.4.1 m_pEventPort

```
void* m_pEventPort [protected]
```

10.27.4.2 m_pNode

```
CNodePtr m_pNode [protected]
```

10.27.4.3 m_pPortAdapter

```
std::shared_ptr<PortAdapter> m_pPortAdapter [protected]
```

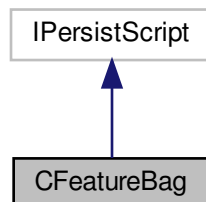
The documentation for this class was generated from the following file:

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

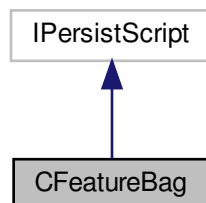
10.28 CFeatureBag Class Reference

Bag holding streamable features of a nodetree.

Inheritance diagram for CFeatureBag:



Collaboration diagram for CFeatureBag:



Public Member Functions

- [CFeatureBag](#) ()
- virtual [~CFeatureBag](#) ()
- virtual void [SetInfo](#) ([GenICam::gcstring](#) &Info)
sets information about the node map
- virtual void [PersistFeature](#) ([IValue](#) &item)
Stores a feature.
- bool [LoadFromBag](#) ([INodeMap](#) *pNodeMap, bool [Verify](#)=true, [GenICam::gcstring_vector](#) *pErrorList=NULL)
Loads the features from the bag to the node tree.
- int64_t [StoreToBag](#) ([INodeMap](#) *pNodeMap, const int MaxNumPersistSkriptEntries=-1)
Stores the streamable nodes to this feature bag.
- bool [operator==](#) (const [CFeatureBag](#) &FeatureBag) const
compares the content of two feature bags
- void * [GetFeatureBagHandle](#) ()

10.28.1 Detailed Description

Bag holding streamable features of a nodetree.

10.28.2 Constructor & Destructor Documentation

10.28.2.1 CFeatureBag()

```
CFeatureBag ( )
```

10.28.2.2 ~CFeatureBag()

```
virtual ~CFeatureBag ( ) [virtual]
```

10.28.3 Member Function Documentation

10.28.3.1 GetFeatureBagHandle()

```
void* GetFeatureBagHandle ( )
```

10.28.3.2 LoadFromBag()

```
bool LoadFromBag (
    INodeMap * pNodeMap,
    bool Verify = true,
    GenICam::gcstring\_vector * pErrorList = NULL )
```

Loads the features from the bag to the node tree.

Parameters

| | |
|-------------------|---|
| <i>pNodeMap</i> | The node map |
| <i>Verify</i> | If true, all streamable features are read back |
| <i>pErrorList</i> | If an error occurs during loading the error message is stored in the list and the loading continues |

For Verify=true the list of names in the feature bag is replayed again. If a node is a selector it's value is set to the value from the feature bag If not the value is read from the camera and compared with the value from the feature bag.

10.28.3.3 operator==()

```
bool operator== (
    const CFeatureBag & FeatureBag ) const
```

compares the content of two feature bags

10.28.3.4 PersistFeature()

```
virtual void PersistFeature (
    IValue & item ) [virtual]
```

Stores a feature.

10.28.3.5 SetInfo()

```
virtual void SetInfo (
    GenICam::gcstring & Info ) [virtual]
```

sets information about the node map

10.28.3.6 StoreToBag()

```
int64_t StoreToBag (
    INodeMap * pNodeMap,
    const int MaxNumPersistSkriptEntries = -1 )
```

Stores the streamable nodes to this feature bag.

Parameters

| | |
|-----------------------------------|--|
| <i>pNodeMap</i> | The node map to persist |
| <i>MaxNumPersistSkriptEntries</i> | The max number of entries in the container; -1 means unlimited |

Returns

number of entries in the bag

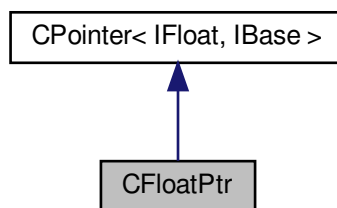
The documentation for this class was generated from the following file:

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

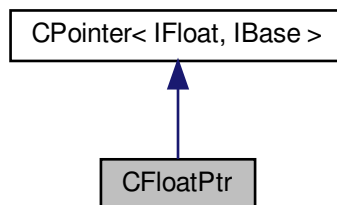
10.29 CFloatPtr Class Reference

SmartPointer for IFloat interface pointer.

Inheritance diagram for CFloatPtr:



Collaboration diagram for CFloatPtr:

**Public Member Functions**

- [CFloatPtr](#) () throw ()
Default constructor.
- [CFloatPtr](#) (IBase *pB)
Constructor from IBase pointer type.
- void [operator=](#) (IBase *pB)
Assign IBase Pointer.
- [IInteger](#) * [GetIntAlias](#) ()
gets the interface of an integer alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an enum alias node.

Additional Inherited Members

10.29.1 Detailed Description

SmartPointer for IFloat interface pointer.

10.29.2 Constructor & Destructor Documentation

10.29.2.1 CFloatPtr() [1/2]

```
CFloatPtr ( ) throw ( ) [inline]
```

Default constructor.

10.29.2.2 CFloatPtr() [2/2]

```
CFloatPtr (
    IBase * pB ) [inline]
```

Constructor from IBase pointer type.

10.29.3 Member Function Documentation

10.29.3.1 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( ) [inline]
```

gets the interface of an enum alias node.

10.29.3.2 GetIntAlias()

```
IInteger* GetIntAlias ( ) [inline]
```

gets the interface of an integer alias node.

10.29.3.3 operator=()

```
void operator= (
    IBase * pB ) [inline]
```

Assign IBase Pointer.

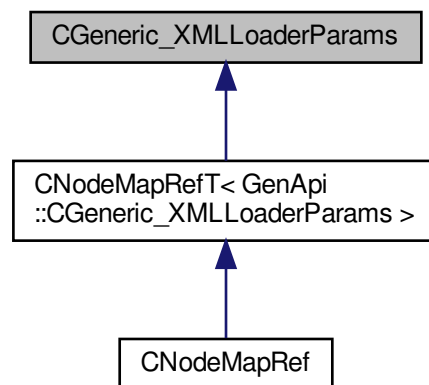
The documentation for this class was generated from the following file:

- include/SpinGenApi/Pointer.h

10.30 CGeneric_XMLLoaderParams Class Reference

Empty base class used by class [CNodeMapRef](#) as generic template argument.

Inheritance diagram for CGeneric_XMLLoaderParams:



Protected Member Functions

- virtual void [_Initialize](#) ([GenApi::INodeMap](#) *)

10.30.1 Detailed Description

Empty base class used by class [CNodeMapRef](#) as generic template argument.

10.30.2 Member Function Documentation

10.30.2.1 _Initialize()

```
virtual void _Initialize (
    GenApi::INodeMap * ) [inline], [protected], [virtual]
```

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

- include/SpinGenApi/NodeMapRef.h

10.31 CGlobalLock Class Reference

Named global lock which can be used over process boundaries.

Public Member Functions

- [CGlobalLock](#) (const char *pszName)
Creates a global lock object name pszName.
- [CGlobalLock](#) (const [gcstring](#) &strName)
Creates a global lock object name strName.
- [~CGlobalLock](#) ()
- bool [IsValid](#) (void) const
tests whether the lock is valid
- bool [Lock](#) (unsigned int timeout_ms)
enters the lock (may block)
- bool [TryLock](#) (void)
tries to enter the lock and returns immediately when not possible
- void [Unlock](#) (void)
leaves the lock

Protected Attributes

- long [m_DebugCount](#)

10.31.1 Detailed Description

Named global lock which can be used over process boundaries.

10.31.2 Constructor & Destructor Documentation

10.31.2.1 CGlobalLock() [1/2]

```
CGlobalLock (
    const char * pszName ) [explicit]
```

Creates a global lock object name pszName.

In case an object with the same name already exists a reference to the existing object will be created. If pszName is NULL an unnamed object will be created.

10.31.2.2 CGlobalLock() [2/2]

```
CGlobalLock (
    const gcstring & strName ) [explicit]
```

Creates a global lock object name strName.

In case an object with the same name already exists a reference to the existing object will be created. If strName is empty an unnamed object will be created.

10.31.2.3 ~CGlobalLock()

```
~CGlobalLock ( )
```

10.31.3 Member Function Documentation

10.31.3.1 IsValid()

```
bool IsValid (
    void ) const
```

tests whether the lock is valid

10.31.3.2 Lock()

```
bool Lock (
    unsigned int timeout_ms )
```

enters the lock (may block)

10.31.3.3 TryLock()

```
bool TryLock (
    void )
```

tries to enter the lock and returns immediately when not possible

10.31.3.4 Unlock()

```
void Unlock (
    void )
```

leaves the lock

10.31.4 Member Data Documentation

10.31.4.1 m_DebugCount

```
long m_DebugCount [mutable], [protected]
```

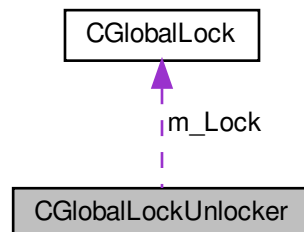
The documentation for this class was generated from the following file:

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

10.32 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](#)

10.32.1 Detailed Description

Unlocks the global lock object on destruction.

This is for automatic UNLOCKING only. We can't do automatic locking here since there is no returnvalue for constructors

10.32.2 Constructor & Destructor Documentation

10.32.2.1 CGlobalLockUnlocker()

```
CGlobalLockUnlocker (
    CGlobalLock & lock ) [inline]
```

10.32.2.2 ~CGlobalLockUnlocker()

```
~CGlobalLockUnlocker ( ) [inline]
```

10.32.3 Member Function Documentation

10.32.3.1 UnlockEarly()

```
void UnlockEarly (
    void ) [inline]
```

This function allows to unlock the object early before the object is destroyed.

10.32.4 Member Data Documentation

10.32.4.1 m_enabled

`bool m_enabled [protected]`

10.32.4.2 m_Lock

`CGlobalLock& m_Lock [protected]`

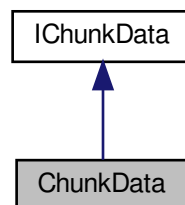
The documentation for this class was generated from the following file:

- `include/SpinGenApi/GCSynch.h`

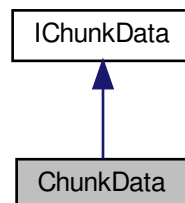
10.33 ChunkData Class Reference

The chunk data which contains additional information about an image.

Inheritance diagram for ChunkData:



Collaboration diagram for ChunkData:



Public Member Functions

- [ChunkData](#) ()
- [ChunkData](#) (const [ChunkData](#) &src)
- virtual [~ChunkData](#) (void)
- void [SetChunks](#) ([GenApi::INodeMap](#) &pNodeMap)
- [float64_t](#) [GetBlackLevel](#) () const
Description: Returns the black level used to capture the image.
- [int64_t](#) [GetFrameID](#) () const
Description: Returns the image frame ID.
- [float64_t](#) [GetExposureTime](#) () const
Description: Returns the exposure time used to capture the image.
- [int64_t](#) [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 GetInferenceResult () const`

Description: Visibility: Expert.

- `float64_t GetInferenceConfidence () const`

Description: Visibility: Expert.

Additional Inherited Members

10.33.1 Detailed Description

The chunk data which contains additional information about an image.

10.33.2 Constructor & Destructor Documentation

10.33.2.1 ChunkData() [1/2]

`ChunkData ()`

10.33.2.2 ChunkData() [2/2]

```
ChunkData (
    const ChunkData & src )
```

10.33.2.3 ~ChunkData()

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

10.33.3 Member Function Documentation

10.33.3.1 GetBlackLevel()

```
float64_t GetBlackLevel ( ) const [virtual]
```

Description: Returns the black level used to capture the image.

Visibility:

Implements [IChunkData](#).

10.33.3.2 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](#).

10.33.3.3 GetCRC()

```
int64_t GetCRC ( ) const [virtual]
```

Description: Returns the CRC of the image payload.

Visibility:

Implements [IChunkData](#).

10.33.3.4 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](#).

10.33.3.5 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](#).

10.33.3.6 GetExposureTime()

```
float64_t GetExposureTime ( ) const [virtual]
```

Description: Returns the exposure time used to capture the image.

Visibility:

Implements [IChunkData](#).

10.33.3.7 GetFrameID()

```
int64_t GetFrameID ( ) const [virtual]
```

Description: Returns the image frame ID.

Visibility:

Implements [IChunkData](#).

10.33.3.8 GetGain()

```
float64_t GetGain ( ) const [virtual]
```

Description: Returns the gain used to capture the image.

Visibility:

Implements [IChunkData](#).

10.33.3.9 GetHeight()

```
int64_t GetHeight ( ) const [virtual]
```

Description: Returns the height of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.10 GetImage()

```
int64_t GetImage ( ) const [virtual]
```

Description: Returns the image payload.

Visibility:

Implements [IChunkData](#).

10.33.3.11 GetInferenceConfidence()

```
float64_t GetInferenceConfidence ( ) const [virtual]
```

Description: Visibility: Expert.

Implements [IChunkData](#).

10.33.3.12 GetInferenceResult()

```
int64_t GetInferenceResult ( ) const [virtual]
```

Description: Visibility: Expert.

Implements [IChunkData](#).

10.33.3.13 GetLinePitch()

```
int64_t GetLinePitch ( ) const [virtual]
```

Description: Returns the LinePitch of the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.14 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](#).

10.33.3.15 GetOffsetX()

```
int64_t GetOffsetX ( ) const [virtual]
```

Description: Returns the Offset X of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.16 GetOffsetY()

```
int64_t GetOffsetY ( ) const [virtual]
```

Description: Returns the Offset Y of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.17 GetPartSelector()

```
int64_t GetPartSelector ( ) const [virtual]
```

Description: Selects the part to access in chunk data in a multipart transmission.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.18 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](#).

10.33.3.19 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](#).

10.33.3.20 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](#).

10.33.3.21 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](#).

10.33.3.22 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](#).

10.33.3.23 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](#).

10.33.3.24 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](#).

10.33.3.25 GetScan3dInvalidDataValue()

```
float64_t GetScan3dInvalidDataValue ( ) const [virtual]
```

Description: Returns the Invalid Data Value used for the image included in the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.26 GetScan3dTransformValue()

```
float64_t GetScan3dTransformValue ( ) const [virtual]
```

Description: Returns the transform value.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.27 GetScanLineSelector()

```
int64_t GetScanLineSelector ( ) const [virtual]
```

Description: Index for vector representation of one chunk value per line in an image.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.28 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](#).

10.33.3.29 GetSerialDataLength()

```
int64_t GetSerialDataLength ( ) const [virtual]
```

Description: Returns the length of the received serial data that was included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.30 GetStreamChannelID()

```
int64_t GetStreamChannelID ( ) const [virtual]
```

Description: Returns identifier of the stream channel used to carry the block.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.31 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](#).

10.33.3.32 GetTimestamp()

```
int64_t GetTimestamp ( ) const [virtual]
```

Description: Returns the Timestamp of the image.

Visibility:

Implements [IChunkData](#).

10.33.3.33 GetTimestampLatchValue()

```
int64_t GetTimestampLatchValue ( ) const [virtual]
```

Description: Returns the last Timestamp latched with the TimestampLatch command.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.34 GetTransferBlockID()

```
int64_t GetTransferBlockID ( ) const [virtual]
```

Description: Returns the unique identifier of the transfer block used to transport the payload.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.35 GetTransferQueueCurrentBlockCount()

```
int64_t GetTransferQueueCurrentBlockCount ( ) const [virtual]
```

Description: Returns the current number of blocks in the transfer queue.

Visibility: Expert

Implements [IChunkData](#).

10.33.3.36 GetWidth()

```
int64_t GetWidth ( ) const [virtual]
```

Description: Returns the width of the image included in the payload.

Visibility:

Implements [IChunkData](#).

10.33.3.37 SetChunks()

```
void SetChunks (
    GenApi::INodeMap & pNodeMap ) [virtual]
```

Implements [IChunkData](#).

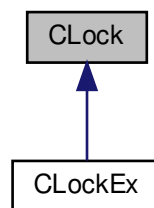
The documentation for this class was generated from the following file:

- include/[ChunkData.h](#)

10.34 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](#)

10.34.1 Detailed Description

A lock class.

10.34.2 Constructor & Destructor Documentation

10.34.2.1 CLock() [1/2]

```
CLock ( )
```

Constructor.

10.34.2.2 CLock() [2/2]

```
CLock (
    void * pLock )
```

Constructor.

10.34.2.3 ~CLock()

```
~CLock ( )
```

Destructor.

10.34.3 Member Function Documentation

10.34.3.1 Lock()

```
void Lock ( )
```

enters the critical section (may block)

10.34.3.2 TryLock()

```
bool TryLock ( )
```

tries to enter the critical section; returns true if successful

10.34.3.3 Unlock()

```
void Unlock ( )
```

leaves the critical section

10.34.4 Friends And Related Function Documentation

10.34.4.1 NodeMap

```
friend class NodeMap [friend]
```

10.34.5 Member Data Documentation

10.34.5.1 m_bOwnLock

```
bool m_bOwnLock [protected]
```

10.34.5.2 m_lock

```
void* m_lock [protected]
```

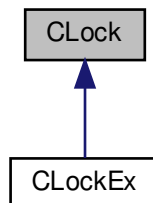
The documentation for this class was generated from the following file:

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

10.35 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

10.35.1 Detailed Description

A lock class.

10.35.2 Constructor & Destructor Documentation

10.35.2.1 CLock()

[CLock](#) ()

Constructor.

10.35.2.2 ~CLOCK()

`~CLOCK ()`

Destructor.

10.35.3 Member Function Documentation

10.35.3.1 Lock()

`void Lock ()`

enters the critical section (may block)

10.35.3.2 TryLock()

`bool TryLock ()`

tries to enter the critical section; returns true if successful

10.35.3.3 Unlock()

`void Unlock ()`

leaves the critical section

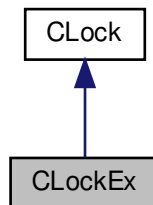
The documentation for this class was generated from the following file:

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

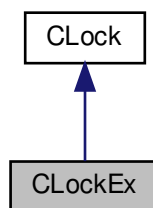
10.36 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Protected Attributes

- void * [m_lockEx](#)

Additional Inherited Members

10.36.1 Detailed Description

This class is for testing purposes only.

It should not be used for client code because it exists only for Windows but not for Linux since it uses internal data structures of a Win32 object

10.36.2 Member Data Documentation

10.36.2.1 m_lockEx

```
void* m_lockEx [protected]
```

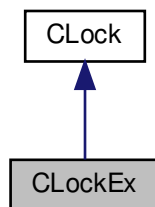
The documentation for this class was generated from the following file:

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

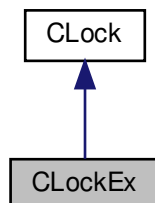
10.37 CLockEx Class Reference

This class is for testing purposes only.

Inheritance diagram for CLockEx:



Collaboration diagram for CLockEx:



Additional Inherited Members

10.37.1 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

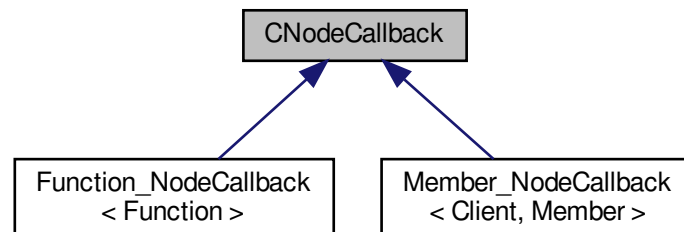
The documentation for this class was generated from the following file:

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

10.38 CNodeCallback Class Reference

callback body instance for INode pointers

Inheritance diagram for CNodeCallback:



Public Member Functions

- [CNodeCallback](#) ([INode](#) *pNode, [ECallbackType](#) CallbackType)
- virtual [~CNodeCallback](#) ()
virtual destructor
- virtual void [operator\(\)](#) ([ECallbackType](#) CallbackType) const =0
fires the callback if th type is right
- virtual void [Destroy](#) ()=0
destroys the object
- [INode](#) * [GetNode](#) ()
returns the node the callback is registered to
- [ECallbackType](#) [GetCallbackType](#) ()

Protected Attributes

- [INode * m_pNode](#)
the node were the callback is installed
- [ECallbackType m_CallbackType](#)
the type of the callback

10.38.1 Detailed Description

callback body instance for INode pointers

10.38.2 Constructor & Destructor Documentation

10.38.2.1 CNodeCallback()

```
CNodeCallback (
    INode * pNode,
    ECallbackType CallbackType ) [inline]
```

10.38.2.2 ~CNodeCallback()

```
virtual ~CNodeCallback ( ) [inline], [virtual]
```

virtual destructor

10.38.3 Member Function Documentation

10.38.3.1 Destroy()

```
virtual void Destroy ( ) [pure virtual]
```

destroys the object

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

10.38.3.2 GetCallbackType()

```
ECallbackType GetCallbackType ( ) [inline]
```

10.38.3.3 GetNode()

```
INode* GetNode ( ) [inline]
```

returns the node the callback is registered to

10.38.3.4 operator()

```
virtual void operator() (
    ECallbackType CallbackType ) const [pure virtual]
```

fires the callback if th type is right

Implemented in [Member_NodeCallback< Client, Member >](#), and [Function_NodeCallback< Function >](#).

10.38.4 Member Data Documentation

10.38.4.1 m_CallbackType

```
ECallbackType m_CallbackType [protected]
```

the type of the callback

10.38.4.2 m_pNode

```
INode* m_pNode [protected]
```

the node were the callback is installed

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

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

10.39 CNodeMapFactory Class Reference

The node map factory is used for creating node maps from camera description files.

Classes

- struct [NodeStatistics_t](#)

Public Member Functions

- [CNodeMapFactory](#) ()
Creates an empty node map factory for assigning a non-empty node map factory later.
- virtual [~CNodeMapFactory](#) ()
Destroys the node map factory data if all references to the data have been released.
- [CNodeMapFactory](#) (const [CNodeMapFactory](#) &)
Creates another reference to the node map factory data.
- [CNodeMapFactory](#) & [operator=](#) (const [CNodeMapFactory](#) &)
Creates another reference to the assigned node map factory data.
- [CNodeMapFactory](#) ([EContentType_t](#) FileType, const [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 [CNode↵](#) [DataMap](#) (not part of the public interface).
- bool [IsPreprocessed](#) () const
Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.
- void [ReleaseCameraDescriptionFileData](#) ()
Advanced: Releases any in constructors provided camera description file data buffers or files.
- bool [IsCameraDescriptionFileDataReleased](#) () const
Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.
- [INodeMap](#) * [CreateNodeMap](#) (const [GenlCam::gcstring](#) &DeviceName="Device", bool DoReleaseCamera↵ DescriptionFileData=true)
Creates a node map from the preprocessed memory internal representation of the camera description file(s).

- **INodeMap * CreateNodeMap** (CLOCK &UserProvidedLock, const **GenlCam::gcstring** &DeviceName="Device", bool DoReleaseCameraDescriptionFileData=true)
Creates a node map from the preprocessed memory internal representation of the camera description file(s).
- void **GetSupportedSchemaVersions** (**GenlCam::gcstring_vector** &SchemaVersions) const
- **GenlCam::gcstring ToString** () const
Outputs the pre-processed node map in string form (for debug purpose)
- **GenlCam::gcstring ToXml** () const
Outputs the pre-processed node map in XML form (mainly for debug purpose)
- void **GetNodeStatistics** (**NodeStatistics_t** &NodeStatistics)
- const **GenlCam::gcstring ApplyStyleSheet** (const **GenlCam::gcstring** &StyleSheetFileName)
Applies a style sheet to the pre-processed node map.

Static Public Member Functions

- static **INodeMap * CreateEmptyNodeMap** ()
Creates an empty node map usable as placeholder, e.g.
- static bool **ClearCache** ()
Deletes all preprocessed camera description files from the cache.
- static **CNodeDataMap * CreateNodeDataFromNodeMap** (**INodeMap** *pNodeMap)

10.39.1 Detailed Description

The node map factory is used for creating node maps from camera description files.

Examples

```
// Simple node map creation from buffer, downloaded from a device for instance.
CNodeMapFactory cameraNodeMapFactory( ContentType_ZippedXml, buffer,
    bufferSize);

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation with injecting additional xml fragments.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// Node map creation and additional extraction of a category subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
// Extract a subtree for later chunk parsing.
CNodeMapFactory chunkDataNodeMapFactory = cameraParameters.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraParameters.CreateNodeMap();
// The next step is attaching the device port (not shown).
```

```
// Node map creation with injecting additional xml fragments and additional extraction of a category
subtree.
CNodeMapFactory cameraNodeMapFactory( ContentType_Xml, buffer, bufferSize);
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename1));
cameraParameters.AddInjectionData( CNodeMapFactory(
    ContentType_Xml, filename2));
CNodeMapFactory chunkDataNodeMapFactory = cameraNodeMapFactory.ExtractSubtree("ChunkData");

// Create the node map. The node map can be destroyed using the IDestroy interface later.
INodeMap* pNodeMap = cameraNodeMapFactory.CreateNodeMap();
// The next step is attaching the device port (not shown).

// A node map factory can create multiple node maps from the provided camera description file(s).
for(int i = 0; i < 20; ++i)
{
    INodeMap* pNodeMapChunks = chunkDataNodeMapFactory.CreateNodeMap();
    //...
}
```

Attention

The is [CNodeMapFactory](#) not thread-safe.

You need to take care when camera description file data can be actually be freed, see method documentation of the node map factory for more detail.

10.39.2 Constructor & Destructor Documentation

10.39.2.1 CNodeMapFactory() [1/5]

```
CNodeMapFactory ( )
```

Creates an empty node map factory for assigning a non-empty node map factory later.

10.39.2.2 ~CNodeMapFactory()

```
virtual ~CNodeMapFactory ( ) [virtual]
```

Destroys the node map factory data if all references to the data have been released.

10.39.2.3 CNodeMapFactory() [2/5]

```
CNodeMapFactory (
    const CNodeMapFactory & )
```

Creates another reference to the node map factory data.

No data is copied.

10.39.2.4 CNodeMapFactory() [3/5]

```
CNodeMapFactory (
    EContentType_t FileType,
    const GenICam::gcstring & FileName,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the full path to the provided camera description file data.

Parameters

| | | |
|----|------------------------------|---|
| in | <i>FileType</i> | Defines how the camera description file is stored, e.g. as zipped XML text. |
| in | <i>FileName</i> | The full path of the camera description file to process. |
| in | <i>CacheUsage</i> | Defines if and how to use the cache for preprocessed camera description files. |
| in | <i>SuppressStringsOnLoad</i> | Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint. |

Throws an invalid argument exception if *FileName* is empty. Throws if environment variables in *FileName* cannot be resolved.

Attention

The given file must be readable until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

10.39.2.5 CNodeMapFactory() [4/5]

```
CNodeMapFactory (
    EContentType_t ContentType,
    const void * pData,
    size_t DataSize,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and simply stores the pointer and the size of the provided camera description file data.

Parameters

| | | |
|----|------------------------------|---|
| in | <i>ContentType</i> | Defines how the camera description file is stored, e.g. as zipped XML text. |
| in | <i>pData</i> | The pointer to the camera description file data. |
| in | <i>DataSize</i> | The size of the camera description file data. |
| in | <i>CacheUsage</i> | Defines if and how to use the cache for preprocessed camera description files. |
| in | <i>SuppressStringsOnLoad</i> | Suppresses loading strings that are not needed for most use cases, e.g. node tooltip or description, for reducing the memory footprint. |

Throws an invalid argument exception if *pData* is NULL or *DataSize* is 0.

Attention

The given buffer must not be freed or changed until the camera description file data has been released. The [IsCameraDescriptionFileDataReleased\(\)](#) method can be used to check if releasing has been done.

10.39.2.6 CNodeMapFactory() [5/5]

```
CNodeMapFactory (
    const GenICam::gcstring & XmlData,
    ECacheUsage_t CacheUsage = CacheUsage_Automatic,
    bool SuppressStringsOnLoad = false )
```

Creates the node map factory and copies the provided camera description file string.

Parameters

| | | |
|----|------------------------------|--|
| in | <i>XmlData</i> | The camera description file data as XML text. The provided text is copied. You can use the overloaded constructor accepting a buffer to avoid that. <code>gcstring cdfData; //... fill cdfData ... CNodeMapFactory factory(ContentType_Xml, cdfData.c_str(), cdfData.size()); // Create the node map. The node map can be destroyed using the IDestroy interface later. INodeMap* pNodeMap = factory.CreateNodeMap(); // The next step is attaching the device port (not shown).</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.

10.39.3 Member Function Documentation

10.39.3.1 AddInjectionData()

```
void AddInjectionData (
    CNodeMapFactory & injectionData )
```

Adds a node map factory representing a camera description file to inject.

Parameters

| | | |
|----|----------------------|--|
| in | <i>injectionData</i> | A node map factory representing a camera description file to inject. |
|----|----------------------|--|

The injected files are injected in the order they are added. InjectionData must not be preprocessed. The [IsPreprocessed\(\)](#) method can be used to check if preprocessing has been done before. The cache usage of injection data is ignored.

10.39.3.2 ApplyStyleSheet()

```
const GenICam::gcstring ApplyStyleSheet (
    const GenICam::gcstring & StyleSheetFileName )
```

Applies a style sheet to the pre-processed node map.

10.39.3.3 ClearCache()

```
static bool ClearCache ( ) [static]
```

Deletes all preprocessed camera description files from the cache.

10.39.3.4 CreateEmptyNodeMap()

```
static INodeMap* CreateEmptyNodeMap ( ) [static]
```

Creates an empty node map usable as placeholder, e.g.

if certain features are not supported by a module.

10.39.3.5 CreateNodeDataFromNodeMap()

```
static CNodeDataMap* CreateNodeDataFromNodeMap (
    INodeMap * pNodeMap ) [static]
```

10.39.3.6 CreateNodeMap() [1/2]

```
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.

10.39.3.7 CreateNodeMap() [2/2]

```
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.

10.39.3.8 ExtractSubtree()

```
CNodeMapFactory ExtractSubtree (
    const GenICam::gcstring & SubTreeRootNodeName,
    bool doRenameToRoot = false )
```

The name of the node that represents the root of the subtree that shall be extracted.

Parameters

| | | |
|----|----------------------------|--|
| in | <i>SubTreeRootNodeName</i> | The root of the branch to extract, e.g. "ChunkData". |
| in | <i>doRenameToRoot</i> | Renames the extracted subtree root node SubTreeRootNodeName to "Root", sets the IsFeature property. Preprocess() is automatically called if needed to create the memory internal representation of the camera description file(s). The preprocessed result can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. |

10.39.3.9 GetNodeStatistics()

```
void GetNodeStatistics (
    NodeStatistics_t & NodeStatistics )
```

10.39.3.10 GetSupportedSchemaVersions()

```
void GetSupportedSchemaVersions (
    GenICam::gcstring_vector & SchemaVersions ) const
```

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

10.39.3.11 IsCameraDescriptionFileDataReleased()

```
bool IsCameraDescriptionFileDataReleased ( ) const
```

Can be used to check whether the [ReleaseCameraDescriptionFileData\(\)](#) processing step has been performed.

10.39.3.12 IsEmpty()

```
bool IsEmpty ( ) const
```

Returns true if nothing is loaded ([IsLoaded\(\)](#)) and no source data is available, e.g.

when the node map factory has been created with the default constructor.

10.39.3.13 IsLoaded()

```
bool IsLoaded ( ) const
```

Can be used to check whether the [LoadAndInject\(\)](#) processing step has been performed.

Returns true if [IsPreprocessed\(\)](#) returns true (Preprocessed Data has been loaded from cache).

10.39.3.14 IsPreprocessed()

```
bool IsPreprocessed ( ) const
```

Can be used to check whether the [Preprocess\(\)](#) processing step has been performed.

10.39.3.15 LoadAndInject()

```
void LoadAndInject ( )
```

Advanced: Loads, Parses, and Injects the camera description files recursively.

The result is a memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Prevents cache read if called manually.

10.39.3.16 operator=()

```
CNodeMapFactory& operator= (
    const CNodeMapFactory & )
```

Creates another reference to the assigned node map factory data.

Destroys the "overwritten" node map factory data if all references to the data have been released.

10.39.3.17 Preprocess()

```
void Preprocess ( )
```

Advanced: Creates the preprocessed memory internal representation of the camera description file(s), the CNodeDataMap (not part of the public interface).

This step is usually done automatically. Preprocessed data can be read from the cache or written to the cache in this step. This depends on the availability of a cache and the used CacheUsage setting. By calling this method directly direct cache load is suppressed, see [CreateNodeMap\(\)](#) for more information.

10.39.3.18 ReleaseCameraDescriptionFileData()

```
void ReleaseCameraDescriptionFileData ( )
```

Advanced: Releases any in constructors provided camera description file data buffers or files.

This step is usually done automatically. All references to added injection data are dropped in this step to free the data. After this step any in constructors provided buffers can be freed or any in constructors given files can be deleted.

10.39.3.19 ToString()

```
GenICam::gcstring ToString ( ) const
```

Outputs the pre-processed node map in string form (for debug purpose)

10.39.3.20 ToXml()

```
GenICam::gcstring ToXml ( ) const
```

Outputs the pre-processed node map in XML form (mainly for debug purpose)

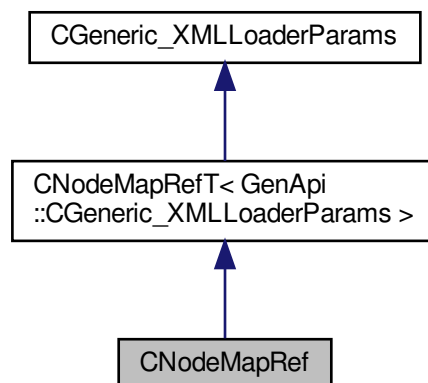
The documentation for this class was generated from the following file:

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

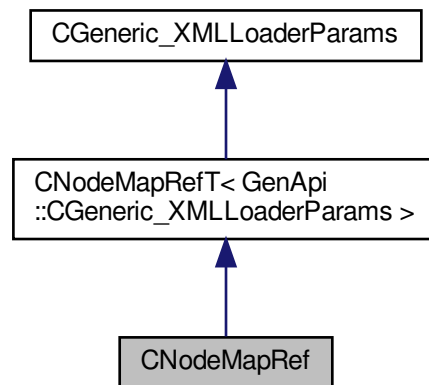
10.40 CNodeMapRef Class Reference

Smartpointer for NodeMaps with create function.

Inheritance diagram for CNodeMapRef:



Collaboration diagram for CNodeMapRef:



Public Member Functions

- **CNodeMapRef** (const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (INodeMap *pNodeMap, const [GenICam::gcstring](#) &DeviceName="Device")
Constructor.
- **CNodeMapRef** (const **CNodeMapRef** &Them)
Copy constructor.
- **CNodeMapRef** & **operator=** (const **CNodeMapRef** &Them)
Assignment.
- **CNodeMapRef** & **operator=** (INodeMap *pNodeMap)
Assignment of an INodeMap.*

Additional Inherited Members

10.40.1 Detailed Description

Smartpointer for NodeMaps with create function.

Note

This class is a simple typedef definition. The class syntax is only used, because Doxygen has to generate a useful documentation.

10.40.2 Constructor & Destructor Documentation

10.40.2.1 CNodeMapRef() [1/3]

```
CNodeMapRef (
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

10.40.2.2 CNodeMapRef() [2/3]

```
CNodeMapRef (
    INodeMap * pNodeMap,
    const GenICam::gcstring & DeviceName = "Device" ) [inline]
```

Constructor.

10.40.2.3 CNodeMapRef() [3/3]

```
CNodeMapRef (
    const CNodeMapRef & Them ) [inline]
```

Copy constructor.

10.40.3 Member Function Documentation

10.40.3.1 operator=() [1/2]

```
CNodeMapRef& operator= (
    const CNodeMapRef & Them ) [inline]
```

Assignment.

10.40.3.2 operator=() [2/2]

```
CNodeMapRef& operator= (
    INodeMap * pNodeMap ) [inline]
```

Assignment of an INodeMap*.

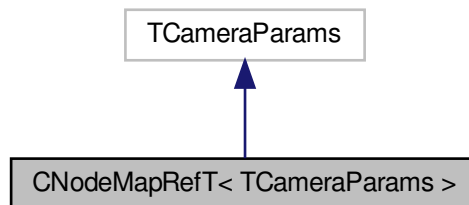
The documentation for this class was generated from the following file:

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

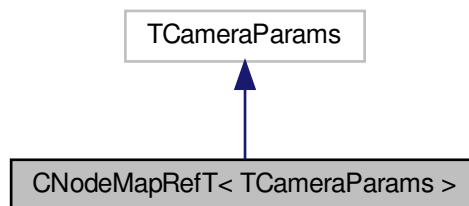
10.41 CNodeMapRefT< TCameraParams > Class Template Reference

Smartpointer template for NodeMaps with create function.

Inheritance diagram for CNodeMapRefT< TCameraParams >:



Collaboration diagram for CNodeMapRefT< TCameraParams >:



Public Member Functions

- `CNodeMapRefT` (const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (`INodeMap` *pNodeMap, const `GenICam::gcstring` &DeviceName="Device")
Constructor.
- `CNodeMapRefT` (const `CNodeMapRefT` &Them)
Copy constructor.
- `CNodeMapRefT` & `operator=` (const `CNodeMapRefT` &Them)
Assignment.
- `CNodeMapRefT` & `operator=` (`INodeMap` *pNodeMap)
Assignment of an INodeMap.*
- virtual `~CNodeMapRefT` ()
Destructor.
- void `_Destroy` ()

- Destroys the node map.*
- void `_LoadXMLFromFile` (const `GenICam::gcstring` &FileName)
Creates the object from a XML file with given file name.
- void `_LoadXMLFromZIPFile` (const `GenICam::gcstring` &ZipFileName)
Creates the object from a ZIP'd XML file with given file name.
- void `_LoadXMLFromZIPData` (const void *zipData, size_t zipSize)
Creates the object from a ZIP'd XML file given in a string.
- void `_LoadXMLFromFileInject` (const `GenICam::gcstring` &TargetFileName, const `GenICam::gcstring` &InjectFileName)
Creates the object from a XML target and an inject file with given file name.
- void `_LoadXMLFromString` (const `GenICam::gcstring` &XMLData)
Creates the object from XML data given in a string.
- void `_LoadXMLFromStringInject` (const `GenICam::gcstring` &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`.

10.41.1 Detailed Description

```
template<class TCameraParams>
class Spinnaker::GenApi::CNodeMapRefT< TCameraParams >
```

Smartpointer template for NodeMaps with create function.

Parameters

| | |
|----------------------|---|
| <i>TCameraParams</i> | The camera specific parameter class (auto generated from camera xml file) |
|----------------------|---|

10.41.2 Member Function Documentation

10.41.2.1 _ClearXMLCache()

```
static bool _ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

10.41.2.2 _Connect() [1/2]

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

Connects a port to a port node with given name.

10.41.2.3 _Connect() [2/2]

```
virtual bool _Connect (
    IPort * pPort ) [virtual]
```

Connects a port to the standard port "Device".

10.41.2.4 _GetDeviceName()

```
virtual GenICam::gcstring _GetDeviceName ( ) [virtual]
```

Get device name.

10.41.2.5 _GetNode()

```
virtual INode\* _GetNode (
    const GenICam::gcstring & key ) [virtual]
```

Retrieves the node from the central map by name.

10.41.2.6 _GetNodes()

```
virtual void _GetNodes (
    NodeList\_t & Nodes ) [virtual]
```

Retrieves all nodes in the node map.

10.41.2.7 _GetSupportedSchemaVersions()

```
virtual void _GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers. Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only.

10.41.2.8 _InvalidateNodes()

```
virtual void _InvalidateNodes ( ) [virtual]
```

Invalidates all nodes.

10.41.2.9 _LoadXMLFromFile()

```
void _LoadXMLFromFile (
    const GenICam::gcstring & FileName )
```

Creates the object from a XML file with given file name.

10.41.2.10 _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.

10.41.2.11 _LoadXMLFromString()

```
void _LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

10.41.2.12 _LoadXMLFromStringInject()

```
void _LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

10.41.2.13 _LoadXMLFromZIPData()

```
void _LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

10.41.2.14 _LoadXMLFromZIPFile()

```
void _LoadXMLFromZIPFile (
    const GenICam::gcstring & ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

10.41.2.15 _Poll()

```
virtual void _Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

10.41.3 Member Data Documentation

10.41.3.1 _Ptr

```
INodeMap* _Ptr
```

Pointer to the [NodeMap](#).

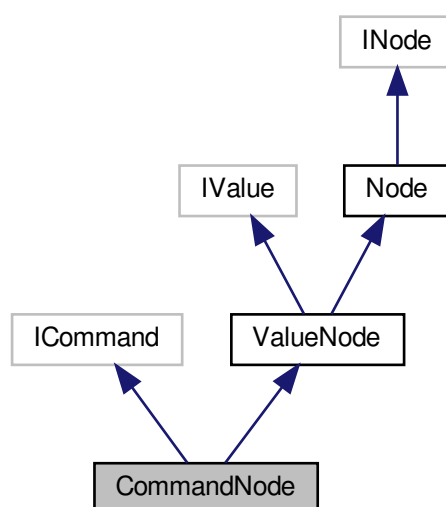
The documentation for this class was generated from the following file:

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

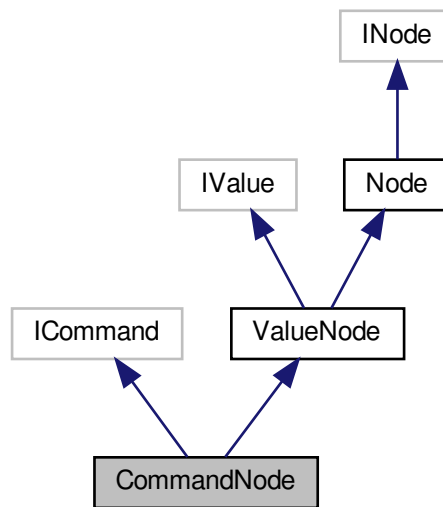
10.42 CommandNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for CommandNode:



Collaboration diagram for CommandNode:



Public Member Functions

- [CommandNode](#) ()
- [CommandNode](#) (std::shared_ptr< Node::NodeImpl > pCommand)
- virtual [~CommandNode](#) ()
- virtual void [Execute](#) (bool [Verify](#)=true)
Execute the command.
- virtual void [operator\(\)](#) ()
Execute the command.
- virtual bool [IsDone](#) (bool [Verify](#)=true)
Query whether the command is executed.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value

Additional Inherited Members

10.42.1 Detailed Description

[Interface](#) for string properties.

10.42.2 Constructor & Destructor Documentation

10.42.2.1 CommandNode() [1/2]

```
CommandNode ( )
```

10.42.2.2 CommandNode() [2/2]

```
CommandNode (
    std::shared_ptr< Node::NodeImpl > pCommand )
```

10.42.2.3 ~CommandNode()

```
virtual ~CommandNode ( ) [virtual]
```

10.42.3 Member Function Documentation

10.42.3.1 Execute()

```
virtual void Execute (
    bool Verify = true ) [virtual]
```

Execute the command.

Parameters

| | |
|---------------|--|
| <i>Verify</i> | Enables AccessMode and Range verification (default = true) |
|---------------|--|

10.42.3.2 IsDone()

```
virtual bool IsDone (
    bool Verify = true ) [virtual]
```

Query whether the command is executed.

Parameters

| | |
|---------------|--|
| <i>Verify</i> | Enables Range verification (default = false). The AccessMode is always checked |
|---------------|--|

Returns

True if the Execute command has finished; false otherwise

10.42.3.3 operator()

```
virtual void operator() ( ) [virtual]
```

Execute the command.

10.42.3.4 SetReference()

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

overload SetReference for Value

Reimplemented from [ValueNode](#).

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

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

10.43 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](#) ()

10.43.1 Detailed Description

Definition of a simple [Counter](#) class.

10.43.2 Constructor & Destructor Documentation

10.43.2.1 Counter()

```
Counter ( ) [inline]
```

10.43.3 Member Function Documentation

10.43.3.1 GetValue()

```
unsigned int GetValue ( ) const [inline]
```

10.43.3.2 IsZero()

```
bool IsZero ( ) [inline]
```

10.43.3.3 operator unsigned int()

```
operator unsigned int ( ) [inline]
```

10.43.3.4 operator++() [1/2]

```
unsigned int operator++ ( ) [inline]
```

10.43.3.5 operator++() [2/2]

```
unsigned int operator++ (
    int ) [inline]
```

10.43.3.6 operator--() [1/2]

```
unsigned int operator-- (
    int ) [inline]
```

10.43.3.7 operator--() [2/2]

```
unsigned int operator-- ( ) [inline]
```

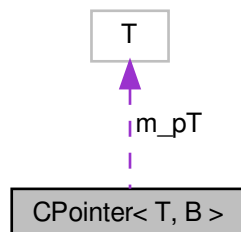
The documentation for this class was generated from the following file:

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

10.44 CPointer< T, B > Class Template Reference

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

Collaboration diagram for CPointer< T, B >:



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.

10.44.1 Detailed Description

```
template<class T, class B = IBase>
class Spinnaker::GenApi::CPointer< T, B >
```

Encapsulates a [GenApi](#) pointer dealing with the `dynamic_cast` automatically.

10.44.2 Constructor & Destructor Documentation

10.44.2.1 CPointer() [1/2]

```
CPointer (
    void ) [inline]
```

Default constructor.

10.44.2.2 CPointer() [2/2]

```
CPointer (
    B * pB ) [inline]
```

Constructor from INode pointer type.

10.44.2.3 ~CPointer()

```
virtual ~CPointer (
    void ) [inline], [virtual]
```

10.44.3 Member Function Documentation

10.44.3.1 IsValid()

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

true if the pointer is valid

10.44.3.2 operator bool()

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

true if the pointer is valid

10.44.3.3 operator T*()

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

Dereferencing.

10.44.3.4 operator!=() [1/5]

```
bool operator!= (
    const CPointer< T, B > & rT ) const [inline]
```

pointer unequal

10.44.3.5 operator!=() [2/5]

```
bool operator!= (
    T * pT ) const [inline]
```

pointer unequal

10.44.3.6 operator!=() [3/5]

```
bool operator!= (
    const long int nMustBeNull ) const [inline]
```

pointer unequal

10.44.3.7 operator!=() [4/5]

```
bool operator!= (
    const int nMustBeNull ) const [inline]
```

pointer unequal

10.44.3.8 operator!=() [5/5]

```
bool operator!= (
    const std::nullptr_t nullptr ) const [inline]
```

pointer unequal

10.44.3.9 operator()

```
T& operator() (
    void ) const [inline]
```

Dereferencing.

10.44.3.10 operator*()

```
T& operator* (
    void ) const [inline]
```

Dereferencing.

10.44.3.11 operator->()

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

Dereferencing.

10.44.3.12 operator=()

```
void operator= (
    B * pB ) [inline]
```

Assign INode Pointer.

10.44.3.13 operator==() [1/3]

```
bool operator== (
    T * pT ) const [inline]
```

pointer equal

10.44.3.14 operator==() [2/3]

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

pointer equal

10.44.3.15 operator==() [3/3]

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

pointer equal

10.44.4 Member Data Documentation

10.44.4.1 m_pT

`T* m_pT` [protected]

Underlying raw pointer.

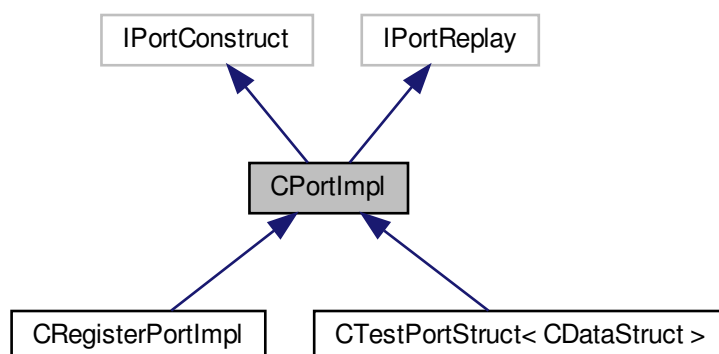
The documentation for this class was generated from the following file:

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

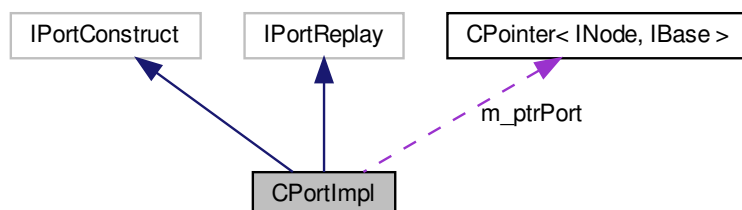
10.45 CPortImpl Class Reference

Standard implementation for a port.

Inheritance diagram for CPortImpl:



Collaboration diagram for CPortImpl:



Public Member Functions

- [CPortImpl](#) ()
Constructor.
- virtual [~CPortImpl](#) ()
Destructor.
- virtual [EAccessMode GetAccessMode](#) () const =0
Get the access mode of the node.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))=0
Writes a chunk of bytes to the port.
- virtual void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual [EYesNo GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual void [Replay](#) (IPortWriteList *pPortRecorder, bool [Invalidate](#)=true)
sends the commands to the camera.
- void [InvalidateNode](#) ()

Protected Attributes

- [CNodePtr m_ptrPort](#)
Pointer to the node holding a reference to this implementation.

10.45.1 Detailed Description

Standard implementation for a port.

10.45.2 Constructor & Destructor Documentation

10.45.2.1 CPortImpl()

```
CPortImpl ( ) [inline]
```

Constructor.

10.45.2.2 ~CPortImpl()

```
virtual ~CPortImpl ( ) [inline], [virtual]
```

Destructor.

10.45.3 Member Function Documentation

10.45.3.1 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 >](#).

10.45.3.2 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [inline], [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.45.3.3 InvalidateNode()

```
void InvalidateNode ( ) [inline]
```

10.45.3.4 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 >](#).

10.45.3.5 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.

10.45.3.6 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented in [CRegisterPortImpl](#).

10.45.3.7 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 >](#).

10.45.4 Member Data Documentation

10.45.4.1 m_ptrPort

```
CNodePtr m_ptrPort [protected]
```

Pointer to the node holding a reference to this implementation.

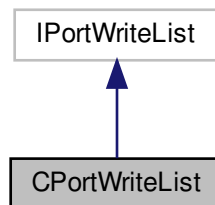
The documentation for this class was generated from the following file:

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

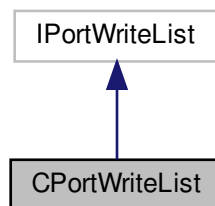
10.46 CPortWriteList Class Reference

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](#)

10.46.1 Detailed Description

Container holding a list of port write commands.

10.46.2 Constructor & Destructor Documentation

10.46.2.1 CPortWriteList()

```
CPortWriteList ( )
```

Constructor.

10.46.2.2 ~CPortWriteList()

```
~CPortWriteList ( )
```

Destructor.

10.46.3 Member Function Documentation

10.46.3.1 GetCookie()

```
virtual int64_t GetCookie ( ) [virtual]
```

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

10.46.3.2 GetPortWriteListHandle()

```
void* GetPortWriteListHandle ( )
```

10.46.3.3 Replay()

```
virtual void Replay (
    IPort * pPort ) [virtual]
```

Replays the write command to the given port interface.

10.46.3.4 SetCookie()

```
virtual void SetCookie (
    const int64_t Value ) [virtual]
```

Sets a cookie in case the port implementation want to cache a command list.

10.46.3.5 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

10.46.4 Member Data Documentation

10.46.4.1 m_pWriteList

```
void* m_pWriteList [protected]
```

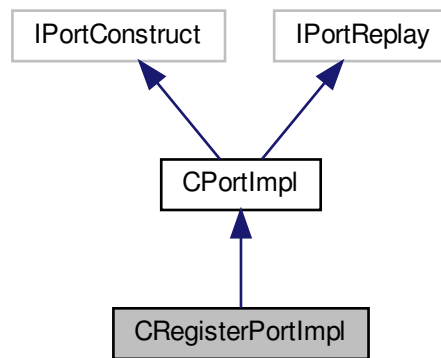
The documentation for this class was generated from the following file:

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

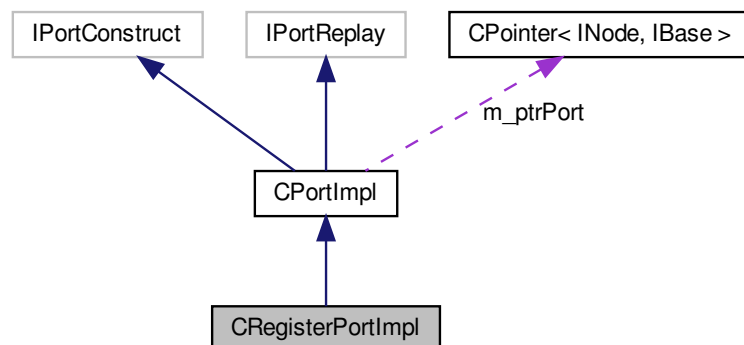
10.47 CRegisterPortImpl Class Reference

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

10.47.1 Detailed Description

Standard implementation for a port using a register based transport layer.

10.47.2 Constructor & Destructor Documentation

10.47.2.1 CRegisterPortImpl()

```
CRegisterPortImpl (
    int MaxNumQuadlets = 1,
    bool TransportLayerSwapsEndianness = false ) [inline]
```

Constructor.

10.47.2.2 ~CRegisterPortImpl()

```
virtual ~CRegisterPortImpl ( ) [inline], [virtual]
```

Destructor.

10.47.3 Member Function Documentation

10.47.3.1 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](#).

10.47.3.2 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.47.3.3 ReadRegister()

```
virtual void ReadRegister (
    uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Reads an array of quadlets from the port.

10.47.3.4 SetPortImpl()

```
virtual void SetPortImpl (
    IPort * pPort ) [inline], [virtual]
```

Sets pointer the real port implementation; this function may called only once.

Reimplemented from [CPortImpl](#).

10.47.3.5 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](#).

10.47.3.6 WriteRegister()

```
virtual void WriteRegister (
    const uint32_t * pRegisters,
    int64_t Address,
    int64_t Length ) [pure virtual]
```

Writes an array of quadlets to the port.

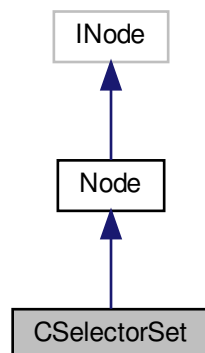
The documentation for this class was generated from the following file:

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

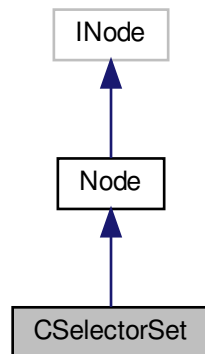
10.48 CSelectorSet Class Reference

The set of selectors selecting a given node.

Inheritance diagram for CSelectorSet:



Collaboration diagram for CSelectorSet:



Public Member Functions

- [CSelectorSet](#) ([IBase](#) *pBase)
Constructor.
- [~CSelectorSet](#) ()
Destructor.
- bool [IsEmpty](#) ()
returns true if no selectors are present
- virtual bool [SetFirst](#) ()
- virtual bool [SetNext](#) (bool Tick=true)
- virtual void [Restore](#) ()
- virtual [GenICam::gcstring ToString](#) ()
- virtual void [GetSelectorList](#) (FeatureList_t &SelectorList, bool Incremental=false)

Additional Inherited Members

10.48.1 Detailed Description

The set of selectors selecting a given node.

10.48.2 Constructor & Destructor Documentation

10.48.2.1 CSelectorSet()

```

CSelectorSet (
    IBase * pBase )
  
```

Constructor.

Parameters

| | |
|--------------|--------------------------------------|
| <i>pBase</i> | Feature selected by the selector set |
|--------------|--------------------------------------|

10.48.2.2 ~CSelectorSet()

```
~CSelectorSet ( )
```

Destructor.

10.48.3 Member Function Documentation

10.48.3.1 GetSelectorList()

```
virtual void GetSelectorList (
    FeatureList_t & SelectorList,
    bool Incremental = false ) [virtual]
```

10.48.3.2 IsEmpty()

```
bool IsEmpty ( )
```

returns true if no selectors are present

10.48.3.3 Restore()

```
virtual void Restore ( ) [virtual]
```

10.48.3.4 SetFirst()

```
virtual bool SetFirst ( ) [virtual]
```

10.48.3.5 SetNext()

```
virtual bool SetNext (
    bool Tick = true ) [virtual]
```

10.48.3.6 ToString()

```
virtual GenICam::gcstring ToString ( ) [virtual]
```

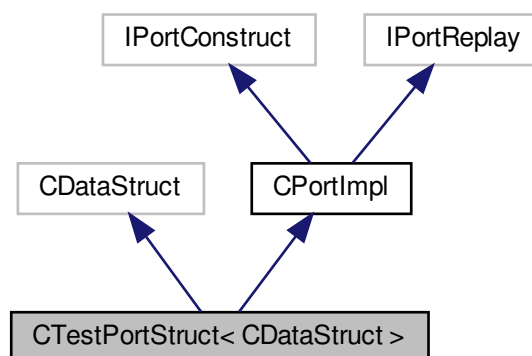
The documentation for this class was generated from the following file:

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

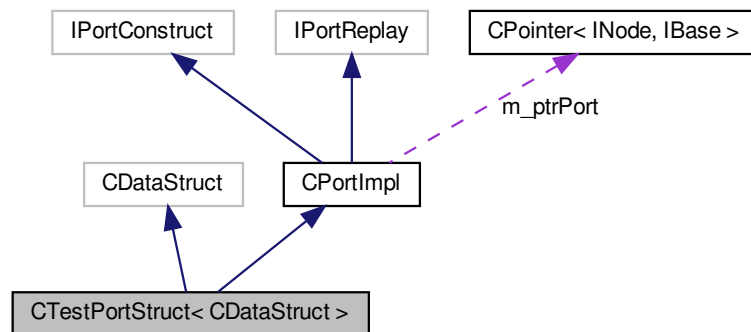
10.49 CTestPortStruct< CDataStruct > Class Template Reference

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

10.49.1 Detailed Description

```
template<class CDataStruct>
class Spinnaker::GenApi::CTestPortStruct< CDataStruct >
```

Implements a register spaces based on a C++ struct.

10.49.2 Constructor & Destructor Documentation

10.49.2.1 CTestPortStruct()

```
CTestPortStruct (
    int64_t BaseAddress = 0 ) [inline]
```

10.49.3 Member Function Documentation

10.49.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [inline], [virtual]
```

Get the access mode of the node.

Implements [CPortImpl](#).

10.49.3.2 GetNumReads()

```
int64_t GetNumReads ( ) [inline]
```

Returns the number of reads since lastReset Statistics.

10.49.3.3 GetNumWrites()

```
int64_t GetNumWrites ( ) [inline]
```

Returns the number of writes since lastReset Statistics.

10.49.3.4 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [inline], [virtual]
```

Get the type of the main interface of a node.

10.49.3.5 MemSet()

```
void MemSet (
    const char FillValue ) [inline]
```

10.49.3.6 Read()

```
virtual void Read (
    void * pBuffer,
    int64_t Address,
    int64_t Length ) [inline], [virtual]
```

Reads a chunk of bytes from the port.

Implements [CPortImpl](#).

10.49.3.7 ResetStatistics()

```
void ResetStatistics ( ) [inline]
```

Resets the read/write statistics.

10.49.3.8 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](#).

10.49.4 Member Data Documentation

10.49.4.1 m_BaseAddress

```
int64_t m_BaseAddress [protected]
```

the base address used for the struct

10.49.4.2 m_NumReads

```
int64_t m_NumReads [protected]
```

Number of reads since last reset.

10.49.4.3 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](#)

10.50 DCAM_CHECKSUM Struct Reference

Public Attributes

- uint32_t [CRCChecksum](#)

10.50.1 Member Data Documentation

10.50.1.1 CRCChecksum

```
uint32_t CRCChecksum
```

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

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

10.51 DCAM_CHUNK_TRAILER Struct Reference

Public Attributes

- SPIN_GUID [ChunkID](#)
- uint32_t [ChunkLength](#)
- uint32_t [InverseChunkLength](#)

10.51.1 Member Data Documentation

10.51.1.1 ChunkID

SPIN_GUID ChunkID

10.51.1.2 ChunkLength

uint32_t ChunkLength

10.51.1.3 InverseChunkLength

uint32_t InverseChunkLength

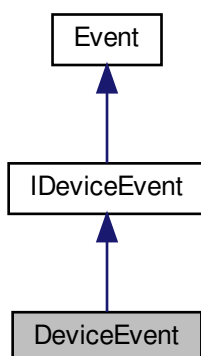
The documentation for this struct was generated from the following file:

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

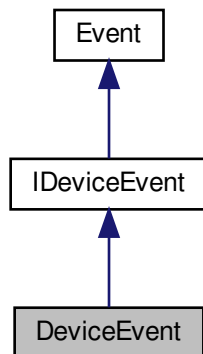
10.52 DeviceEvent Class Reference

A handler to device events.

Inheritance diagram for DeviceEvent:



Collaboration diagram for DeviceEvent:



Public Member Functions

- [DeviceEvent](#) ()
Default constructor.
- virtual [~DeviceEvent](#) ()
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

- [DeviceEvent](#) & [operator=](#) (const [DeviceEvent](#) &)
Assignment operator.

Additional Inherited Members

10.52.1 Detailed Description

A handler to device events.

10.52.2 Constructor & Destructor Documentation

10.52.2.1 DeviceEvent()

`DeviceEvent ()`

Default constructor.

10.52.2.2 ~DeviceEvent()

`virtual ~DeviceEvent () [virtual]`

Virtual destructor.

10.52.3 Member Function Documentation

10.52.3.1 GetDeviceEventId()

`uint64_t GetDeviceEventId () const [virtual]`

Get the ID of the device event.

Returns

The device event ID

Implements [IDeviceEvent](#).

10.52.3.2 GetDeviceEventName()

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

Get the name of the device event.

Returns

The device event name

Implements [IDeviceEvent](#).

10.52.3.3 OnDeviceEvent()

`virtual void OnDeviceEvent (
 Spinnaker::GenICam::gcstring eventName) [pure virtual]`

Device event callback.

Parameters

| | |
|------------------------|-----------------------|
| <code>eventName</code> | The name of the event |
|------------------------|-----------------------|

Implements [IDeviceEvent](#).

10.52.3.4 `operator=()`

```
DeviceEvent& operator= (
    const DeviceEvent & ) [protected]
```

Assignment operator.

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

- include/[DeviceEvent.h](#)

10.53 `double_autovector_t` Class Reference

Vector of doubles with reference counting.

Public Member Functions

- [double_autovector_t](#) ()
- [double_autovector_t](#) (const [double_autovector_t](#) &obj)
- [double_autovector_t](#) (size_t n)
- virtual [~double_autovector_t](#) (void)
- [double_autovector_t](#) & [operator=](#) (const [double_autovector_t](#) &obj)
- void [operator delete](#) (void *pWhere)
- void * [operator new](#) (size_t uiSize)
- double & [operator\[\]](#) (size_t uiIndex)
- const double & [operator\[\]](#) (size_t uiIndex) const
- size_t [size](#) () const

Protected Attributes

- std::vector< double > * [_pv](#)
- ATOMIC_VARIABLE * [_pCount](#)

10.53.1 Detailed Description

Vector of doubles with reference counting.

10.53.2 Constructor & Destructor Documentation

10.53.2.1 `double_autovector_t()` [1/3]

```
double_autovector_t ( )
```

10.53.2.2 `double_autovector_t()` [2/3]

```
double_autovector_t (
    const double_autovector_t & obj )
```

10.53.2.3 `double_autovector_t()` [3/3]

```
double_autovector_t (
    size_t n ) [explicit]
```

10.53.2.4 `~double_autovector_t()`

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

10.53.3 Member Function Documentation

10.53.3.1 `operator delete()`

```
void operator delete (
    void * pWhere )
```

10.53.3.2 `operator new()`

```
void* operator new (
    size_t uiSize )
```

10.53.3.3 `operator=()`

```
double_autovector_t& operator= (
    const double_autovector_t & obj )
```

10.53.3.4 `operator[]()` [1/2]

```
double& operator[] (
    size_t uiIndex )
```

10.53.3.5 `operator[]()` [2/2]

```
const double& operator[] (
    size_t uiIndex ) const
```

10.53.3.6 `size()`

```
size_t size ( ) const
```

10.53.4 Member Data Documentation

10.53.4.1 `_pCount`

```
ATOMIC_VARIABLE* _pCount [protected]
```

10.53.4.2 `_pv`

```
std::vector<double>* _pv [protected]
```

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

- `include/SpinGenApi/Autovector.h`

10.54 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.

10.54.1 Detailed Description

Holds conversion methods for the access mode enumeration.

10.54.2 Member Function Documentation

10.54.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EAccessMode * pValue ) [static]
```

Converts a string to enum value.

10.54.2.2 ToString() [1/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EAccessMode * pValue ) [static]
```

Converts a string to an int32_t property.

10.54.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (  
    EAccessMode Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.55 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.

10.55.1 Detailed Description

Holds conversion methods for the caching mode enumeration.

10.55.2 Member Function Documentation

10.55.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

Converts a string to enum value.

10.55.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ECachingMode * pValue ) [static]
```

10.55.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ECachingMode Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.56 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.

10.56.1 Detailed Description

Holds conversion methods for the notation type of floats.

10.56.2 Member Function Documentation

10.56.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to enum value.

10.56.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EDisplayNotation * pValue ) [static]
```

Converts a string to an int32_t property.

10.56.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EDisplayNotation Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.57 EEndianessClass Class Reference

Holds conversion methods for the endianness 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.

10.57.1 Detailed Description

Holds conversion methods for the endianness enumeration.

10.57.2 Member Function Documentation

10.57.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to enum value.

10.57.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EEndianess * pValue ) [static]
```

Converts a string to an int32_t property.

10.57.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EEndianess Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.58 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.

10.58.1 Detailed Description

helper class converting EGenApiSchemaVersion from and to string

10.58.2 Member Function Documentation

10.58.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to enum value.

10.58.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EGenApiSchemaVersion * pValue ) [static]
```

Converts a string to an int32_t property.

10.58.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EGenApiSchemaVersion Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.59 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.

10.59.1 Detailed Description

Holds conversion methods for the notation type of floats.

10.59.2 Member Function Documentation

10.59.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to enum value.

10.59.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EInputDirection * pValue ) [static]
```

Converts a string to an int32_t property.

10.59.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EInputDirection Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.60 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.

10.60.1 Detailed Description

Holds conversion methods for the namespace enumeration.

10.60.2 Member Function Documentation

10.60.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ENamespace * pValue ) [static]
```

Converts a string to enum value.

10.60.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ENamespace * pValue ) [static]
```

Converts a string to an int32_t property.

10.60.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ENamespace Value ) [static]
```

Converts a string to an int32_t property.

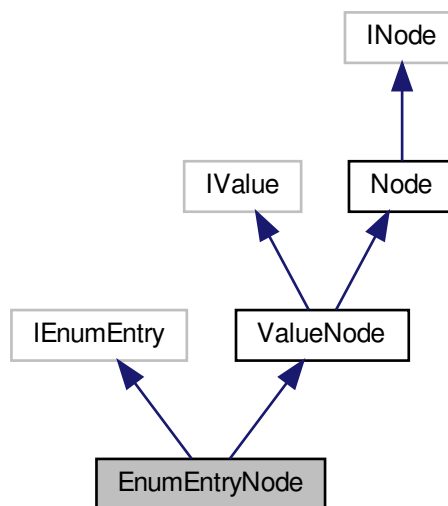
The documentation for this class was generated from the following file:

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

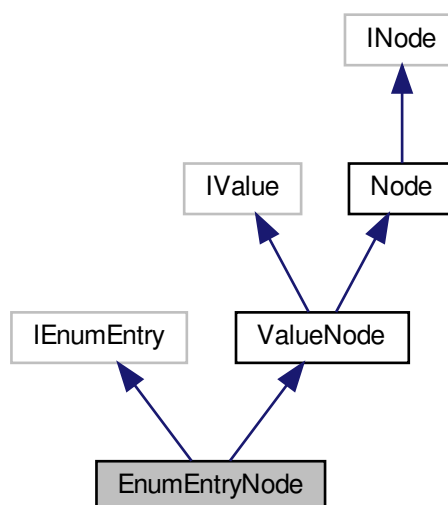
10.61 EnumEntryNode Class Reference

[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

10.61.1 Detailed Description

[Interface](#) for string properties.

10.61.2 Constructor & Destructor Documentation

10.61.2.1 EnumEntryNode() [1/2]

```
EnumEntryNode ( )
```

10.61.2.2 EnumEntryNode() [2/2]

```
EnumEntryNode (
    std::shared_ptr< Node::NodeImpl > pEnumEntry )
```

10.61.2.3 ~EnumEntryNode()

```
virtual ~EnumEntryNode ( ) [virtual]
```

10.61.3 Member Function Documentation

10.61.3.1 GetNumericValue()

```
virtual double GetNumericValue ( ) [virtual]
```

Get double number associated with the entry.

10.61.3.2 GetSymbolic()

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

Get symbolic enum value.

10.61.3.3 GetValue()

```
virtual int64_t GetValue ( ) [virtual]
```

Get numeric enum value.

10.61.3.4 IsSelfClearing()

```
virtual bool IsSelfClearing ( ) [virtual]
```

Indicates if the corresponding EnumEntry is self clearing.

10.61.3.5 SetReference()

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

overload SetReference for EnumEntry

Reimplemented from [ValueNode](#).

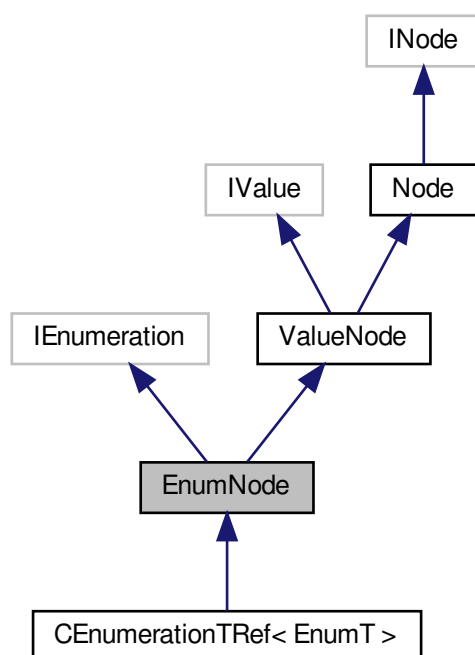
The documentation for this class was generated from the following file:

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

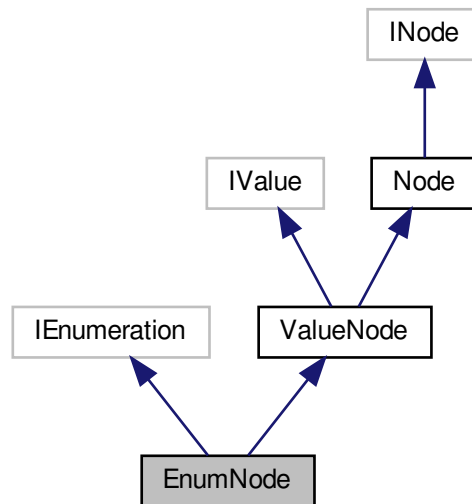
10.62 EnumNode Class Reference

[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`

10.62.1 Detailed Description

[Interface](#) for string properties.

10.62.2 Constructor & Destructor Documentation

10.62.2.1 EnumNode() [1/2]

```
EnumNode ( )
```

10.62.2.2 EnumNode() [2/2]

```
EnumNode (
    std::shared_ptr< Node::NodeImpl > pEnumeration )
```

10.62.2.3 ~EnumNode()

```
virtual ~EnumNode ( ) [virtual]
```

10.62.3 Member Function Documentation

10.62.3.1 GetCurrentEntry()

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

Get the current entry.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.2 GetEntries()

```
virtual void GetEntries (
    NodeList_t & Entries ) [virtual]
```

Get list of entry nodes.

10.62.3.3 GetEntry()

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

Get an entry node by its IntValue.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.4 GetEntryByName()

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

Get an entry node by name.

10.62.3.5 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

10.62.3.6 GetSymbolics()

```
virtual void GetSymbolics (
    StringList_t & Symbolics ) [virtual]
```

Get list of symbolic Values.

10.62.3.7 operator*()

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

Get string node value.

10.62.3.8 operator=()

```
virtual IEnumeration& operator= (
    const GenICam::gcstring & ValueStr ) [virtual]
```

Set string node value.

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.3.9 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) |

10.62.3.10 SetReference()

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

overload SetReference for Enumeration

Reimplemented from [ValueNode](#).

Reimplemented in [CEnumerationTRef< EnumT >](#).

10.62.4 Member Data Documentation

10.62.4.1 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](#)

10.63 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.

10.63.1 Detailed Description

Holds conversion methods for the representation enumeration.

10.63.2 Member Function Documentation

10.63.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to enum value.

10.63.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ERepresentation * pValue ) [static]
```

Converts a string to an int32_t property.

10.63.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ERepresentation Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.64 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.

10.64.1 Detailed Description

Holds conversion methods for the sign enumeration.

10.64.2 Member Function Documentation

10.64.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to enum value.

10.64.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESign * pValue ) [static]
```

Converts a string to an int32_t property.

10.64.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ESign Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.65 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.

10.65.1 Detailed Description

Holds conversion methods for the converter formulas.

10.65.2 Member Function Documentation

10.65.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to enum value.

10.65.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    ESlope * pValue ) [static]
```

Converts a string to an int32_t property.

10.65.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    ESlope Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.66 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.

10.66.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

10.66.2 Member Function Documentation

10.66.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to enum value.

10.66.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EStandardNameSpace * pValue ) [static]
```

Converts a string to an int32_t property.

10.66.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EStandardNameSpace Value ) [static]
```

Converts a string to an int32_t property.

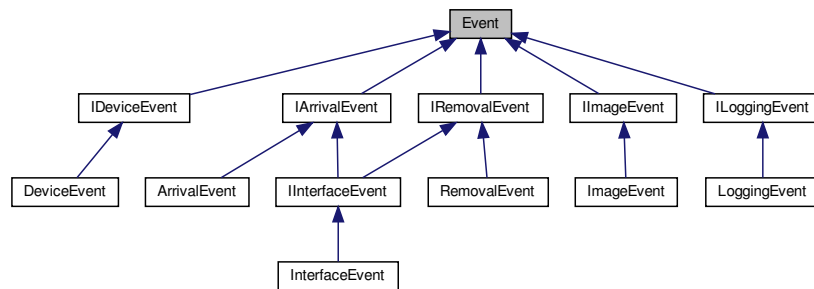
The documentation for this class was generated from the following file:

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

10.67 Event Class Reference

The base class for all event types.

Inheritance diagram for Event:



Public Member Functions

- virtual [~Event](#) ()
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

- [Event](#) ()
- [Event](#) & [operator=](#) (const [Event](#) &)
- void [SetEventPayload](#) (uint8_t *offset, size_t length)

Protected Attributes

- EventData * [m_pEventData](#)

Friends

- class [EventProcessor](#)
- class [IDataStream](#)
- class [Stream](#)

10.67.1 Detailed Description

The base class for all event types.

10.67.2 Constructor & Destructor Documentation

10.67.2.1 ~Event()

```
virtual ~Event ( ) [virtual]
```

Virtual Destructor.

10.67.2.2 Event()

```
Event ( ) [protected]
```

10.67.3 Member Function Documentation

10.67.3.1 GetEventPayloadData()

```
const uint8_t* GetEventPayloadData ( )
```

Gets the event payload data.

Returns

The event payload data

10.67.3.2 GetEventPayloadDataSize()

```
const size_t GetEventPayloadDataSize ( )
```

Gets the event payload data size.

Returns

The event payload data size

10.67.3.3 GetEventType()

```
EventType GetEventType ( )
```

Gets the event type.

Returns

The event type

10.67.3.4 operator=()

```
Event& operator= (
    const Event & ) [protected]
```

10.67.3.5 SetEventPayload()

```
void SetEventPayload (
    uint8_t * offset,
    size_t length ) [protected]
```

10.67.3.6 SetEventType()

```
void SetEventType (
    EventType eventType )
```

Sets the event type.

Parameters

| | |
|------------------|----------------|
| <i>eventType</i> | The event type |
|------------------|----------------|

10.67.4 Friends And Related Function Documentation

10.67.4.1 EventProcessor

```
friend class EventProcessor [friend]
```

10.67.4.2 IDataStream

```
friend class IDataStream [friend]
```

10.67.4.3 Stream

```
friend class Stream [friend]
```

10.67.5 Member Data Documentation

10.67.5.1 m_pEventData

```
EventData* m_pEventData [protected]
```

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

- include/Event.h

10.68 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.

10.68.1 Detailed Description

Holds conversion methods for the visibility enumeration.

10.68.2 Member Function Documentation

10.68.2.1 FromString()

```
static bool FromString (
    const GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to enum value.

10.68.2.2 ToString() [1/2]

```
static void ToString (
    GenICam::gcstring & ValueStr,
    EVisibility * pValue ) [static]
```

Converts a string to an int32_t property.

10.68.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (
    EVisibility Value ) [static]
```

Converts a string to an int32_t property.

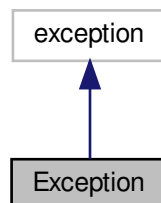
The documentation for this class was generated from the following file:

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

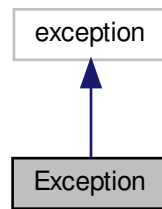
10.69 Exception Class Reference

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 *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

10.69.1 Detailed Description

The [Exception](#) object represents an error that is returned from the library.

Overloaded operators allow comparisons against other [Exception](#) objects.

10.69.2 Constructor & Destructor Documentation

10.69.2.1 Exception() [1/3]

```
Exception ( )
```

Default constructor.

10.69.2.2 Exception() [2/3]

```
Exception (
    int line,
    const char * fileName,
    const char * funcName,
    const char * buildDate,
    const char * buildTime,
    const char * errMsg,
    Error err )
```

Message constructor.

Parameters

| | |
|------------------|---|
| <i>line</i> | Line number where the exception is thrown |
| <i>fileName</i> | Name of the file called |
| <i>funcName</i> | Name of the function called |
| <i>buildDate</i> | Build date |
| <i>buildTime</i> | Build time |
| <i>errMsg</i> | A pointer to the exception message string |
| <i>err</i> | Error code |

10.69.2.3 Exception() [3/3]

```
Exception (
    const Exception & except )
```

Copy constructor.

10.69.2.4 ~Exception()

```
virtual ~Exception ( ) throw ( ) [virtual]
```

Default destructor.

10.69.3 Member Function Documentation

10.69.3.1 GetBuildDate()

```
const char* GetBuildDate ( ) const
```

10.69.3.2 GetBuildTime()

```
const char* GetBuildTime ( ) const
```

10.69.3.3 GetError()

```
Error GetError ( ) const
```

10.69.3.4 GetErrorMessage()

```
const char* GetErrorMessage ( ) const
```

Accessor Functions.

10.69.3.5 GetFileName()

```
const char* GetFileName ( ) const
```

10.69.3.6 GetFullErrorMessage()

```
const char* GetFullErrorMessage ( ) const
```

Gets the error code and full error message including the line, file, function, build date, and time.

10.69.3.7 GetFunctionName()

```
const char* GetFunctionName ( ) const
```

10.69.3.8 GetLineNumber()

```
int GetLineNumber ( ) const
```

10.69.3.9 operator!=(())

```
bool operator!= (
    const Error err ) const
```

Inequality operator.

10.69.3.10 operator=()

```
Exception& operator= (
    const Exception & except )
```

Assignment operator.

10.69.3.11 operator==(())

```
bool operator== (
    const Error err ) const
```

Equality operator.

10.69.3.12 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](#)

10.70 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.

10.70.1 Detailed Description

Holds conversion methods for the standard namespace enumeration.

10.70.2 Member Function Documentation

10.70.2.1 FromString()

```
static bool FromString (  
    const GenICam::gcstring & ValueStr,  
    EYesNo * pValue ) [static]
```

Converts a string to enum value.

10.70.2.2 ToString() [1/2]

```
static void ToString (  
    GenICam::gcstring & ValueStr,  
    EYesNo * pValue ) [static]
```

Converts a string to an int32_t property.

10.70.2.3 ToString() [2/2]

```
static GenICam::gcstring ToString (  
    EYesNo Value ) [static]
```

Converts a string to an int32_t property.

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

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

10.71 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.

10.71.1 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

10.71.2 Constructor & Destructor Documentation

10.71.2.1 FileProtocolAdapter()

```
FileProtocolAdapter ( )
```

Constructor.

10.71.2.2 ~FileProtocolAdapter()

```
virtual ~FileProtocolAdapter ( ) [virtual]
```

10.71.3 Member Function Documentation

10.71.3.1 attach()

```
bool attach (
    ::Spinnaker::GenApi::INodeMap * pInterface )
```

attach file protocol adapter to [NodeMap](#)

Parameters

| | |
|-------------------|--|
| <i>pInterface</i> | NodeMap of the device to which the FileProtocolAdapter is attached |
|-------------------|--|

Returns

true if attach was successful, false if not

10.71.3.2 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

10.71.3.3 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

10.71.3.4 getBufSize()

```
int64_t getBufSize (
    const char * pFileName,
    std::ios_base::openmode mode )
```


fetch max FileAccessBuffer length for a file

Parameters

| | |
|------------------|---|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
| <i>mode</i> | mode to open the file. The mode must exist in the Enumeration FileOpenMode |

Returns

max length of FileAccessBuffer in the given mode on the given file

10.71.3.5 openFile()

```
bool openFile (
    const char * pFileName,
    std::ios_base::openmode mode )
```

open a file on the device

Parameters

| | |
|------------------|---|
| <i>pFileName</i> | filename of the file to open. The filename must exist in the Enumeration FileSelector |
| <i>mode</i> | mode to open the file. The mode must exist in the Enumeration FileOpenMode |

Returns

true on success, false on error

10.71.3.6 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

10.71.3.7 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

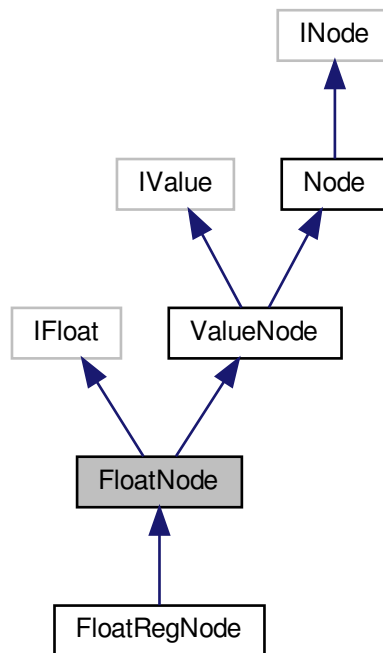
The documentation for this class was generated from the following file:

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

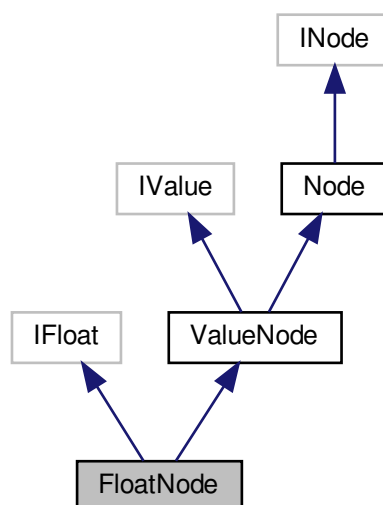
10.72 FloatNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for FloatNode:



Collaboration diagram for FloatNode:



Public Member Functions

- [FloatNode](#) ()
- [FloatNode](#) (std::shared_ptr< Node::NodeImpl > pFloat)
- virtual [~FloatNode](#) ()
- virtual void [SetValue](#) (double Value, bool [Verify](#)=true)
Set node value.
- virtual [IFloat](#) & [operator=](#) (double Value)
Set node value.
- virtual double [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
Get node value.
- virtual double [operator\(\)](#) ()
Get node value.
- virtual double [operator*](#) ()
Get node value.
- virtual double [GetMin](#) ()
Get minimum value allowed.
- virtual double [GetMax](#) ()
Get maximum value allowed.
- virtual bool [HasInc](#) ()
True if the float has a constant increment.
- virtual [EIncMode](#) [GetIncMode](#) ()
Get increment mode.
- virtual double [GetInc](#) ()
Get the constant increment if there is any.
- virtual [double_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
Get list of valid value.
- virtual [ERepresentation](#) [GetRepresentation](#) ()
Get recommended representation.
- virtual [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.
- [Integer](#) * [GetIntAlias](#) ()
gets the interface of an alias node.
- [IEnumeration](#) * [GetEnumAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (double Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (double Value)
Restrict maximum value.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Float

Additional Inherited Members

10.72.1 Detailed Description

[Interface](#) for string properties.

10.72.2 Constructor & Destructor Documentation

10.72.2.1 FloatNode() [1/2]

```
FloatNode ( )
```

10.72.2.2 FloatNode() [2/2]

```
FloatNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

10.72.2.3 ~FloatNode()

```
virtual ~FloatNode ( ) [virtual]
```

10.72.3 Member Function Documentation

10.72.3.1 GetDisplayNotation()

```
virtual EDisplayNotation GetDisplayNotation ( ) const [virtual]
```

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

10.72.3.2 GetDisplayPrecision()

```
virtual int64_t GetDisplayPrecision ( ) const [virtual]
```

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

10.72.3.3 GetEnumAlias()

```
IEnumeration* GetEnumAlias ( )
```

gets the interface of an alias node.

10.72.3.4 GetInc()

```
virtual double GetInc ( ) [virtual]
```

Get the constant increment if there is any.

10.72.3.5 GetIncMode()

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

Get increment mode.

10.72.3.6 GetIntAlias()

```
IInteger* GetIntAlias ( )
```

gets the interface of an alias node.

10.72.3.7 GetListOfValidValues()

```
virtual double_autovector_t GetListOfValidValues (
    bool bounded = true ) [virtual]
```

Get list of valid value.

10.72.3.8 GetMax()

```
virtual double GetMax ( ) [virtual]
```

Get maximum value allowed.

10.72.3.9 GetMin()

```
virtual double GetMin ( ) [virtual]
```

Get minimum value allowed.

10.72.3.10 GetRepresentation()

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

Get recommended representation.

10.72.3.11 GetUnit()

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

Get the physical unit name.

10.72.3.12 GetValue()

```
virtual double GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

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

Returns

The value read

10.72.3.13 HasInc()

```
virtual bool HasInc ( ) [virtual]
```

True if the float has a constant increment.

10.72.3.14 ImposeMax()

```
virtual void ImposeMax (
    double Value ) [virtual]
```

Restrict maximum value.

10.72.3.15 `ImposeMin()`

```
virtual void ImposeMin (
    double Value ) [virtual]
```

Restrict minimum value.

10.72.3.16 `operator()`

```
virtual double operator() ( ) [virtual]
```

Get node value.

10.72.3.17 `operator*()`

```
virtual double operator* ( ) [virtual]
```

Get node value.

10.72.3.18 `operator=()`

```
virtual IFloat& operator= (
    double Value ) [virtual]
```

Set node value.

10.72.3.19 `SetReference()`

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

overload SetReference for Float

Reimplemented from [ValueNode](#).

Reimplemented in [FloatRegNode](#).

10.72.3.20 `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) |

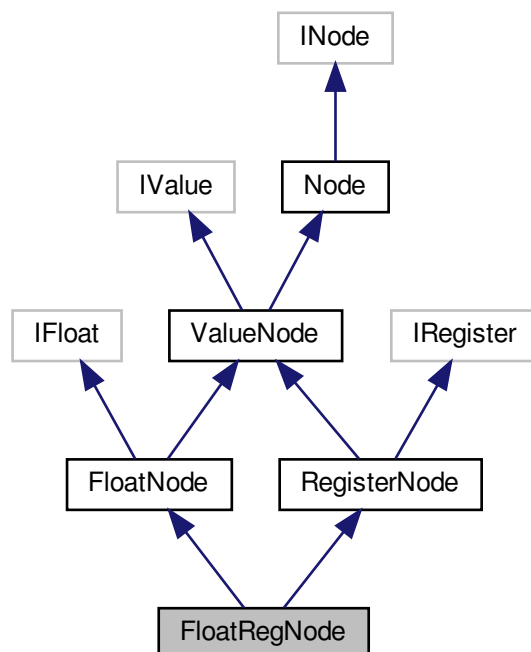
The documentation for this class was generated from the following file:

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

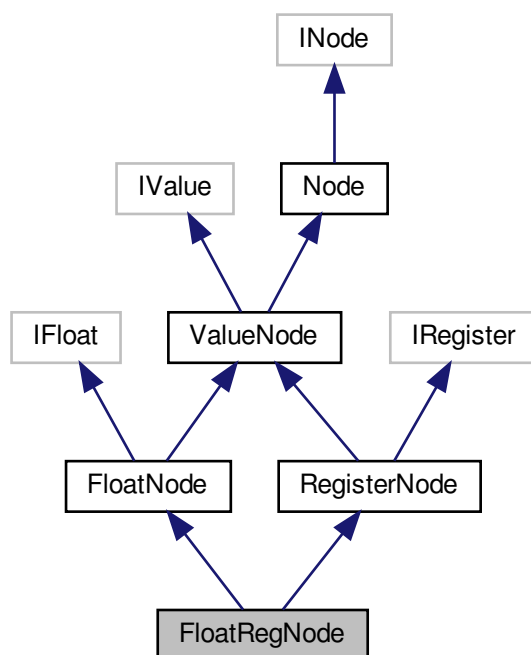
10.73 FloatRegNode Class Reference

[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

10.73.1 Detailed Description

[Interface](#) for string properties.

10.73.2 Constructor & Destructor Documentation

10.73.2.1 FloatRegNode() [1/2]

```
FloatRegNode ( )
```

10.73.2.2 FloatRegNode() [2/2]

```
FloatRegNode (
    std::shared_ptr< Node::NodeImpl > pFloat )
```

10.73.2.3 ~FloatRegNode()

```
virtual ~FloatRegNode ( ) [virtual]
```

10.73.3 Member Function Documentation

10.73.3.1 SetReference()

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

overload SetReference for Value

Reimplemented from [FloatNode](#).

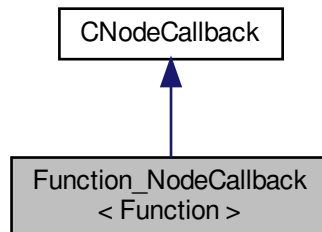
The documentation for this class was generated from the following file:

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

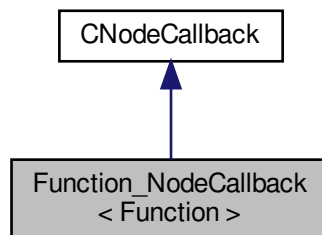
10.74 Function_NodeCallback< Function > Class Template Reference

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

10.74.1 Detailed Description

```
template<class Function>
```

```
class Spinnaker::GenApi::Function_NodeCallback< Function >
```

Container for a function pointer.

10.74.2 Constructor & Destructor Documentation

10.74.2.1 `Function_NodeCallback()`

```
Function_NodeCallback (
    INode * pNode,
    const Function & function,
    ECallbackType CallbackType ) [inline]
```

Constructor.

10.74.3 Member Function Documentation

10.74.3.1 `Destroy()`

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys teh object

Implements [CNodeCallback](#).

10.74.3.2 `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`

10.75 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)

10.75.1 Constructor & Destructor Documentation

10.75.1.1 [gcstring\(\)](#) [1/5]

```
gcstring ( )
```

10.75.1.2 [gcstring\(\)](#) [2/5]

```
gcstring (  
    const char * pc )
```

10.75.1.3 [gcstring\(\)](#) [3/5]

```
gcstring (  
    const char * pc,  
    size_t n )
```

10.75.1.4 [gcstring\(\)](#) [4/5]

```
gcstring (  
    size_t count,  
    char ch )
```

10.75.1.5 `gcstring()` [5/5]

```
gcstring (
    const gcstring & str )
```

10.75.1.6 `~gcstring()`

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

10.75.2 Member Function Documentation**10.75.2.1** `_npos()`

```
static size_t _npos (
    void ) [static]
```

10.75.2.2 `append()` [1/2]

```
virtual gcstring& append (
    const gcstring & str ) [virtual]
```

10.75.2.3 `append()` [2/2]

```
virtual gcstring& append (
    size_t count,
    char ch ) [virtual]
```

10.75.2.4 `assign()` [1/4]

```
virtual gcstring& assign (
    const gcstring & str ) [virtual]
```


10.75.2.5 assign() [2/4]

```
virtual gcstring& assign (  
    size_t count,  
    char ch ) [virtual]
```

10.75.2.6 assign() [3/4]

```
virtual gcstring& assign (  
    const char * pc ) [virtual]
```

10.75.2.7 assign() [4/4]

```
virtual gcstring& assign (  
    const char * pc,  
    size_t n ) [virtual]
```

10.75.2.8 c_str()

```
virtual const char* c_str (  
    void ) const [virtual]
```

10.75.2.9 compare()

```
virtual int compare (  
    const gcstring & str ) const [virtual]
```

10.75.2.10 empty()

```
virtual bool empty (  
    void ) const [virtual]
```

10.75.2.11 find() [1/5]

```
virtual size_t find (  
    char ch,  
    size_t offset = 0 ) const [virtual]
```

10.75.2.12 find() [2/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

10.75.2.13 find() [3/5]

```
virtual size_t find (  
    const gcstring & str,  
    size_t offset,  
    size_t count ) const [virtual]
```

10.75.2.14 find() [4/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset = 0 ) const [virtual]
```

10.75.2.15 find() [5/5]

```
virtual size_t find (  
    const char * pc,  
    size_t offset,  
    size_t count ) const [virtual]
```

10.75.2.16 find_first_not_of()

```
virtual size_t find_first_not_of (  
    const gcstring & str,  
    size_t offset = 0 ) const [virtual]
```

10.75.2.17 find_first_of()

```
virtual size_t find_first_of (
    const gcstring & str,
    size_t offset = 0 ) const [virtual]
```

10.75.2.18 length()

```
virtual size_t length (
    void ) const [virtual]
```

10.75.2.19 max_size()

```
virtual size_t max_size ( ) const [virtual]
```

10.75.2.20 operator const char *()

```
operator const char * (
    void ) const
```

10.75.2.21 operator delete() [1/2]

```
void operator delete (
    void * pWhere )
```

10.75.2.22 operator delete() [2/2]

```
void operator delete (
    void * pWhere,
    void * pNewWhere )
```

10.75.2.23 operator new() [1/2]

```
void* operator new (
    size_t uiSize )
```

10.75.2.24 operator new() [2/2]

```
void* operator new (
    size_t uiSize,
    void * pWhere )
```

10.75.2.25 operator!=() [1/2]

```
bool operator!= (
    const gcstring & str ) const
```

10.75.2.26 operator!=() [2/2]

```
bool operator!= (
    const char * pc ) const
```

10.75.2.27 operator+=() [1/5]

```
gcstring& operator+= (
    const gcstring & str )
```

10.75.2.28 operator+=() [2/5]

```
gcstring operator+= (
    const gcstring & str ) const
```

10.75.2.29 operator+=() [3/5]

```
gcstring& operator+= (
    const char * pc )
```

10.75.2.30 operator+=() [4/5]

```
gcstring& operator+= (
    char ch )
```

10.75.2.31 operator+=() [5/5]

```
gcstring operator+= (
    char ch ) const
```

10.75.2.32 operator<()

```
bool operator< (
    const gcstring & str ) const
```

10.75.2.33 operator=()

```
gcstring& operator= (
    const gcstring & str )
```

10.75.2.34 operator==() [1/2]

```
bool operator== (
    const gcstring & str ) const
```

10.75.2.35 operator==() [2/2]

```
bool operator== (
    const char * pc ) const
```

10.75.2.36 operator>()

```
bool operator> (
    const gcstring & str ) const
```

10.75.2.37 resize()

```
virtual void resize (
    size_t n ) [virtual]
```

10.75.2.38 size()

```
virtual size_t size (
    void ) const [virtual]
```

10.75.2.39 substr()

```
virtual gcstring substr (
    size_t offset = 0,
    size_t count = GCSTRING_NPOS ) const [virtual]
```

10.75.2.40 swap()

```
virtual void swap (
    gcstring & Right ) [virtual]
```

10.75.3 Friends And Related Function Documentation

10.75.3.1 operator+ [1/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const gcstring & right ) [friend]
```

10.75.3.2 operator+ [2/3]

```
SPINNAKER_API friend gcstring operator+ (
    const gcstring & left,
    const char * right ) [friend]
```

10.75.3.3 operator+ [3/3]

```
SPINNAKER_API friend gcstring operator+ (
    const char * left,
    const gcstring & right ) [friend]
```

10.75.4 Member Data Documentation

10.75.4.1 npos

```
const size_t npos [static]
```

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

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

10.76 GVCP_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

10.76.1 Detailed Description

header of a GVCP request packet

10.76.2 Member Data Documentation

10.76.2.1 ChunkID

```
uint32_t ChunkID
```

10.76.2.2 ChunkLength

```
uint32_t ChunkLength
```

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

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

10.77 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](#)

10.77.1 Detailed Description

layout of a GVCP event item (Extended ID flag not set)

10.77.2 Member Data Documentation

10.77.2.1 BlockId

uint16_t BlockId

10.77.2.2 EventId

uint16_t EventId

10.77.2.3 ReservedOrEventSize

uint16_t ReservedOrEventSize

10.77.2.4 StreamChannelId

uint16_t StreamChannelId

10.77.2.5 TimestampHigh

uint32_t TimestampHigh

10.77.2.6 TimestampLow

uint32_t TimestampLow

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

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

10.78 GVCP_EVENT_ITEM_BASIC Struct Reference

layout of a GVCP event item (common to all types)

Public Attributes

- uint16_t [ReservedOrEventSize](#)
- uint16_t [EventId](#)

10.78.1 Detailed Description

layout of a GVCP event item (common to all types)

10.78.2 Member Data Documentation

10.78.2.1 EventId

uint16_t EventId

10.78.2.2 ReservedOrEventSize

uint16_t ReservedOrEventSize

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

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

10.79 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](#)

10.79.1 Detailed Description

layout of a GVCP event item (Extended ID flag set)

10.79.2 Member Data Documentation

10.79.2.1 BlockId

uint16_t BlockId

10.79.2.2 BlockId64High

uint32_t BlockId64High

10.79.2.3 BlockId64Low

uint32_t BlockId64Low

10.79.2.4 EventId

uint16_t EventId

10.79.2.5 ReservedOrEventSize

```
uint16_t ReservedOrEventSize
```

10.79.2.6 StreamChannelId

```
uint16_t StreamChannelId
```

10.79.2.7 TimestampHigh

```
uint32_t TimestampHigh
```

10.79.2.8 TimestampLow

```
uint32_t TimestampLow
```

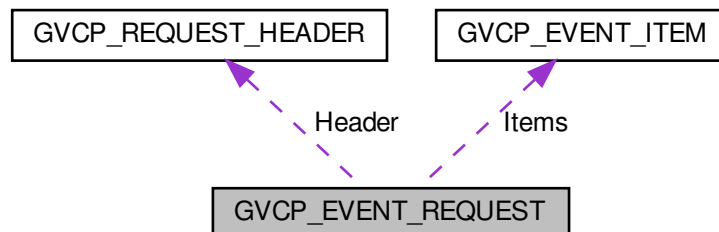
The documentation for this struct was generated from the following file:

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

10.80 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]

10.80.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag not set)

10.80.2 Member Data Documentation

10.80.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

10.80.2.2 Items

[GVCP_EVENT_ITEM](#) Items [1]

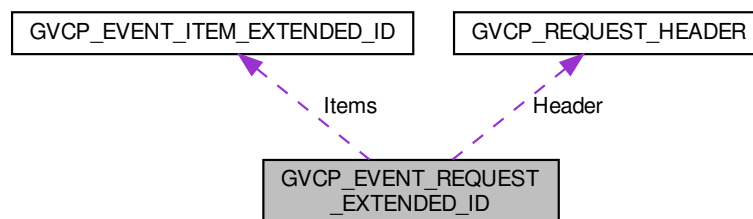
The documentation for this struct was generated from the following file:

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

10.81 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]

10.81.1 Detailed Description

Layout of a GVCP event request packet (Extended ID flag set)

10.81.2 Member Data Documentation

10.81.2.1 Header

[GVCP_REQUEST_HEADER](#) Header

10.81.2.2 Items

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Items [1]

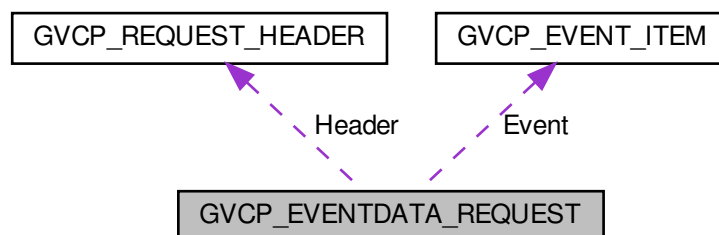
The documentation for this struct was generated from the following file:

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

10.82 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]

10.82.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag not set)

10.82.2 Member Data Documentation

10.82.2.1 Data

```
uint32_t Data[1]
```

10.82.2.2 Event

[GVCP_EVENT_ITEM](#) Event

10.82.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

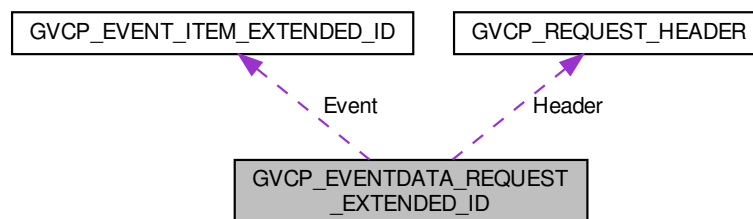
The documentation for this struct was generated from the following file:

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

10.83 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]

10.83.1 Detailed Description

Layout of a GVCP event data request packet (Extended ID flag set)

10.83.2 Member Data Documentation

10.83.2.1 Data

[uint32_t](#) [Data](#)[1]

10.83.2.2 Event

[GVCP_EVENT_ITEM_EXTENDED_ID](#) Event

10.83.2.3 Header

[GVCP_REQUEST_HEADER](#) Header

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

- `include/SpinGenApi/EventAdapterGEV.h`

10.84 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](#)

10.84.1 Detailed Description

header of a GVCP request packet

10.84.2 Member Data Documentation

10.84.2.1 Command

`uint16_t` Command

10.84.2.2 Flags

`uint8_t` Flags

10.84.2.3 Length

`uint16_t` Length

10.84.2.4 Magic

`uint8_t` Magic

10.84.2.5 ReqId

`uint16_t` ReqId

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

- `include/SpinGenApi/EventAdapterGEV.h`

10.85 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.

10.85.1 Detailed Description

Options for saving H264 files.

10.85.2 Constructor & Destructor Documentation

10.85.2.1 H264Option()

```
H264Option ( ) [inline]
```

10.85.3 Member Data Documentation

10.85.3.1 bitrate

```
unsigned int bitrate
```

Bit-rate to encode at.

10.85.3.2 frameRate

```
float frameRate
```

Frame rate of the stream.

10.85.3.3 height

```
unsigned int height
```

Height of source image.

10.85.3.4 reserved

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

Reserved for future use.

10.85.3.5 width

```
unsigned int width
```

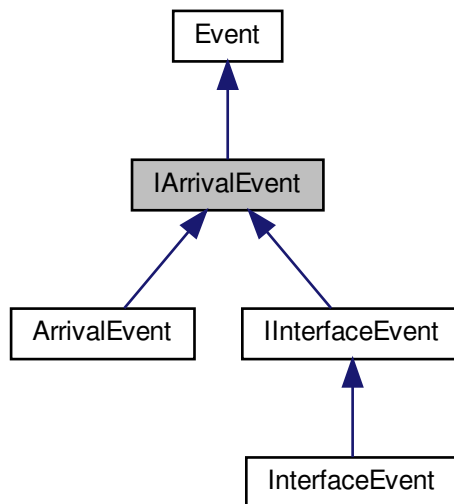
Width of source image.

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

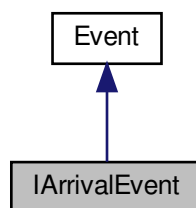
- [include/SpinVideoDefs.h](#)

10.86 IArrivalEvent Class Reference

Inheritance diagram for IArrivalEvent:



Collaboration diagram for IArrivalEvent:



Public Member Functions

- virtual [~IArrivalEvent](#) ()
- virtual void [OnDeviceArrival](#) (uint64_t serialNumber)=0

Protected Member Functions

- [IArrivalEvent](#) ()
- [IArrivalEvent](#) (const [IArrivalEvent](#) &)
- [IArrivalEvent](#) & [operator=](#) (const [IArrivalEvent](#) &)

Additional Inherited Members

10.86.1 Constructor & Destructor Documentation

10.86.1.1 `~IArrivalEvent()`

`virtual ~IArrivalEvent () [inline], [virtual]`

10.86.1.2 `IArrivalEvent()` [1/2]

`IArrivalEvent () [inline], [protected]`

10.86.1.3 `IArrivalEvent()` [2/2]

`IArrivalEvent (`
 `const IArrivalEvent &) [inline], [protected]`

10.86.2 Member Function Documentation

10.86.2.1 `OnDeviceArrival()`

`virtual void OnDeviceArrival (`
 `uint64_t serialNumber) [pure virtual]`

Implemented in [ArrivalEvent](#), [InterfaceEvent](#), and [IInterfaceEvent](#).

10.86.2.2 `operator=()`

`IArrivalEvent& operator= (`
 `const IArrivalEvent &) [protected]`

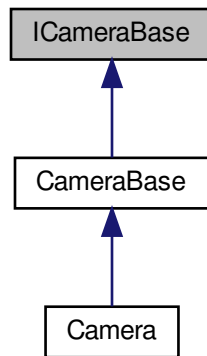
The documentation for this class was generated from the following file:

- [include/Interface/IArrivalEvent.h](#)

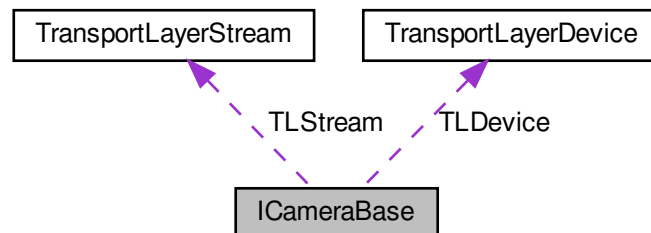
10.87 ICameraBase Class Reference

The interface file for base class for the camera object.

Inheritance diagram for ICameraBase:



Collaboration diagram for ICameraBase:



Public Member Functions

- virtual `~ICameraBase` (void)
- virtual void `Init` ()=0
- virtual void `Delinit` ()=0
- virtual bool `IsInitialized` ()=0
- virtual bool `IsValid` ()=0
- virtual `GenApi::INodeMap & GetNodeMap` () const =0
- virtual `GenApi::INodeMap & GetTLDeviceNodeMap` () const =0
- virtual `GenApi::INodeMap & GetTLStreamNodeMap` () 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 [RegisterEvent](#) ([Event](#) &evtToRegister)=0
- virtual void [RegisterEvent](#) ([Event](#) &evtToRegister, const [GenICam::gcstring](#) &eventName)=0
- virtual void [UnregisterEvent](#) ([Event](#) &evtToUnregister)=0
- virtual unsigned int [GetNumImagesInUse](#) ()=0
- virtual unsigned int [GetNumDataStreams](#) ()=0
- virtual unsigned int [DiscoverMaxPacketSize](#) ()=0
- virtual void [ForceIP](#) ()=0

Public Attributes

- [TransportLayerDevice](#) [TLDevice](#)
Gets vital camera information by connecting to the camera's bootstrap registers.
- [TransportLayerStream](#) [TLStream](#)
Gets information about the stream data by connecting to the camera's bootstrap registers.

Protected Member Functions

- [ICameraBase](#) ()
- [ICameraBase](#) (const [ICameraBase](#) &)
- [ICameraBase](#) & [operator=](#) (const [ICameraBase](#) &)

Protected Attributes

- CameraBaseData * [m_pCameraBaseData](#)

Friends

- class [CameraInternal](#)
- class [InterfacImpl](#)

10.87.1 Detailed Description

The interface file for base class for the camera object.

10.87.2 Constructor & Destructor Documentation

10.87.2.1 ~ICameraBase()

```
virtual ~ICameraBase (
    void ) [inline], [virtual]
```

10.87.2.2 ICameraBase() [1/2]

```
ICameraBase ( ) [inline], [protected]
```

10.87.2.3 ICameraBase() [2/2]

```
ICameraBase (
    const ICameraBase & ) [inline], [protected]
```

10.87.3 Member Function Documentation

10.87.3.1 BeginAcquisition()

```
virtual void BeginAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.2 DeInit()

```
virtual void DeInit ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.3 DiscoverMaxPacketSize()

```
virtual unsigned int DiscoverMaxPacketSize ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.4 EndAcquisition()

```
virtual void EndAcquisition ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.5 ForceIP()

```
virtual void ForceIP ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.6 GetAccessMode()

```
virtual GenApi::EAccessMode GetAccessMode ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.7 GetBufferOwnership()

```
virtual BufferOwnership GetBufferOwnership ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.8 GetGuiXml()

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

Implemented in [CameraBase](#).

10.87.3.9 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout = EVENT_TIMEOUT_INFINITE,
    uint64_t streamID = 0 ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.10 GetNodeMap()

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

Implemented in [CameraBase](#).

10.87.3.11 GetNumDataStreams()

```
virtual unsigned int GetNumDataStreams ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.12 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.13 GetTLDeviceNodeMap()

```
virtual GenApi::INodeMap& GetTLDeviceNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.14 GetTLStreamNodeMap()

```
virtual GenApi::INodeMap& GetTLStreamNodeMap ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.15 GetUniqueID()

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

Implemented in [CameraBase](#).

10.87.3.16 GetUserBufferCount()

```
virtual uint64_t GetUserBufferCount ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.17 GetUserBufferSize()

```
virtual uint64_t GetUserBufferSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.18 GetUserBufferTotalSize()

```
virtual uint64_t GetUserBufferTotalSize ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.19 Init()

```
virtual void Init ( ) [pure virtual]
```

Implemented in [CameraBase](#), and [Camera](#).

10.87.3.20 IsInitialized()

```
virtual bool IsInitialized ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.21 IsStreaming()

```
virtual bool IsStreaming ( ) const [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.22 IsValid()

```
virtual bool IsValid ( ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.23 operator=()

```
ICameraBase& operator= (
    const ICameraBase & ) [protected]
```

10.87.3.24 ReadPort()

```
virtual void ReadPort (
    uint64_t iAddress,
    void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.25 RegisterEvent() [1/2]

```
virtual void RegisterEvent (
    Event & evtToRegister ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.26 RegisterEvent() [2/2]

```
virtual void RegisterEvent (
    Event & evtToRegister,
    const GenICam::gcstring & eventName ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.27 SetBufferOwnership()

```
virtual void SetBufferOwnership (
    const BufferOwnership mode ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.28 SetUserBuffers() [1/2]

```
virtual void SetUserBuffers (
    void *const pMemBuffers,
    uint64_t totalSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.29 SetUserBuffers() [2/2]

```
virtual void SetUserBuffers (
    void **const ppMemBuffers,
    const uint64_t bufferCount,
    const uint64_t bufferSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.30 UnregisterEvent()

```
virtual void UnregisterEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.3.31 WritePort()

```
virtual void WritePort (
    uint64_t iAddress,
    const void * pBuffer,
    size_t iSize ) [pure virtual]
```

Implemented in [CameraBase](#).

10.87.4 Friends And Related Function Documentation

10.87.4.1 CameraInternal

```
friend class CameraInternal [friend]
```

10.87.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

10.87.5 Member Data Documentation

10.87.5.1 m_pCameraBaseData

```
CameraBaseData* m_pCameraBaseData [protected]
```

10.87.5.2 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.

10.87.5.3 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.

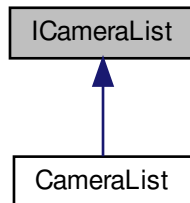
The documentation for this class was generated from the following file:

- [include/Interface/ICameraBase.h](#)

10.88 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 void `Clear ()=0`
- virtual void `RemoveBySerial (std::string serialNumber)=0`
- virtual void `RemoveByIndex (unsigned int index)=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`

10.88.1 Detailed Description

Used to hold a list of camera objects.

10.88.2 Constructor & Destructor Documentation

10.88.2.1 ~ICameraList()

```
virtual ~ICameraList ( ) [inline], [virtual]
```

10.88.2.2 ICameraList() [1/2]

```
ICameraList ( ) [inline], [protected]
```

10.88.2.3 ICameraList() [2/2]

```
ICameraList (
    const ICameraList & ) [inline], [protected]
```

10.88.3 Member Function Documentation

10.88.3.1 Append()

```
virtual void Append (
    CameraList & otherList ) [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.2 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.3 GetByIndex()

```
virtual CameraPtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.4 GetBySerial()

```
virtual CameraPtr GetBySerial (
    std::string serialNumber ) const [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.5 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.6 operator=()

```
ICameraList& operator= (
    const ICameraList & ) [protected]
```

10.88.3.7 operator[]()

```
virtual CameraPtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.8 RemoveByIndex()

```
virtual void RemoveByIndex (
    unsigned int index ) [pure virtual]
```

Implemented in [CameraList](#).

10.88.3.9 RemoveBySerial()

```
virtual void RemoveBySerial (
    std::string serialNumber ) [pure virtual]
```

Implemented in [CameraList](#).

10.88.4 Friends And Related Function Documentation

10.88.4.1 CameraListImpl

```
friend class CameraListImpl [friend]
```

10.88.4.2 InterfaceImpl

```
friend class InterfaceImpl [friend]
```

10.88.5 Member Data Documentation

10.88.5.1 m_pCameraListData

```
CameraListData* m_pCameraListData [protected]
```

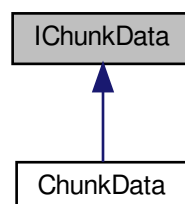
The documentation for this class was generated from the following file:

- [include/Interface/ICameraList.h](#)

10.89 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](#) [GetInferenceResult](#) () const =0
- virtual [float64_t](#) [GetInferenceConfidence](#) () const =0

Protected Member Functions

- [IChunkData](#) ()

10.89.1 Detailed Description

The [Interface](#) file for [ChunkData](#).

10.89.2 Constructor & Destructor Documentation

10.89.2.1 ~IChunkData()

```
virtual ~IChunkData ( ) [inline], [virtual]
```

10.89.2.2 IChunkData()

```
IChunkData ( ) [inline], [protected]
```

10.89.3 Member Function Documentation

10.89.3.1 GetBlackLevel()

```
virtual float64_t GetBlackLevel ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.2 GetCounterValue()

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

Implemented in [ChunkData](#).

10.89.3.3 GetCRC()

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

Implemented in [ChunkData](#).

10.89.3.4 GetEncoderValue()

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

Implemented in [ChunkData](#).

10.89.3.5 GetExposureEndLineStatusAll()

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

Implemented in [ChunkData](#).

10.89.3.6 GetExposureTime()

```
virtual float64_t GetExposureTime ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.7 GetFrameID()

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

Implemented in [ChunkData](#).

10.89.3.8 GetGain()

```
virtual float64_t GetGain ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.9 GetHeight()

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

Implemented in [ChunkData](#).

10.89.3.10 GetImage()

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

Implemented in [ChunkData](#).

10.89.3.11 GetInferenceConfidence()

```
virtual float64_t GetInferenceConfidence ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.12 GetInferenceResult()

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

Implemented in [ChunkData](#).

10.89.3.13 GetLinePitch()

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

Implemented in [ChunkData](#).

10.89.3.14 GetLineStatusAll()

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

Implemented in [ChunkData](#).

10.89.3.15 GetOffsetX()

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

Implemented in [ChunkData](#).

10.89.3.16 GetOffsetY()

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

Implemented in [ChunkData](#).

10.89.3.17 GetPartSelector()

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

Implemented in [ChunkData](#).

10.89.3.18 GetPixelDynamicRangeMax()

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

Implemented in [ChunkData](#).

10.89.3.19 GetPixelDynamicRangeMin()

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

Implemented in [ChunkData](#).

10.89.3.20 GetScan3dAxisMax()

```
virtual float64_t GetScan3dAxisMax ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.21 GetScan3dAxisMin()

```
virtual float64_t GetScan3dAxisMin ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.22 GetScan3dCoordinateOffset()

```
virtual float64_t GetScan3dCoordinateOffset ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.23 GetScan3dCoordinateReferenceValue()

```
virtual float64_t GetScan3dCoordinateReferenceValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.24 GetScan3dCoordinateScale()

```
virtual float64_t GetScan3dCoordinateScale ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.25 GetScan3dInvalidDataValue()

```
virtual float64_t GetScan3dInvalidDataValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.26 GetScan3dTransformValue()

```
virtual float64_t GetScan3dTransformValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.27 GetScanLineSelector()

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

Implemented in [ChunkData](#).

10.89.3.28 GetSequencerSetActive()

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

Implemented in [ChunkData](#).

10.89.3.29 GetSerialDataLength()

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

Implemented in [ChunkData](#).

10.89.3.30 GetStreamChannelID()

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

Implemented in [ChunkData](#).

10.89.3.31 GetTimerValue()

```
virtual float64_t GetTimerValue ( ) const [pure virtual]
```

Implemented in [ChunkData](#).

10.89.3.32 GetTimestamp()

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

Implemented in [ChunkData](#).

10.89.3.33 GetTimestampLatchValue()

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

Implemented in [ChunkData](#).

10.89.3.34 GetTransferBlockID()

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

Implemented in [ChunkData](#).

10.89.3.35 GetTransferQueueCurrentBlockCount()

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

Implemented in [ChunkData](#).

10.89.3.36 GetWidth()

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

Implemented in [ChunkData](#).

10.89.3.37 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](#)

10.90 IDataStream Class Reference**Public Member Functions**

- virtual [~IDataStream](#) ()
- virtual void [AnnouncelImage](#) (size_t size)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pPrivate)=0
- virtual void [AnnouncelImage](#) (size_t size, void *pData, void *pPrivate)=0
- virtual void [RevokelImages](#) ()=0
- virtual void [StartStream](#) (const unsigned int stream_index=0)=0
- virtual void [StopStream](#) ()=0
- virtual [ImagePtr](#) [GetNextImage](#) (uint64_t grabTimeout)=0
- virtual [ImagePtr](#) [GetNextImageInternal](#) (void **ppPrivate, uint64_t grabTimeout)=0
- virtual void [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 void [RegisterImageEvent](#) (IImageEvent &imageEvent, EventPollingOptions pollingOption)=0
- virtual void [UnregisterImageEvent](#) (IImageEvent &imageEvent)=0
- virtual void [WaitOnImageEvent](#) (uint64_t timeout)=0
- virtual void [InitChunkAdapter](#) (GenApi::INodeMap &nodemap)=0
- virtual void [CleanupChunkAdapter](#) ()=0
- virtual void [AddChunks](#) (GenApi::INodeMap &nodemap, [ImagePtr](#) pImage)=0
- virtual void [FillCRCInfo](#) (GenApi::INodeMap &nodemap, [ImagePtr](#) img)=0
- virtual [GenApi::INodeMap](#) & [GetNodeMap](#) () const =0
- virtual Port & [GetPort](#) () const =0
- virtual const [TransportLayerStream](#) & [TransportLayerStreamInfo](#) () const =0

Protected Member Functions

- [IDataStream\(\)](#)

10.90.1 Constructor & Destructor Documentation

10.90.1.1 ~IDataStream()

```
virtual ~IDataStream ( ) [inline], [virtual]
```

10.90.1.2 IDataStream()

```
IDataStream ( ) [inline], [protected]
```

10.90.2 Member Function Documentation

10.90.2.1 AddChunks()

```
virtual void AddChunks (
    GenApi::INodeMap & nodemap,
    ImagePtr pImage ) [pure virtual]
```

10.90.2.2 AnnounceImage() [1/3]

```
virtual void AnnounceImage (
    size_t size ) [pure virtual]
```

10.90.2.3 AnnounceImage() [2/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pPrivate ) [pure virtual]
```

10.90.2.4 AnnounceImage() [3/3]

```
virtual void AnnounceImage (
    size_t size,
    void * pData,
    void * pPrivate ) [pure virtual]
```

10.90.2.5 CleanupChunkAdapter()

```
virtual void CleanupChunkAdapter ( ) [pure virtual]
```

10.90.2.6 FillCRCInfo()

```
virtual void FillCRCInfo (
    GenApi::INodeMap & nodemap,
    ImagePtr img ) [pure virtual]
```

10.90.2.7 FlushQueueAllDiscard()

```
virtual void FlushQueueAllDiscard ( ) [pure virtual]
```

10.90.2.8 GetNextImage()

```
virtual ImagePtr GetNextImage (
    uint64_t grabTimeout ) [pure virtual]
```

10.90.2.9 GetNextImageInternal()

```
virtual ImagePtr GetNextImageInternal (
    void ** ppPrivate,
    uint64_t grabTimeout ) [pure virtual]
```

10.90.2.10 GetNodeMap()

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

10.90.2.11 GetNumImagesInUse()

```
virtual unsigned int GetNumImagesInUse ( ) const [pure virtual]
```

10.90.2.12 GetPort()

```
virtual Port& GetPort ( ) const [pure virtual]
```

10.90.2.13 InitChunkAdapter()

```
virtual void InitChunkAdapter (
    GenApi::INodeMap & nodemap ) [pure virtual]
```

10.90.2.14 IsImageInUse()

```
virtual bool IsImageInUse (
    const uint64_t imageID ) [pure virtual]
```

10.90.2.15 IsStreaming()

```
virtual bool IsStreaming ( ) [pure virtual]
```

10.90.2.16 KillBufferEvent()

```
virtual void KillBufferEvent ( ) [pure virtual]
```

10.90.2.17 RegisterImageEvent()

```
virtual void RegisterImageEvent (
    IImageEvent & imageEvent,
    EventPollingOptions pollingOption ) [pure virtual]
```

10.90.2.18 ReleaseImage()

```
virtual void ReleaseImage (
    const uint64_t imageID ) [pure virtual]
```

10.90.2.19 RevokeImages()

```
virtual void RevokeImages ( ) [pure virtual]
```

10.90.2.20 StartStream()

```
virtual void StartStream (
    const unsigned int stream_index = 0 ) [pure virtual]
```

10.90.2.21 StopStream()

```
virtual void StopStream ( ) [pure virtual]
```

10.90.2.22 TransportLayerStreamInfo()

```
virtual const TransportLayerStream& TransportLayerStreamInfo ( ) const [pure virtual]
```

10.90.2.23 UnregisterImageEvent()

```
virtual void UnregisterImageEvent (
    IImageEvent & imageEvent ) [pure virtual]
```

10.90.2.24 WaitOnImageEvent()

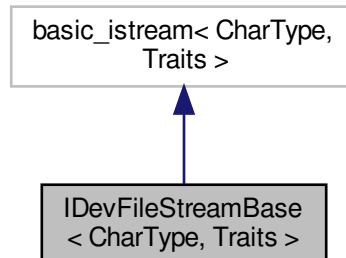
```
virtual void WaitOnImageEvent (
    uint64_t timeout ) [pure virtual]
```

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

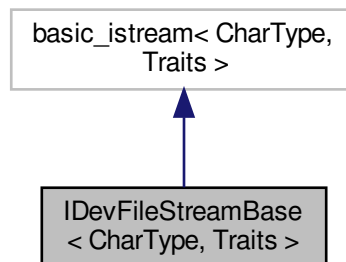
- [include/Interface/IStream.h](#)

10.91 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.

10.91.1 Member Typedef Documentation

10.91.1.1 filebuf_type

```
typedef IDevFileStreamBuf<CharType, Traits> filebuf_type
```

10.91.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

10.91.1.3 istream_type

```
typedef std::basic_istream<CharType, Traits> istream_type
```

10.91.2 Member Function Documentation

10.91.2.1 close()

```
void close ( ) [inline]
```

Close the file on the device.

10.91.2.2 is_open()

```
bool is_open ( ) const [inline]
```

10.91.2.3 open()

```
void open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

Open file on device in write mode.

Parameters

| | |
|-------------------|--|
| <i>pInterface</i> | NodeMap of the device to which the FileProtocolAdapter is attached |
| <i>pFileName</i> | Name of the file to open |
| <i>mode</i> | open mode |

10.91.2.4 rdbuf()

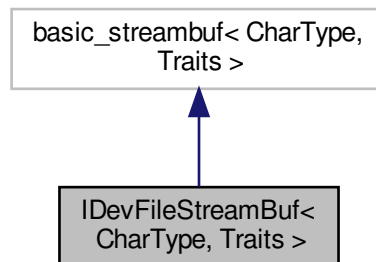
```
filebuf_type* rdbuf ( ) const [inline]
```

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

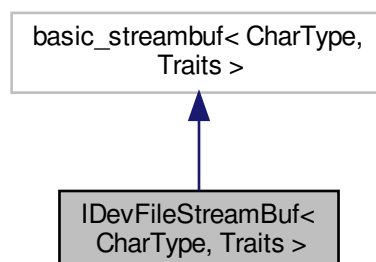
- include/SpinGenApi/[Filestream.h](#)

10.92 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)

10.92.1 Constructor & Destructor Documentation

10.92.1.1 IDevFileStreamBuf()

```
IDevFileStreamBuf ( ) [inline]
```

10.92.1.2 ~IDevFileStreamBuf()

```
~IDevFileStreamBuf ( ) [inline]
```

10.92.2 Member Function Documentation

10.92.2.1 close()

```
filebuf_type* close ( ) [inline]
```

10.92.2.2 is_open()

```
bool is_open ( ) const [inline]
```

10.92.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::in ) [inline]
```

10.92.2.4 pbackfail()

```
int_type pbackfail (
    int_type c ) [inline], [protected]
```

10.92.2.5 underflow()

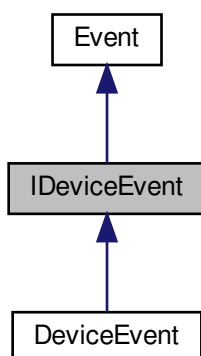
```
int_type underflow ( ) [inline], [protected]
```

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

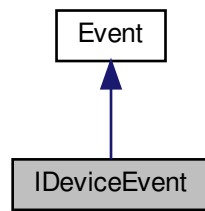
- include/SpinGenApi/[Filestream.h](#)

10.93 IDeviceEvent Class Reference

Inheritance diagram for IDeviceEvent:



Collaboration diagram for IDeviceEvent:



Public Member Functions

- virtual `~IDeviceEvent()`
- virtual void `OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0`
- virtual uint64_t `GetDeviceEventId()` const =0
- virtual `GenICam::gcstring GetDeviceEventName()` const =0

Protected Member Functions

- `IDeviceEvent()`
- `IDeviceEvent(const IDeviceEvent &)`
- `IDeviceEvent & operator= (const IDeviceEvent &)`

Additional Inherited Members

10.93.1 Constructor & Destructor Documentation

10.93.1.1 `~IDeviceEvent()`

```
virtual ~IDeviceEvent ( ) [inline], [virtual]
```

10.93.1.2 `IDeviceEvent()` [1/2]

```
IDeviceEvent ( ) [inline], [protected]
```

10.93.1.3 IDeviceEvent() [2/2]

```
IDeviceEvent (
    const IDeviceEvent & ) [inline], [protected]
```

10.93.2 Member Function Documentation

10.93.2.1 GetDeviceEventId()

```
virtual uint64_t GetDeviceEventId ( ) const [pure virtual]
```

Implemented in [DeviceEvent](#).

10.93.2.2 GetDeviceEventName()

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

Implemented in [DeviceEvent](#).

10.93.2.3 OnDeviceEvent()

```
virtual void OnDeviceEvent (
    Spinnaker::GenICam::gcstring eventName ) [pure virtual]
```

Implemented in [DeviceEvent](#).

10.93.2.4 operator=()

```
IDeviceEvent& operator= (
    const IDeviceEvent & ) [protected]
```

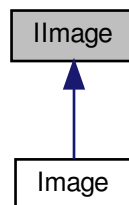
The documentation for this class was generated from the following file:

- [include/Interface/IDeviceEvent.h](#)

10.94 IImage Class Reference

The interface file for [Image](#).

Inheritance diagram for IImage:



Public Member Functions

- virtual [~IImage](#) ()
- virtual [ColorProcessingAlgorithm GetColorProcessing](#) () const =0
- virtual [ImagePtr Convert](#) ([Spinnaker::PixelFormatEnums](#) format, [ColorProcessingAlgorithm](#) color↔
Algorithm=DEFAULT) const =0
- [DEPRECATED_FUNC](#) ("This function has been deprecated. Please use more specific functions de-
fined in the [ImageUtilityPolarization](#) class.", virtual [ImagePtr ExtractPolarization](#)(const [PolarizationAlgorithm](#)
polarizationAlgorithm, const [PolarizationResolution](#) resolution) const =0;)
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormat↔
Enums](#) pixelFormat)=0
- virtual void [ResetImage](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormat↔
Enums](#) 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 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 [PayloadTypeInfoIds GetTLPayloadType](#) () const =0
- virtual uint64_t [GetTLPixelFormat](#) () const =0
- virtual [PixelFormatNamespaceID GetTLPixelFormatNamespace](#) () const =0

- virtual [GenlCam::gcstring GetPixelFormatName](#) () const =0
- virtual [Spinnaker::PixelFormatEnums GetPixelFormat](#) () const =0
- virtual [Spinnaker::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, [PGMOption](#) &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
- [DEPRECATED_FUNC](#) ("This function has been deprecated. Polarization images created through the [ImageUtilityPolarization](#) class now use an appropriate pixel format to hold the raw polarization values.", virtual float *GetPolarizationValues() const =0;)
- [DEPRECATED_FUNC](#) ("This function has been deprecated. Polarization algorithms are applied through specific functions defined in the [ImageUtilityPolarization](#) class.", virtual PolarizationAlgorithm GetPolarizationAlgorithm() const =0;)

Protected Member Functions

- [IImage](#) ()

10.94.1 Detailed Description

The interface file for [Image](#).

10.94.2 Constructor & Destructor Documentation

10.94.2.1 ~IImage()

```
virtual ~IImage ( ) [inline], [virtual]
```

10.94.2.2 IImage()

`IImage` () [inline], [protected]

10.94.3 Member Function Documentation

10.94.3.1 CalculateStatistics()

```
virtual void CalculateStatistics (
    ImageStatistics & pStatistics ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.2 CheckCRC()

```
virtual bool CheckCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.3 Convert()

```
virtual ImagePtr Convert (
    Spinnaker::PixelFormatEnums format,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.4 DeepCopy()

```
virtual void DeepCopy (
    const ImagePtr pSrcImage ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.5 DEPRECATED_FUNC() [1/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use more specific functions defined
    in the ImageUtilityPolarization class." ,
    virtual ImagePtr ExtractPolarization(const PolarizationAlgorithm polarization←
Alogrithm, const PolarizationResolution resolution) const = 0; )
```

10.94.3.6 DEPRECATED_FUNC() [2/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Polarization images created through the
ImageUtilityPolarization class now use an appropriate pixel format to hold the raw polarization
values." ,
    virtual float *GetPolarizationValues() const = 0; )
```

10.94.3.7 DEPRECATED_FUNC() [3/3]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Polarization algorithms are applied through
specific functions defined in the ImageUtilityPolarization class." ,
    virtual PolarizationAlgorithm GetPolarizationAlgorithm() const = 0; )
```

10.94.3.8 GetBitsPerPixel()

```
virtual size_t GetBitsPerPixel ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.9 GetBufferSize()

```
virtual size_t GetBufferSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.10 GetChunkData()

```
virtual const ChunkData& GetChunkData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.11 GetChunkLayoutId()

```
virtual uint64_t GetChunkLayoutId ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.12 GetColorProcessing()

```
virtual ColorProcessingAlgorithm GetColorProcessing ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.13 GetData()

```
virtual void* GetData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.14 GetFrameID()

```
virtual uint64_t GetFrameID ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.15 GetHeight()

```
virtual size_t GetHeight ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.16 GetID()

```
virtual uint64_t GetID ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.17 GetImageSize()

```
virtual size_t GetImageSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.18 GetImageStatus()

```
virtual ImageStatus GetImageStatus ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.19 GetNumChannels()

```
virtual size_t GetNumChannels ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.20 GetPayloadType()

```
virtual size_t GetPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.21 GetPixelFormat()

```
virtual Spinnaker::PixelFormatEnums GetPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.22 GetPixelFormatIntType()

```
virtual Spinnaker::PixelFormatIntType GetPixelFormatIntType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.23 GetPixelFormatName()

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

Implemented in [Image](#).

10.94.3.24 GetPrivateData()

```
virtual void* GetPrivateData ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.25 GetStride()

```
virtual size_t GetStride ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.26 GetTimeStamp()

```
virtual uint64_t GetTimeStamp ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.27 GetTLPayloadType()

```
virtual PayloadTypeInfoIDs GetTLPayloadType ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.28 GetTLPixelFormat()

```
virtual uint64_t GetTLPixelFormat ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.29 GetTLPixelFormatNamespace()

```
virtual PixelFormatNamespaceID GetTLPixelFormatNamespace ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.30 GetValidPayloadSize()

```
virtual size_t GetValidPayloadSize ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.31 GetWidth()

```
virtual size_t GetWidth ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.32 GetXOffset()

```
virtual size_t GetXOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.33 GetXPadding()

```
virtual size_t GetXPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.34 GetYOffset()

```
virtual size_t GetYOffset ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.35 GetYPadding()

```
virtual size_t GetYPadding ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.36 HasCRC()

```
virtual bool HasCRC ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.37 IsIncomplete()

```
virtual bool IsIncomplete ( ) const [pure virtual]
```

Implemented in [Image](#).

10.94.3.38 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.39 Release()

```
virtual void Release ( ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.40 ResetImage() [1/2]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.41 ResetImage() [2/2]

```
virtual void ResetImage (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.42 Save() [1/8]

```
virtual void Save (
    const char * pFilename,
    ImageFileFormat format = FROM_FILE_EXT ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.43 Save() [2/8]

```
virtual void Save (
    const char * pFilename,
    PNGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.44 Save() [3/8]

```
virtual void Save (
    const char * pFilename,
    PPMOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.45 Save() [4/8]

```
virtual void Save (
    const char * pFilename,
    PGMOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.46 Save() [5/8]

```
virtual void Save (  
    const char * pFilename,  
    TIFFOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.47 Save() [6/8]

```
virtual void Save (  
    const char * pFilename,  
    JPEGOption & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.48 Save() [7/8]

```
virtual void Save (  
    const char * pFilename,  
    JPG2Option & pOption ) [pure virtual]
```

Implemented in [Image](#).

10.94.3.49 Save() [8/8]

```
virtual void Save (  
    const char * pFilename,  
    BMPOption & pOption ) [pure virtual]
```

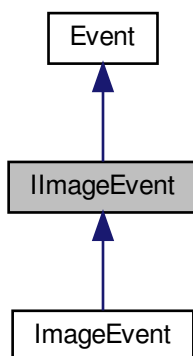
Implemented in [Image](#).

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

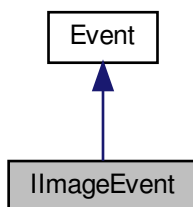
- [include/Interface/lImage.h](#)

10.95 QImageEvent Class Reference

Inheritance diagram for QImageEvent:



Collaboration diagram for QImageEvent:



Public Member Functions

- virtual [~QImageEvent](#) ()
- virtual void [OnQImageEvent](#) (QImagePtr image)=0

Protected Member Functions

- [QImageEvent](#) ()
- [QImageEvent](#) (const [QImageEvent](#) &)
- [QImageEvent](#) & [operator=](#) (const [QImageEvent](#) &)

Additional Inherited Members

10.95.1 Constructor & Destructor Documentation

10.95.1.1 ~IImageEvent()

```
virtual ~IImageEvent ( ) [inline], [virtual]
```

10.95.1.2 IImageEvent() [1/2]

```
IImageEvent ( ) [inline], [protected]
```

10.95.1.3 IImageEvent() [2/2]

```
IImageEvent (
    const IImageEvent & ) [inline], [protected]
```

10.95.2 Member Function Documentation

10.95.2.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

Implemented in [ImageEvent](#).

10.95.2.2 operator=()

```
IImageEvent& operator= (
    const IImageEvent & ) [protected]
```

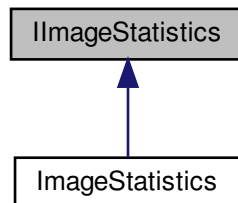
The documentation for this class was generated from the following file:

- [include/Interface/IImageEvent.h](#)

10.96 ImageStatistics Class Reference

The interface file for image statistics.

Inheritance diagram for ImageStatistics:



Public Member Functions

- virtual [~ImageStatistics](#) ()
- 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

- [ImageStatistics](#) ()
- [ImageStatistics](#) (const [ImageStatistics](#) &)

10.96.1 Detailed Description

The interface file for image statistics.

10.96.2 Constructor & Destructor Documentation

10.96.2.1 ~IImageStatistics()

`virtual ~IImageStatistics () [inline], [virtual]`

10.96.2.2 IImageStatistics() [1/2]

`IImageStatistics () [inline], [protected]`

10.96.2.3 IImageStatistics() [2/2]

`IImageStatistics (`
 `const IImageStatistics &) [inline], [protected]`

10.96.3 Member Function Documentation

10.96.3.1 DisableAll()

`virtual void DisableAll () [pure virtual]`

Implemented in [ImageStatistics](#).

10.96.3.2 EnableAll()

`virtual void EnableAll () [pure virtual]`

Implemented in [ImageStatistics](#).

10.96.3.3 EnableGreyOnly()

`virtual void EnableGreyOnly () [pure virtual]`

Implemented in [ImageStatistics](#).

10.96.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.6 GetChannelStatus()

```
virtual void GetChannelStatus (
    StatisticsChannel channel,
    bool * pEnabled ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.7 GetHistogram()

```
virtual void GetHistogram (
    StatisticsChannel channel,
    int ** ppHistogram ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.8 GetMean()

```
virtual void GetMean (
    StatisticsChannel channel,
    float * pPixelValueMean ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.9 GetNumPixelValues()

```
virtual void GetNumPixelValues (
    StatisticsChannel channel,
    unsigned int * pNumPixelValues ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.10 GetPixelValueRange()

```
virtual void GetPixelValueRange (
    StatisticsChannel channel,
    unsigned int * pPixelValueMin,
    unsigned int * pPixelValueMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.11 GetRange()

```
virtual void GetRange (
    StatisticsChannel channel,
    unsigned int * pMin,
    unsigned int * pMax ) const [pure virtual]
```

Implemented in [ImageStatistics](#).

10.96.3.12 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](#).

10.96.3.13 SetChannelStatus()

```
virtual void SetChannelStatus (
    StatisticsChannel channel,
    bool enabled ) [pure virtual]
```

Implemented in [ImageStatistics](#).

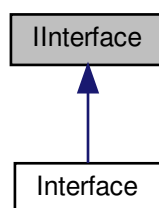
The documentation for this class was generated from the following file:

- [include/Interface/IImageStatistics.h](#)

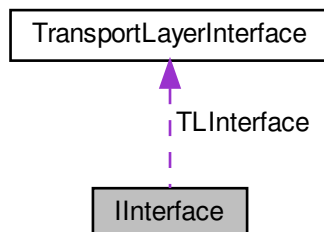
10.97 IInterface Class Reference

The interface file for [Interface](#).

Inheritance diagram for IInterface:



Collaboration diagram for IInterface:



Public Member Functions

- virtual [~IInterface](#) ()
- virtual [CameraList](#) [GetCameras](#) (bool updateCameras=true) const =0
- virtual bool [UpdateCameras](#) ()=0
- virtual [GenApi::INodeMap](#) & [GetTLNodeMap](#) () const =0
- virtual void [RegisterEvent](#) ([Event](#) &evtToRegister)=0
- virtual void [UnregisterEvent](#) ([Event](#) &evtToUnregister)=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

Public Attributes

- [TransportLayerInterface](#) [TLInterface](#)

Protected Member Functions

- [IInterface](#) ()
- [IInterface](#) (const [IInterface](#) &)
- [IInterface](#) & [operator=](#) (const [IInterface](#) &)

Protected Attributes

- InterfaceData * [m_pInterfaceData](#)

Friends

- class [InterfaceInternal](#)

10.97.1 Detailed Description

The interface file for [Interface](#).

10.97.2 Constructor & Destructor Documentation

10.97.2.1 ~IInterface()

```
virtual ~IInterface ( ) [inline], [virtual]
```

10.97.2.2 Interface() [1/2]

```
IInterface ( ) [inline], [protected]
```

10.97.2.3 Interface() [2/2]

```
IInterface (
    const IInterface & ) [inline], [protected]
```

10.97.3 Member Function Documentation

10.97.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateCameras = true ) const [pure virtual]
```

Implemented in [Interface](#).

10.97.3.2 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [Interface](#).

10.97.3.3 IsInUse()

```
virtual bool IsInUse ( ) const [pure virtual]
```

Implemented in [Interface](#).

10.97.3.4 operator=()

```
IInterface& operator= (
    const IInterface & ) [protected]
```


10.97.3.5 RegisterEvent()

```
virtual void RegisterEvent (
    Event & evtToRegister ) [pure virtual]
```

Implemented in [Interface](#).

10.97.3.6 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](#).

10.97.3.7 UnregisterEvent()

```
virtual void UnregisterEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [Interface](#).

10.97.3.8 UpdateCameras()

```
virtual bool UpdateCameras ( ) [pure virtual]
```

Implemented in [Interface](#).

10.97.4 Friends And Related Function Documentation

10.97.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

10.97.5 Member Data Documentation

10.97.5.1 m_pInterfaceData

```
InterfaceData* m_pInterfaceData [protected]
```

10.97.5.2 TLInterface

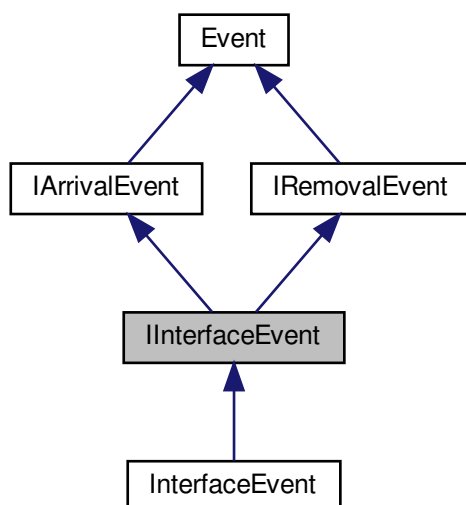
```
TransportLayerInterface TLInterface
```

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

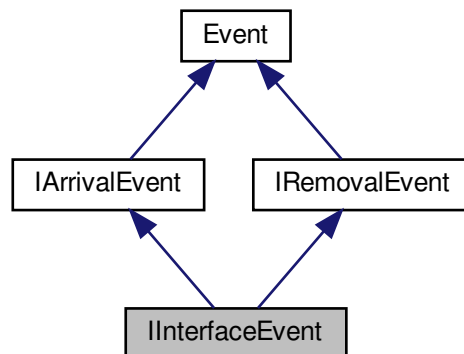
- [include/Interface/IInterface.h](#)

10.98 IInterfaceEvent Class Reference

Inheritance diagram for IInterfaceEvent:



Collaboration diagram for IInterfaceEvent:



Public Member Functions

- virtual `~IInterfaceEvent` ()
- virtual void `OnDeviceArrival` (uint64_t serialNumber)=0
- virtual void `OnDeviceRemoval` (uint64_t serialNumber)=0

Protected Member Functions

- `IInterfaceEvent` ()
- `IInterfaceEvent` (const `IInterfaceEvent` &)
- `IInterfaceEvent` & `operator=` (const `IInterfaceEvent` &)

Additional Inherited Members

10.98.1 Constructor & Destructor Documentation

10.98.1.1 `~IInterfaceEvent()`

```
virtual ~IInterfaceEvent ( ) [inline], [virtual]
```

10.98.1.2 `IInterfaceEvent()` [1/2]

```
IInterfaceEvent ( ) [inline], [protected]
```

10.98.1.3 `IInterfaceEvent()` [2/2]

```
IInterfaceEvent (
    const IInterfaceEvent & ) [inline], [protected]
```

10.98.2 Member Function Documentation

10.98.2.1 `OnDeviceArrival()`

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IArrivalEvent](#).

Implemented in [InterfaceEvent](#).

10.98.2.2 `OnDeviceRemoval()`

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implements [IRemovalEvent](#).

Implemented in [InterfaceEvent](#).

10.98.2.3 `operator=()`

```
IInterfaceEvent& operator= (
    const IInterfaceEvent & ) [protected]
```

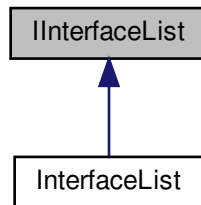
The documentation for this class was generated from the following file:

- [include/Interface/IInterfaceEvent.h](#)

10.99 IInterfaceList Class Reference

The interface file for [IInterfaceList](#) class.

Inheritance diagram for IInterfaceList:



Public Member Functions

- virtual [~IInterfaceList](#) (void)
- virtual [InterfacePtr operator\[\]](#) (unsigned int index)=0
- virtual unsigned int [GetSize](#) () const =0
- virtual [InterfacePtr GetByIndex](#) (unsigned int index) const =0
- virtual void [Clear](#) ()=0

Protected Member Functions

- [IInterfaceList](#) (void)
- [IInterfaceList](#) (const [IInterfaceList](#) &)
- [IInterfaceList](#) & [operator=](#) (const [IInterfaceList](#) &)

Protected Attributes

- InterfaceListData * [m_pInterfaceListData](#)

10.99.1 Detailed Description

The interface file for [IInterfaceList](#) class.

10.99.2 Constructor & Destructor Documentation

10.99.2.1 ~IInterfaceList()

```
virtual ~IInterfaceList (
    void ) [inline], [virtual]
```

10.99.2.2 IInterfaceList() [1/2]

```
IInterfaceList (
    void ) [inline], [protected]
```

10.99.2.3 IInterfaceList() [2/2]

```
IInterfaceList (
    const IInterfaceList & ) [inline], [protected]
```

10.99.3 Member Function Documentation

10.99.3.1 Clear()

```
virtual void Clear ( ) [pure virtual]
```

Implemented in [InterfaceList](#).

10.99.3.2 GetByIndex()

```
virtual InterfacePtr GetByIndex (
    unsigned int index ) const [pure virtual]
```

Implemented in [InterfaceList](#).

10.99.3.3 GetSize()

```
virtual unsigned int GetSize ( ) const [pure virtual]
```

Implemented in [InterfaceList](#).

10.99.3.4 operator=()

```
IInterfaceList& operator= (
    const IInterfaceList & ) [protected]
```

10.99.3.5 operator[]()

```
virtual InterfacePtr operator[] (
    unsigned int index ) [pure virtual]
```

Implemented in [InterfaceList](#).

10.99.4 Member Data Documentation

10.99.4.1 m_pInterfaceListData

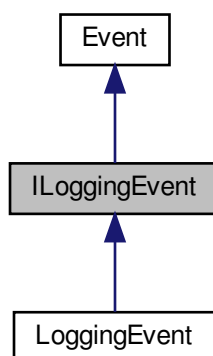
```
InterfaceListData* m_pInterfaceListData [protected]
```

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

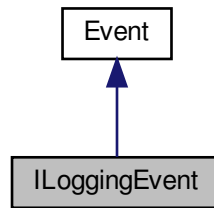
- [include/Interface/IInterfaceList.h](#)

10.100 ILoggingEvent Class Reference

Inheritance diagram for ILoggingEvent:



Collaboration diagram for ILoggingEvent:



Public Member Functions

- virtual `~ILoggingEvent()`
- virtual void `OnLogEvent(LoggingEventDataPtr eventPtr)=0`

Protected Member Functions

- `ILoggingEvent()`
- `ILoggingEvent(const ILoggingEvent &)`
- `ILoggingEvent & operator=(const ILoggingEvent &)`

Additional Inherited Members

10.100.1 Constructor & Destructor Documentation

10.100.1.1 `~ILoggingEvent()`

```
virtual ~ILoggingEvent ( ) [inline], [virtual]
```

10.100.1.2 `ILoggingEvent()` [1/2]

```
ILoggingEvent ( ) [inline], [protected]
```


10.100.1.3 ILoggingEvent() [2/2]

```
ILoggingEvent (
    const ILoggingEvent & ) [inline], [protected]
```

10.100.2 Member Function Documentation

10.100.2.1 OnLogEvent()

```
virtual void OnLogEvent (
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

Implemented in [LoggingEvent](#).

10.100.2.2 operator=()

```
ILoggingEvent& operator= (
    const ILoggingEvent & ) [protected]
```

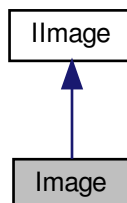
The documentation for this class was generated from the following file:

- [include/Interface/ILoggingEvent.h](#)

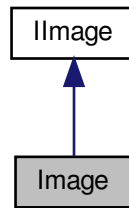
10.101 Image Class Reference

The image object class.

Inheritance diagram for Image:



Collaboration diagram for Image:



Public Member Functions

- virtual `~Image ()`
Virtual destructor.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static `ImagePtr` `CreateHeatMap(const ImagePtr &srcImage)`)
Computes a heatmap image.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `CreateHeatMap(const ImagePtr &srcImage, ImagePtr &destImage)`)
Computes a heatmap image.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", 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.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `GetHeatMapColorGradient(HeatMapColor ¤tLowColor, HeatMapColor ¤tHighColor)`)
Returns the current heatmap gradient color range.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", 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.
- **DEPRECATED_FUNC** ("This function has been deprecated. Please use the equivalent function in the `ImageUtilityHeatmap` class.", static void `GetHeatMapRange(unsigned int ¤tLowValue, unsigned int ¤tHighValue)`)
Returns the current high and low values used in heatmap representations.
- `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 `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.
- 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.
- `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()
- `DEPRECATED_FUNC` ("This function has been deprecated. Please use more specific functions defined in the `ImageUtilityPolarization` class.", `ImagePtr ExtractPolarization(const PolarizationAlgorithm polarization←Algorithm, const PolarizationResolution resolution) const`)

Extracts an image from a monochrome-polarized sensor.
- `DEPRECATED_FUNC` ("This function has been deprecated. Polarization images created through the `ImageUtilityPolarization` class now use an appropriate pixel format to hold the raw polarization values.", `float *GetPolarizationValues() const`)

Returns the polarization values associated with an extracted polarization image.
- `DEPRECATED_FUNC` ("This function has been deprecated. Polarization algorithms are applied through specific functions defined in the `ImageUtilityPolarization` class.", `PolarizationAlgorithm GetPolarizationAlgorithm() const`)

Returns the polarization algorithm used to extract a polarization image.
- `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

- [Image](#) ()
- [Image](#) (const [ImagePtr](#) image)
- [Image](#) (size_t width, size_t height, size_t offsetX, size_t offsetY, [Spinnaker::PixelFormatEnums](#) pixelFormat, void *pData)
- [ImagePtr CreateShared](#) () const
- void [DeepCopy](#) (const [Image](#) &pSrcImage)
- void [Convert](#) ([Spinnaker::PixelFormatEnums](#) format, [Image](#) &pDestImage, [ColorProcessingAlgorithm](#) colorAlgorithm=[DEFAULT](#)) const

Protected Attributes

- [ImageData](#) * [m_pImageData](#)

Friends

- class [IDataStream](#)
- class [Stream](#)
- class [ImageConverter](#)
- class [ImageFiler](#)
- class [ImageStatsCalculator](#)
- class [ImageUtilityImpl](#)

10.101.1 Detailed Description

The image object class.

10.101.2 Constructor & Destructor Documentation

10.101.2.1 ~Image()

```
virtual ~Image ( ) [virtual]
```

Virtual destructor.

10.101.2.2 Image() [1/3]

```
Image ( ) [protected]
```

10.101.2.3 Image() [2/3]

```
Image (
    const ImagePtr image ) [protected]
```

10.101.2.4 Image() [3/3]

```
Image (
    size_t width,
    size_t height,
    size_t offsetX,
    size_t offsetY,
    Spinnaker::PixelFormatEnums pixelFormat,
    void * pData ) [protected]
```

10.101.3 Member Function Documentation

10.101.3.1 CalculateStatistics()

```
void CalculateStatistics (
    ImageStatistics & pStatistics ) [virtual]
```

Retrieves a number of pixel statistics for an image including a histogram array of the range of pixel values.

Parameters

| | |
|--------------------|-----------------------------|
| <i>pStatistics</i> | The statistics of an image. |
|--------------------|-----------------------------|

Implements [IImage](#).

10.101.3.2 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](#).

10.101.3.3 Convert() [1/2]

```
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](#).

10.101.3.4 Convert() [2/2]

```
void Convert (
    Spinnaker::PixelFormatEnums format,
    Image & pDestImage,
    ColorProcessingAlgorithm colorAlgorithm = DEFAULT ) const [protected]
```

10.101.3.5 Create() [1/3]

```
static ImagePtr Create ( ) [static]
```

Create an image object.

10.101.3.6 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 |
|--------------|-------------------------|

10.101.3.7 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 |

10.101.3.8 CreateShared()

```
ImagePtr CreateShared ( ) const [protected]
```


10.101.3.9 DeepCopy() [1/2]

```
void DeepCopy (
    const ImagePtr pSrcImage ) [virtual]
```

Performs a deep copy of the [Image](#).

After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The [Image](#)'s current buffer will not be released.

Parameters

| | |
|------------------|--|
| <i>pSrcImage</i> | The Image to copy the data from. |
|------------------|--|

Implements [IImage](#).

10.101.3.10 DeepCopy() [2/2]

```
void DeepCopy (
    const Image & pSrcImage ) [protected]
```

10.101.3.11 DEPRECATED_FUNC() [1/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
    static ImagePtr CreateHeatMap(const ImagePtr &srcImage )
```

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

10.101.3.12 DEPRECATED_FUNC() [2/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
    static void CreateHeatMapconst ImagePtr &srcImage, ImagePtr &destImage )
```

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()

10.101.3.13 DEPRECATED_FUNC() [3/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
    static void SetHeatMapColorGradientconst HeatMapColor newLowColor, const Heat↵
    MapColor newHighColor )
```

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. |

10.101.3.14 DEPRECATED_FUNC() [4/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
```

```
static void GetHeatMapColorGradientHeatMapColor &currentLowColor, HeatMapColor
&currentHighColor )
```

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()

10.101.3.15 DEPRECATED_FUNC() [5/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
    static void SetHeatMapRangeconst 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.

Acceptable values range from 0 to 100.

Parameters

| | |
|---------------------|---|
| <i>newLowValue</i> | New value at which to begin color representation. |
| <i>newHighValue</i> | New value at which to end color representation. |

10.101.3.16 DEPRECATED_FUNC() [6/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use the equivalent function in the
    ImageUtilityHeatmap class." ,
    static void GetHeatMapRangeunsigned int &currentLowValue, unsigned int &current↔
    HighValue )
```

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()

10.101.3.17 DEPRECATED_FUNC() [7/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Please use more specific functions defined
in the ImageUtilityPolarization class." ,
    ImagePtr ExtractPolarization(const PolarizationAlgorithm polarizationAlgorithm,
const PolarizationResolution resolution) const )
```

Extracts an image from a monochrome-polarized sensor.

The extracted image will be returned as Mono8 or BGRa8 for heatmap images.

Parameters

| | |
|------------------------------|--|
| <i>polarizationAlgorithm</i> | Desired polarization algorithm to use. |
| <i>resolution</i> | Desired resolution of output image. |

Returns

The converted image.

10.101.3.18 DEPRECATED_FUNC() [8/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated. Polarization images created through the
ImageUtilityPolarization class now use an appropriate pixel format to hold the raw polarization
values." ,
    float *GetPolarizationValues() const )
```

Returns the polarization values associated with an extracted polarization image.

Note that standard quadrants (QUADRANT_I0_GRAYSCALE - QUADRANT_I135_GRAYSCALE) do not provide polarization values.

Returns

The polarization values associated with a polarization image.

10.101.3.19 DEPRECATED_FUNC() [9/9]

```
DEPRECATED_FUNC (
    "This function has been deprecated.  Polarization algorithms are applied through
    specific functions defined in the ImageUtilityPolarization class." ,
    PolarizationAlgorithm GetPolarizationAlgorithm() const )
```

Returns the polarization algorithm used to extract a polarization image.

Returns

The polarization algorithm used to extract the polarization image.

10.101.3.20 GetBitsPerPixel()

```
size_t GetBitsPerPixel ( ) const [virtual]
```

Gets the number of bits used per pixel in the image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The number of bits used per pixel.

Implements [Image](#).

10.101.3.21 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 [Image](#).

10.101.3.22 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](#).

10.101.3.23 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](#).

10.101.3.24 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](#).

10.101.3.25 GetData()

```
void* GetData ( ) const [virtual]
```

Gets a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is released. The pointer may also be invalidated if the [Image](#) object is passed to [Image::Release\(\)](#).

Returns

A pointer to the image data.

Implements [Image](#).

10.101.3.26 GetDefaultColorProcessing()

```
static ColorProcessingAlgorithm GetDefaultColorProcessing ( ) [static]
```

Gets the default color processing algorithm.

See also

[SetDefaultColorProcessing\(\)](#)

Returns

The default color processing algorithm.

10.101.3.27 GetFrameID()

```
uint64_t GetFrameID ( ) const [virtual]
```

Gets the frame ID for this image.

Returns

The frame ID.

Implements [Image](#).

10.101.3.28 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](#).

10.101.3.29 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](#).

10.101.3.30 GetImageSize()

```
size_t GetImageSize ( ) const [virtual]
```

Returns the size of the image.

Returns

The image size in bytes.

Implements [IImage](#).

10.101.3.31 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](#).

10.101.3.32 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.

10.101.3.33 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](#).

10.101.3.34 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](#).

10.101.3.35 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](#).

10.101.3.36 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](#).

10.101.3.37 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](#).

10.101.3.38 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](#).

10.101.3.39 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](#).

10.101.3.40 GetTimeStamp()

```
uint64_t GetTimeStamp ( ) const [virtual]
```

Gets the time stamp for the image in nanoseconds.

Returns

The time stamp of the image.

Implements [IImage](#).

10.101.3.41 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](#).

10.101.3.42 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](#).

10.101.3.43 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](#).

10.101.3.44 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. [Get↵BufferSize\(\)](#) returns the total size of bytes allocated for the image.

See also

[GetBufferSize\(\)](#)

Returns

size_t value representing valid payload.

Implements [IImage](#).

10.101.3.45 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](#).

10.101.3.46 GetXOffset()

```
size_t GetXOffset ( ) const [virtual]
```

Gets the ROI x offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x offset in pixels.

Implements [Image](#).

10.101.3.47 GetXPadding()

```
size_t GetXPadding ( ) const [virtual]
```

Gets the x padding in bytes for this image.

This is the number of bytes at the end of each line to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The x padding in bytes.

Implements [Image](#).

10.101.3.48 GetYOffset()

```
size_t GetYOffset ( ) const [virtual]
```

Gets the ROI y offset in pixels for this image.

This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y offset in pixels.

Implements [Image](#).

10.101.3.49 GetYPadding()

```
size_t GetYPadding ( ) const [virtual]
```

Gets the y padding in bytes for this image.

This is the number of bytes at the end of each image to facilitate alignment in buffers. This information is retrieved from the Transport Layer [Image](#) format headers. It is retrieved on a per image basis.

Returns

The y padding in bytes.

Implements [Image](#).

10.101.3.50 HasCRC()

```
bool HasCRC ( ) const [virtual]
```

Checks if the image contains ImageCRC checksum from chunk data.

Returns

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

Implements [Image](#).

10.101.3.51 IsCompressed()

```
bool IsCompressed ( ) const
```

Returns a boolean value indicating whether this image is compressed.

Returns

Returns true if image is compressed, false otherwise.

10.101.3.52 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 [Image](#).

10.101.3.53 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](#).

10.101.3.54 Release()

```
void Release ( ) [virtual]
```

Implements [IImage](#).

10.101.3.55 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](#).

10.101.3.56 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 [Image](#).

10.101.3.57 Save() [1/8]

```

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 [Image](#).

10.101.3.58 Save() [2/8]

```

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](#).

10.101.3.59 Save() [3/8]

```
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](#).

10.101.3.60 Save() [4/8]

```
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](#).

10.101.3.61 Save() [5/8]

```
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](#).

10.101.3.62 Save() [6/8]

```
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](#).

10.101.3.63 Save() [7/8]

```
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](#).

10.101.3.64 Save() [8/8]

```
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](#).

10.101.3.65 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\(\)](#)

10.101.4 Friends And Related Function Documentation

10.101.4.1 IDataStream

```
friend class IDataStream [friend]
```

10.101.4.2 ImageConverter

```
friend class ImageConverter [friend]
```

10.101.4.3 ImageFiler

```
friend class ImageFiler [friend]
```

10.101.4.4 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

10.101.4.5 ImageUtilityImpl

```
friend class ImageUtilityImpl [friend]
```

10.101.4.6 Stream

```
friend class Stream [friend]
```

10.101.5 Member Data Documentation

10.101.5.1 m_pImageData

```
ImageData* m_pImageData [protected]
```

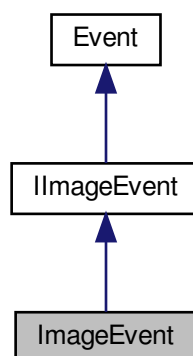
The documentation for this class was generated from the following file:

- [include/Image.h](#)

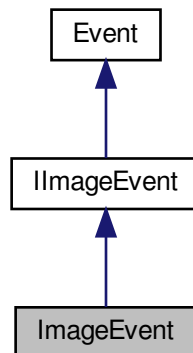
10.102 ImageEvent Class Reference

A handler for capturing image arrival events.

Inheritance diagram for ImageEvent:



Collaboration diagram for ImageEvent:



Public Member Functions

- [ImageEvent](#) ()
Default Constructor.
- virtual [~ImageEvent](#) ()
Virtual Destructor.
- virtual void [OnImageEvent](#) ([ImagePtr](#) image)=0
[Image](#) event callback.

Protected Member Functions

- [ImageEvent](#) & [operator=](#) (const [ImageEvent](#) &)
Assignment operator.

Additional Inherited Members

10.102.1 Detailed Description

A handler for capturing image arrival events.

10.102.2 Constructor & Destructor Documentation

10.102.2.1 ImageEvent()

```
ImageEvent ( )
```

Default Constructor.

10.102.2.2 ~ImageEvent()

```
virtual ~ImageEvent ( ) [virtual]
```

Virtual Destructor.

10.102.3 Member Function Documentation

10.102.3.1 OnImageEvent()

```
virtual void OnImageEvent (
    ImagePtr image ) [pure virtual]
```

[Image](#) event callback.

Parameters

| | |
|--------------|-------------------------------------|
| <i>image</i> | The ImagePtr object |
|--------------|-------------------------------------|

Implements [IImageEvent](#).

10.102.3.2 operator=()

```
ImageEvent& operator= (
    const ImageEvent & ) [protected]
```

Assignment operator.

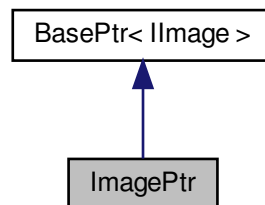
The documentation for this class was generated from the following file:

- [include/ImageEvent.h](#)

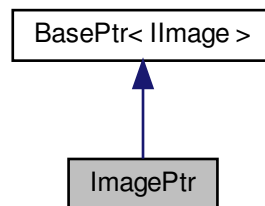
10.103 ImagePtr Class Reference

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

10.103.1 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.

10.103.2 Constructor & Destructor Documentation

10.103.2.1 ImagePtr() [1/4]

```
ImagePtr ( )
```

Default constructor.

10.103.2.2 ImagePtr() [2/4]

```
ImagePtr (
    const int )
```

Default constructor with argument.

10.103.2.3 ImagePtr() [3/4]

```
ImagePtr (
    const long )
```

Default constructor with argument.

10.103.2.4 ImagePtr() [4/4]

```
ImagePtr (
    const std::nullptr_t )
```

Default constructor with argument.

10.103.2.5 ~ImagePtr()

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

Virtual destructor.

10.103.3 Member Function Documentation

10.103.3.1 operator=()

```
virtual ImagePtr& operator= (
    const ImagePtr & ) [virtual]
```

Assignment operator.

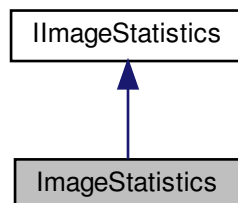
The documentation for this class was generated from the following file:

- include/[ImagePtr.h](#)

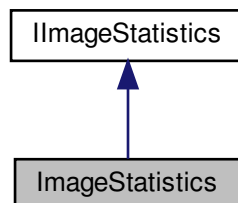
10.104 ImageStatistics Class Reference

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

10.104.1 Detailed Description

Represents image statistics for an image.

10.104.2 Constructor & Destructor Documentation

10.104.2.1 ImageStatistics() [1/2]

```
ImageStatistics ( )
```

Default constructor.

10.104.2.2 ~ImageStatistics()

```
virtual ~ImageStatistics ( ) [virtual]
```

Default destructor.

10.104.2.3 ImageStatistics() [2/2]

```
ImageStatistics (
    const ImageStatistics & other )
```

Copy constructor.

10.104.3 Member Function Documentation

10.104.3.1 DisableAll()

```
virtual void DisableAll ( ) [virtual]
```

Disable all channels.

Implements [IImageStatistics](#).

10.104.3.2 EnableAll()

```
virtual void EnableAll ( ) [virtual]
```

Enable all channels.

Implements [IImageStatistics](#).

10.104.3.3 EnableGreyOnly()

```
virtual void EnableGreyOnly ( ) [virtual]
```

Enable only the grey channel.

Implements [IImageStatistics](#).

10.104.3.4 EnableHSLOnly()

```
virtual void EnableHSLOnly ( ) [virtual]
```

Enable only the HSL channels.

Implements [IImageStatistics](#).

10.104.3.5 EnableRGBOnly()

```
virtual void EnableRGBOnly ( ) [virtual]
```

Enable only the RGB channels.

Implements [IImageStatistics](#).

10.104.3.6 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](#).

10.104.3.7 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](#).

10.104.3.8 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](#).

10.104.3.9 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](#).

10.104.3.10 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](#).

10.104.3.11 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](#).

10.104.3.12 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](#).

10.104.3.13 operator=()

```

ImageStatistics& operator= (
    const ImageStatistics & other )

```

Assignment operator.

Parameters

| | |
|--------------|--|
| <i>other</i> | The ImageStatistics object to copy from. |
|--------------|--|

10.104.3.14 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](#).

10.104.4 Friends And Related Function Documentation

10.104.4.1 ImageStatsCalculator

```
friend class ImageStatsCalculator [friend]
```

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

- include/[ImageStatistics.h](#)

10.105 ImageUtility Class Reference

Static helper functions for the image object class.

Public Types

- enum [ImageScalingAlgorithm](#) { NEAREST_NEIGHBOR }
Image scaling algorithms.

Static Public Member Functions

- static [ImagePtr](#) [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static void [CreateScaled](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, [ImageScalingAlgorithm](#) scalingAlg, double scalingFactor)
Computes a scaled image using the specified parameters.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const [PixelFormatEnums](#) destPixelFormat)
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max)
Computes a normalized image.
- static [ImagePtr](#) [CreateNormalized](#) (const [ImagePtr](#) &srcImage, const double min, const double max, const [PixelFormatEnums](#) destPixelFormat)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage)
Computes a normalized image.
- static void [CreateNormalized](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage, const double min, const double max)
Computes a normalized image.

10.105.1 Detailed Description

Static helper functions for the image object class.

10.105.2 Member Enumeration Documentation

10.105.2.1 ImageScalingAlgorithm

```
enum ImageScalingAlgorithm
```

[Image](#) scaling algorithms.

Enumerator

| | |
|------------------|--|
| NEAREST_NEIGHBOR | |
|------------------|--|

10.105.3 Member Function Documentation

10.105.3.1 CreateNormalized() [1/5]

```
static ImagePtr CreateNormalized (  
    const ImagePtr & srcImage,  
    const PixelFormatEnums destPixelFormat ) [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 |

Returns

The normalized image

10.105.3.2 CreateNormalized() [2/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max ) [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 |

Returns

The normalized image

10.105.3.3 CreateNormalized() [3/5]

```
static ImagePtr CreateNormalized (
    const ImagePtr & srcImage,
    const double min,
    const double max,
    const PixelFormatEnums destPixelFormat ) [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 |

Returns

The normalized image

10.105.3.4 CreateNormalized() [4/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage ) [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 |

10.105.3.5 CreateNormalized() [5/5]

```
static void CreateNormalized (
    const ImagePtr & srcImage,
    ImagePtr & destImage,
    const double min,
    const double max ) [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 |

10.105.3.6 CreateScaled() [1/2]

```
static ImagePtr CreateScaled (
    const ImagePtr & srcImage,
    ImageScalingAlgorithm scalingAlg,
    double scalingFactor ) [static]
```

Computes a scaled image using the specified parameters.

Does not support scaling of raw bayer images.

Parameters

| | |
|----------------------|--|
| <i>srcImage</i> | The source image from which to create scaled image |
| <i>scalingAlg</i> | The desired image scaling algorithm to use |
| <i>scalingFactor</i> | The desired image scaling factor to use |

Returns

The scaled image

10.105.3.7 CreateScaled() [2/2]

```
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 |

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

- [include/ImageUtility.h](#)

10.106 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](#) &srcImage)
Computes a heatmap image.
- static void [CreateHeatmap](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destImage)
Computes a heatmap image.
- static void [SetHeatmapColorGradient](#) (const [HeatmapColor](#) newLowColor, const [HeatmapColor](#) newHighColor)
Sets the heatmap gradient color vector to the new desired range between HEATMAP_BLACK and HEATMAP_WHITE.
- static void [GetHeatmapColorGradient](#) ([HeatmapColor](#) ¤tLowColor, [HeatmapColor](#) ¤tHighColor)
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 ¤tLowValue, unsigned int ¤tHighValue)
Returns the current high and low values used in heatmap representations.

10.106.1 Detailed Description

Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

10.106.2 Member Enumeration Documentation

10.106.2.1 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 | |

10.106.3 Member Function Documentation

10.106.3.1 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

10.106.3.2 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\(\)](#)

10.106.3.3 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\(\)](#)

10.106.3.4 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\(\)](#)

10.106.3.5 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. |

10.106.3.6 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

| | |
|---------------------|---|
| <i>newLowValue</i> | New value at which to begin color representation. |
| <i>newHighValue</i> | New value at which to end color representation. |

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

- [include/ImageUtilityHeatmap.h](#)

10.107 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](#) &srcImage, const [PolarizationQuadrant](#) desiredQuadrant)

Extracts all pixels of a specified degree of linear polarization into a new image object.

- static void [ExtractPolarQuadrant](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destQuadImage, const [PolarizationQuadrant](#) desiredQuadrant)
Extracts all pixels of a specified degree of linear polarization into the provided image object.
- static [ImagePtr](#) [CreateStokesS0](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the overall intensity of light from a polarized image.
- static void [CreateStokesS0](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS0Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the overall intensity of light from a polarized image.
- static [ImagePtr](#) [CreateStokesS1](#) (const [ImagePtr](#) &srcImage, 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](#) &srcImage, [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](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers at 45 and -45 to the horizontal.
- static void [CreateStokesS2](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destStokesS2Image, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the difference in intensity accepted through the polarizers.
- static [ImagePtr](#) [CreateDolp](#) (const [ImagePtr](#) &srcImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the fraction of incident light intensity in the linear polarization states.
- static void [CreateDolp](#) (const [ImagePtr](#) &srcImage, [ImagePtr](#) &destDolpImage, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the fraction of incident light intensity in the linear polarization states.
- static [ImagePtr](#) [CreateAolp](#) (const [ImagePtr](#) &srcImage, 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](#) &srcImage, [ImagePtr](#) &destAolpImg, const [ColorProcessingAlgorithm](#) colorProcessingAlg=DEFAULT)
Computes an image representing the angle at which linearly polarized light oscillates with respect to a reference axis.

10.107.1 Detailed Description

Static functions to create polarization images from image objects of pixel format Polarized8 and BayerRGPolarized8.

10.107.2 Member Enumeration Documentation

10.107.2.1 PolarizationQuadrant

enum [PolarizationQuadrant](#)

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

Enumerator

| | |
|---------------|---------------------------------|
| QUADRANT_I0 | The 0 degree of polarization. |
| QUADRANT_I45 | The 45 degree of polarization. |
| QUADRANT_I90 | The 90 degree of polarization. |
| QUADRANT_I135 | The 135 degree of polarization. |

10.107.3 Member Function Documentation

10.107.3.1 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 (dolp) image

10.107.3.2 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 |

10.107.3.3 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

10.107.3.4 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 |

10.107.3.5 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

10.107.3.6 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 |

10.107.3.7 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

10.107.3.8 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 |

10.107.3.9 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

10.107.3.10 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 |

10.107.3.11 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

10.107.3.12 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](#)

10.108 int64_autovector_t Class Reference

Vector of integers with reference counting.

Public Member Functions

- [int64_autovector_t\(\)](#)
- [int64_autovector_t\(const int64_autovector_t &obj\)](#)
- [int64_autovector_t\(size_t n\)](#)
- [virtual ~int64_autovector_t\(void\)](#)
- [int64_autovector_t & operator= \(const int64_autovector_t &obj\)](#)
- [void operator delete \(void *pWhere\)](#)
- [void * operator new \(size_t uiSize\)](#)
- [int64_t & operator\[\] \(size_t uiIndex\)](#)
- [const int64_t & operator\[\] \(size_t uiIndex\) const](#)
- [size_t size \(\) const](#)

Protected Attributes

- [std::vector< int64_t > * _pv](#)
- [ATOMIC_VARIABLE * _pCount](#)

10.108.1 Detailed Description

Vector of integers with reference counting.

10.108.2 Constructor & Destructor Documentation

10.108.2.1 `int64_autovector_t()` [1/3]

```
int64_autovector_t ( )
```

10.108.2.2 `int64_autovector_t()` [2/3]

```
int64_autovector_t (
    const int64_autovector_t & obj )
```

10.108.2.3 `int64_autovector_t()` [3/3]

```
int64_autovector_t (
    size_t n ) [explicit]
```

10.108.2.4 `~int64_autovector_t()`

```
virtual ~int64_autovector_t (
    void ) [virtual]
```

10.108.3 Member Function Documentation

10.108.3.1 `operator delete()`

```
void operator delete (
    void * pWhere )
```

10.108.3.2 `operator new()`

```
void* operator new (
    size_t uiSize )
```

10.108.3.3 operator=()

```
int64_autovector_t& operator= (
    const int64_autovector_t & obj )
```

10.108.3.4 operator[]() [1/2]

```
int64_t& operator[] (
    size_t uiIndex )
```

10.108.3.5 operator[]() [2/2]

```
const int64_t& operator[] (
    size_t uiIndex ) const
```

10.108.3.6 size()

```
size_t size ( ) const
```

10.108.4 Member Data Documentation

10.108.4.1 _pCount

```
ATOMIC_VARIABLE* _pCount [protected]
```

10.108.4.2 _pv

```
std::vector<int64_t>* _pv [protected]
```

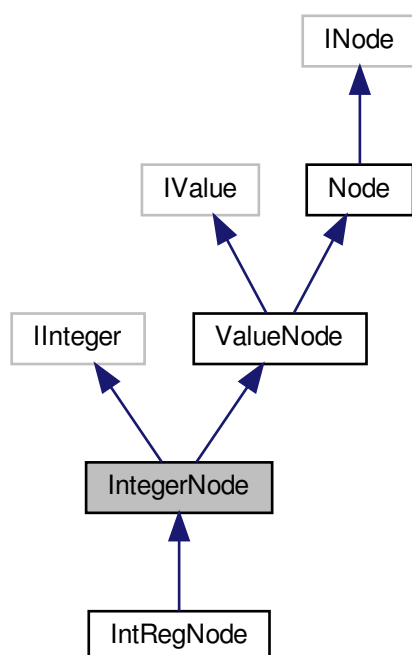
The documentation for this class was generated from the following file:

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

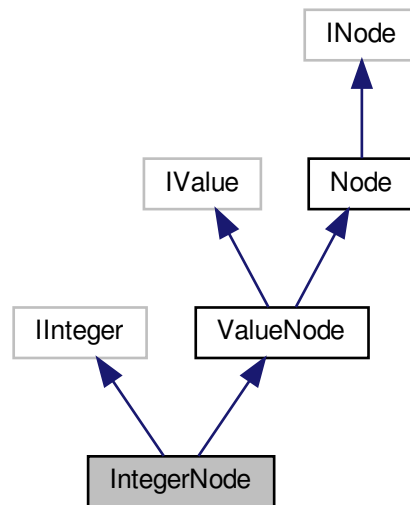
10.109 IntegerNode Class Reference

[Interface](#) for string properties.

Inheritance diagram for IntegerNode:



Collaboration diagram for IntegerNode:



Public Member Functions

- [IntegerNode](#) ()
- [IntegerNode](#) (std::shared_ptr< Node::NodeImpl > pInteger)
- virtual [~IntegerNode](#) ()
- virtual void [SetValue](#) (int64_t Value, bool [Verify](#)=true)
 - Set node value.*
- virtual [IInteger](#) & [operator=](#) (int64_t Value)
 - Set node value.*
- virtual int64_t [GetValue](#) (bool [Verify](#)=false, bool IgnoreCache=false)
 - Get node value.*
- virtual int64_t [operator\(\)](#) ()
 - Get node value.*
- virtual int64_t [operator*](#) ()
 - Get node value.*
- virtual int64_t [GetMin](#) ()
 - Get minimum value allowed.*
- virtual int64_t [GetMax](#) ()
 - Get maximum value allowed.*
- virtual [EIncMode](#) [GetIncMode](#) ()
 - Get increment mode.*
- virtual int64_t [GetInc](#) ()
 - Get increment.*
- virtual [int64_autovector_t](#) [GetListOfValidValues](#) (bool bounded=true)
 - Get list of valid value.*
- virtual [ERepresentation](#) [GetRepresentation](#) ()
 - Get recommended representation.*

- virtual [GenlCam::gcstring](#) [GetUnit](#) ()
Get the physical unit name.
- virtual [IFloat](#) * [GetFloatAlias](#) ()
gets the interface of an alias node.
- virtual void [ImposeMin](#) (int64_t Value)
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)
Restrict maximum value.
- virtual void [SetReference](#) ([INode](#) *pBase)
overload SetReference for Integer

Additional Inherited Members

10.109.1 Detailed Description

[Interface](#) for string properties.

10.109.2 Constructor & Destructor Documentation

10.109.2.1 [IntegerNode](#)() [1/2]

[IntegerNode](#) ()

10.109.2.2 [IntegerNode](#)() [2/2]

[IntegerNode](#) (
 std::shared_ptr< [Node::NodeImpl](#) > *pInteger*)

10.109.2.3 [~IntegerNode](#)()

virtual [~IntegerNode](#) () [virtual]

10.109.3 Member Function Documentation

10.109.3.1 GetFloatAlias()

```
virtual IFloat* GetFloatAlias ( ) [virtual]
```

gets the interface of an alias node.

10.109.3.2 GetInc()

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

Get increment.

10.109.3.3 GetIncMode()

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

Get increment mode.

10.109.3.4 GetListOfValidValues()

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

Get list of valid value.

10.109.3.5 GetMax()

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

Get maximum value allowed.

10.109.3.6 GetMin()

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

Get minimum value allowed.

10.109.3.7 GetRepresentation()

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

Get recommended representation.

10.109.3.8 GetUnit()

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

Get the physical unit name.

10.109.3.9 GetValue()

```
virtual int64_t GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

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

Returns

The value read

10.109.3.10 ImposeMax()

```
virtual void ImposeMax (
    int64_t Value ) [virtual]
```

Restrict maximum value.

10.109.3.11 ImposeMin()

```
virtual void ImposeMin (
    int64_t Value ) [virtual]
```

Restrict minimum value.

10.109.3.12 operator()()

```
virtual int64_t operator() ( ) [virtual]
```

Get node value.

10.109.3.13 operator*()

```
virtual int64_t operator* ( ) [virtual]
```

Get node value.

10.109.3.14 operator=()

```
virtual Integer& operator= (
    int64_t Value ) [virtual]
```

Set node value.

10.109.3.15 SetReference()

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

overload SetReference for Integer

Reimplemented from [ValueNode](#).

Reimplemented in [IntRegNode](#).

10.109.3.16 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) |

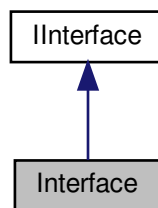
The documentation for this class was generated from the following file:

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

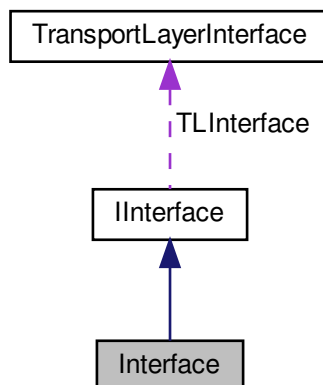
10.110 Interface Class Reference

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 [RegisterEvent](#) ([Event](#) &evtToRegister)

Registers an event for the interface.

- void [UnregisterEvent](#) ([Event](#) &evtToUnregister)

Unregisters an event 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.

Friends

- class [InterfaceInternal](#)

Additional Inherited Members

10.110.1 Detailed Description

An interface object which holds a list of cameras.

10.110.2 Constructor & Destructor Documentation

10.110.2.1 ~Interface()

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

Virtual Destructor.

10.110.3 Member Function Documentation

10.110.3.1 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](#).

10.110.3.2 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](#).

10.110.3.3 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](#).

10.110.3.4 RegisterEvent()

```
void RegisterEvent (
    Event & evtToRegister ) [virtual]
```

Registers an event for the interface.

Parameters

| | |
|----------------------|---|
| <i>evtToRegister</i> | The event to register for the interface |
|----------------------|---|

Implements [IInterface](#).

10.110.3.5 SendActionCommand()

```
void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) const [virtual]
```

Broadcast an Action Command to all devices on interface.

Parameters

| | |
|--------------------|--|
| <i>deviceKey</i> | The Action Command's device key |
| <i>groupKey</i> | The Action Command's group key |
| <i>groupMask</i> | The Action Command's group mask |
| <i>actionTime</i> | (Optional) Time when to assert a future action. Zero means immediate action. |
| <i>pResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted. |
| <i>results</i> | (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

Implements [IInterface](#).

10.110.3.6 UnregisterEvent()

```
void UnregisterEvent (
    Event & evtToUnregister ) [virtual]
```

Unregisters an event for the interface.

Parameters

| | |
|------------------------|--|
| <i>evtToUnregister</i> | The event to unregister from the interface |
|------------------------|--|

Implements [IInterface](#).

10.110.3.7 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](#).

10.110.4 Friends And Related Function Documentation

10.110.4.1 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

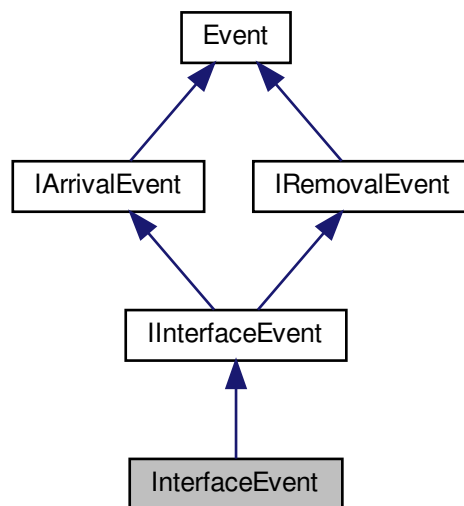
The documentation for this class was generated from the following file:

- [include/Interface.h](#)

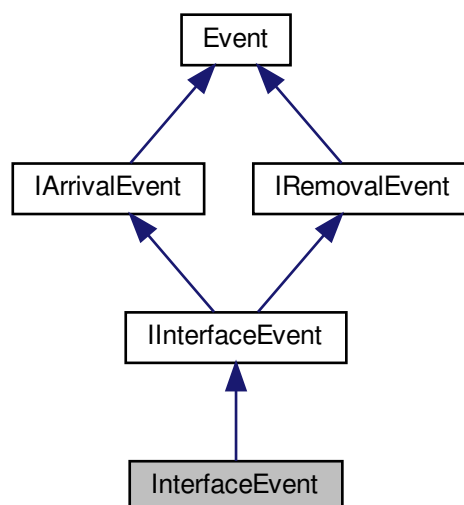
10.111 InterfaceEvent Class Reference

A handler to device arrival and removal events on all interfaces.

Inheritance diagram for InterfaceEvent:



Collaboration diagram for InterfaceEvent:



Public Member Functions

- [InterfaceEvent](#) ()
Default constructor.
- virtual [~InterfaceEvent](#) ()
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

- [InterfaceEvent](#) & [operator=](#) (const [InterfaceEvent](#) &)
Assignment operator.

Additional Inherited Members

10.111.1 Detailed Description

A handler to device arrival and removal events on all interfaces.

10.111.2 Constructor & Destructor Documentation

10.111.2.1 [InterfaceEvent](#)()

```
InterfaceEvent ( )
```

Default constructor.

10.111.2.2 [~InterfaceEvent](#)()

```
virtual ~InterfaceEvent ( ) [virtual]
```

Virtual destructor.

10.111.3 Member Function Documentation

10.111.3.1 OnDeviceArrival()

```
virtual void OnDeviceArrival (
    uint64_t serialNumber ) [pure virtual]
```

Device arrival event callback.

Implements [InterfaceEvent](#).

10.111.3.2 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Callback to the device removal event.

Parameters

| | |
|---------------------|---|
| <i>serialNumber</i> | The serial number of the removed device |
|---------------------|---|

Implements [InterfaceEvent](#).

10.111.3.3 operator=()

```
InterfaceEvent& operator= (
    const InterfaceEvent & ) [protected]
```

Assignment operator.

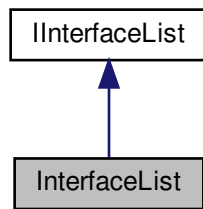
The documentation for this class was generated from the following file:

- [include/InterfaceEvent.h](#)

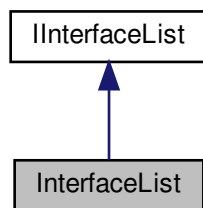
10.112 InterfaceList Class Reference

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

10.112.1 Detailed Description

A list of the available interfaces on the system.

10.112.2 Constructor & Destructor Documentation

10.112.2.1 InterfaceList() [1/2]

```
InterfaceList (
    void )
```

10.112.2.2 ~InterfaceList()

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

10.112.2.3 InterfaceList() [2/2]

```
InterfaceList (
    const InterfaceList & iface )
```

10.112.3 Member Function Documentation

10.112.3.1 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](#).

10.112.3.2 GetByIndex()

```
InterfacePtr GetByIndex (
    unsigned int index ) const [virtual]
```

Returns a pointer to an [Interface](#) object at the "index".

Parameters

| | |
|--------------|---|
| <i>index</i> | The index at which to retrieve the Interface object |
|--------------|---|

Returns

A pointer to an [Interface](#) object.

Implements [IInterfaceList](#).

10.112.3.3 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](#).

10.112.3.4 operator=()

```
InterfaceList& operator= (
    const InterfaceList & iface )
```

Assignment operator.

10.112.3.5 operator[]()

```
InterfacePtr operator[] (
    unsigned int index ) [virtual]
```

Array subscription operators.

Implements [IInterfaceList](#).

10.112.4 Friends And Related Function Documentation

10.112.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

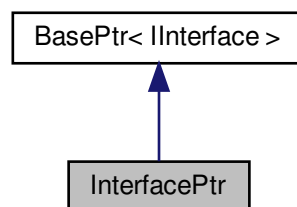
The documentation for this class was generated from the following file:

- include/[InterfaceList.h](#)

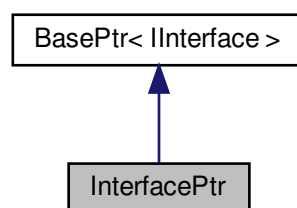
10.113 InterfacePtr Class Reference

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

10.113.1 Detailed Description

A reference tracked pointer to the interface object.

10.113.2 Constructor & Destructor Documentation

10.113.2.1 InterfacePtr() [1/4]

```
InterfacePtr ( ) throw ( ) [inline]
```

Default Constructor.

10.113.2.2 InterfacePtr() [2/4]

```
InterfacePtr (
    const int ) throw ( ) [inline]
```

Default Constructor with argument.

10.113.2.3 InterfacePtr() [3/4]

```
InterfacePtr (
    const long ) throw ( ) [inline]
```

10.113.2.4 InterfacePtr() [4/4]

```
InterfacePtr (
    const std::nullptr_t ) throw ( ) [inline]
```

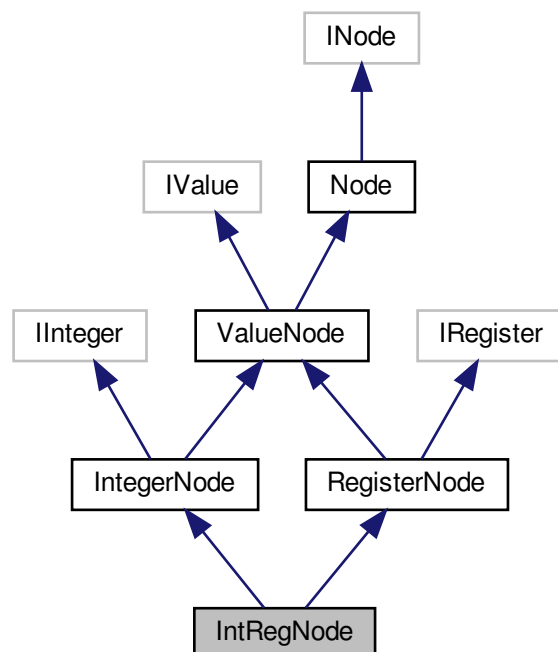
The documentation for this class was generated from the following file:

- [include/InterfacePtr.h](#)

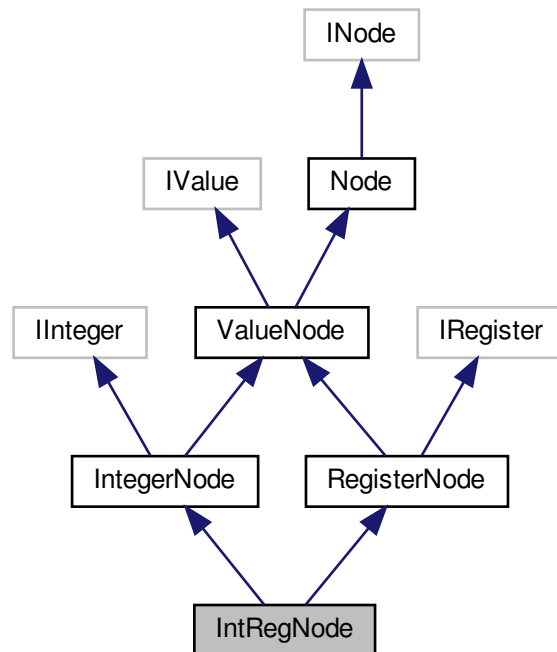
10.114 IntRegNode Class Reference

[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

10.114.1 Detailed Description

[Interface](#) for string properties.

10.114.2 Constructor & Destructor Documentation

10.114.2.1 IntRegNode() [1/2]

[IntRegNode](#) ()

10.114.2.2 IntRegNode() [2/2]

[IntRegNode](#) (
 std::shared_ptr< Node::NodeImpl > *pInteger*)

10.114.2.3 ~IntRegNode()

virtual ~[IntRegNode](#) () [virtual]

10.114.3 Member Function Documentation

10.114.3.1 SetReference()

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

overload SetReference for Value

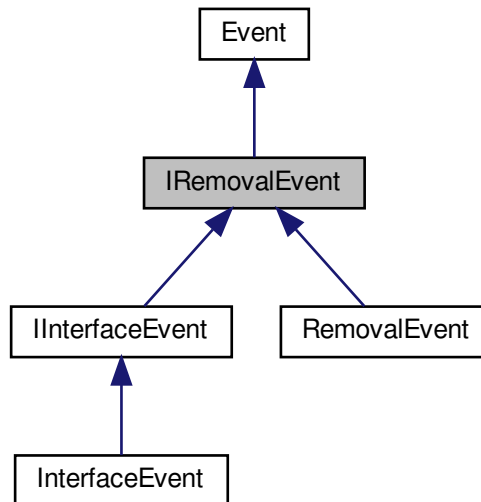
Reimplemented from [IntegerNode](#).

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

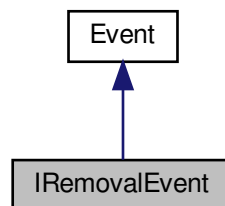
- include/SpinGenApi/[IntRegNode.h](#)

10.115 IRemovalEvent Class Reference

Inheritance diagram for IRemovalEvent:



Collaboration diagram for IRemovalEvent:



Public Member Functions

- virtual [~IRemovalEvent](#) ()
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0

Protected Member Functions

- [IRemovalEvent](#) ()
- [IRemovalEvent](#) (const [IRemovalEvent](#) &)
- [IRemovalEvent](#) & [operator=](#) (const [IRemovalEvent](#) &)

Additional Inherited Members

10.115.1 Constructor & Destructor Documentation

10.115.1.1 ~IRemovalEvent()

```
virtual ~IRemovalEvent ( ) [inline], [virtual]
```

10.115.1.2 IRemovalEvent() [1/2]

```
IRemovalEvent ( ) [inline], [protected]
```

10.115.1.3 IRemovalEvent() [2/2]

```
IRemovalEvent (
    const IRemovalEvent & ) [inline], [protected]
```

10.115.2 Member Function Documentation

10.115.2.1 OnDeviceRemoval()

```
virtual void OnDeviceRemoval (
    uint64_t serialNumber ) [pure virtual]
```

Implemented in [InterfaceEvent](#), [RemovalEvent](#), and [IInterfaceEvent](#).

10.115.2.2 operator=()

```
IRemovalEvent& operator= (
    const IRemovalEvent & ) [protected]
```

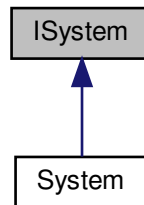
The documentation for this class was generated from the following file:

- [include/Interface/IRemovalEvent.h](#)

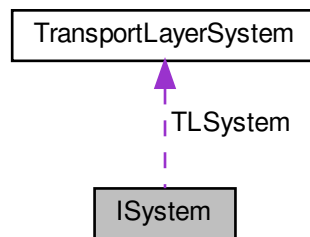
10.116 ISystem Class Reference

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 [RegisterInterfaceEvent](#) ([Event](#) &evtToRegister, bool updateInterface=true)=0
- virtual void [UnregisterInterfaceEvent](#) ([Event](#) &evtToUnregister)=0
- virtual void [RegisterLoggingEvent](#) ([LoggingEvent](#) &handler)=0
- virtual void [UnregisterAllLoggingEvent](#) ()=0
- virtual void [UnregisterLoggingEvent](#) ([LoggingEvent](#) &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](#)

10.116.1 Detailed Description

The interface file for [System](#).

10.116.2 Constructor & Destructor Documentation

10.116.2.1 [~ISystem\(\)](#)

```
virtual ~ISystem ( ) [inline], [virtual]
```

10.116.2.2 [ISystem\(\)](#) [1/2]

```
ISystem ( ) [inline], [protected]
```

10.116.2.3 [ISystem\(\)](#) [2/2]

```
ISystem (  
    const ISystem & ) [inline], [protected]
```

10.116.3 Member Function Documentation

10.116.3.1 GetCameras()

```
virtual CameraList GetCameras (
    bool updateInterfaces = true,
    bool updateCameras = true ) [pure virtual]
```

Implemented in [System](#).

10.116.3.2 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

10.116.3.3 GetLibraryVersion()

```
virtual const LibraryVersion GetLibraryVersion ( ) [pure virtual]
```

Implemented in [System](#).

10.116.3.4 GetLoggingEventPriorityLevel()

```
virtual SpinnakerLogLevel GetLoggingEventPriorityLevel ( ) [pure virtual]
```

Implemented in [System](#).

10.116.3.5 GetTLNodeMap()

```
virtual GenApi::INodeMap& GetTLNodeMap ( ) const [pure virtual]
```

Implemented in [System](#).

10.116.3.6 IsInUse()

```
virtual bool IsInUse ( ) [pure virtual]
```

Implemented in [System](#).

10.116.3.7 operator=()

```
ISystem& operator= (
    const ISystem & ) [protected]
```

10.116.3.8 RegisterInterfaceEvent()

```
virtual void RegisterInterfaceEvent (
    Event & evtToRegister,
    bool updateInterface = true ) [pure virtual]
```

Implemented in [System](#).

10.116.3.9 RegisterLoggingEvent()

```
virtual void RegisterLoggingEvent (
    LoggingEvent & handler ) [pure virtual]
```

Implemented in [System](#).

10.116.3.10 ReleaseInstance()

```
virtual void ReleaseInstance ( ) [pure virtual]
```

Implemented in [System](#).

10.116.3.11 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [pure virtual]
```

Implemented in [System](#).

10.116.3.12 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [pure virtual]
```

Implemented in [System](#).

10.116.3.13 UnregisterAllLoggingEvent()

```
virtual void UnregisterAllLoggingEvent ( ) [pure virtual]
```

Implemented in [System](#).

10.116.3.14 UnregisterInterfaceEvent()

```
virtual void UnregisterInterfaceEvent (
    Event & evtToUnregister ) [pure virtual]
```

Implemented in [System](#).

10.116.3.15 UnregisterLoggingEvent()

```
virtual void UnregisterLoggingEvent (
    LoggingEvent & handler ) [pure virtual]
```

Implemented in [System](#).

10.116.3.16 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [pure virtual]
```

Implemented in [System](#).

10.116.3.17 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [pure virtual]
```

Implemented in [System](#).

10.116.4 Friends And Related Function Documentation

10.116.4.1 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

10.116.5 Member Data Documentation

10.116.5.1 TLSYSTEM

[TransportLayerSystem](#) `TLSYSTEM`

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

- [include/Interface/ISystem.h](#)

10.117 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.

10.117.1 Detailed Description

Options for saving JPEG image.

10.117.2 Constructor & Destructor Documentation

10.117.2.1 JPEGOption()

```
JPEGOption ( ) [inline]
```

10.117.3 Member Data Documentation

10.117.3.1 progressive

```
bool progressive
```

Whether to save as a progressive JPEG file.

10.117.3.2 quality

```
unsigned int quality
```

JPEG image quality in range (0-100).

- 100 - Superb quality.
- 75 - Good quality.
- 50 - Normal quality.
- 10 - Poor quality.

10.117.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.118 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.

10.118.1 Detailed Description

Options for saving JPEG2000 image.

10.118.2 Constructor & Destructor Documentation

10.118.2.1 JPG2Option()

```
JPG2Option ( ) [inline]
```

10.118.3 Member Data Documentation

10.118.3.1 quality

```
unsigned int quality
```

JPEG saving quality in range (1-512).

10.118.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- include/[SpinnakerDefs.h](#)

10.119 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.

10.119.1 Detailed Description

Provides easier access to the current version of [Spinnaker](#).

10.119.2 Member Data Documentation

10.119.2.1 build

```
unsigned int build
```

Build number of the library.

10.119.2.2 major

unsigned int major

Major version of the library.

10.119.2.3 minor

unsigned int minor

Minor version of the library.

10.119.2.4 type

unsigned int type

Version type of the library.

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

- [include/SpinnakerDefs.h](#)

10.120 LockableObject< Object >::Lock Class Reference

A scopelevel [Lock](#) class.

Public Member Functions

- [Lock](#) (const [LockableObject](#)< Object > &obj)
- [~Lock](#) ()

10.120.1 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.

10.120.2 Constructor & Destructor Documentation

10.120.2.1 Lock()

```
Lock (
    const LockableObject< Object > & obj ) [inline]
```

10.120.2.2 ~Lock()

```
~Lock ( ) [inline]
```

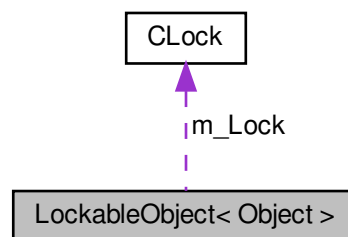
The documentation for this class was generated from the following file:

- include/SpinGenApi/GCSynch.h

10.121 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](#)

10.121.1 Detailed Description

```
template<class Object>
class Spinnaker::GenICam::LockableObject< Object >
```

Instance-Lock for an object.

10.121.2 Member Function Documentation

10.121.2.1 GetLock()

```
Lock GetLock ( ) const [inline]
```

Get a new lock.

10.121.3 Friends And Related Function Documentation

10.121.3.1 Lock

```
friend class Lock [friend]
```

10.121.4 Member Data Documentation

10.121.4.1 m_Lock

`CLock m_Lock [mutable]`

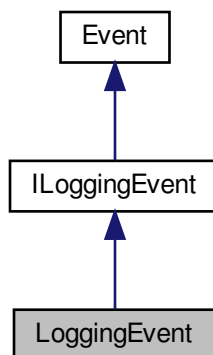
The documentation for this class was generated from the following file:

- `include/SpinGenApi/GCSynch.h`

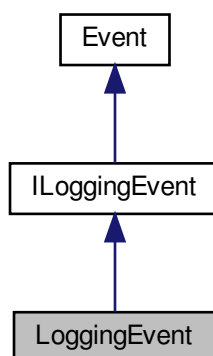
10.122 LoggingEvent Class Reference

An event handler for capturing the device logging event.

Inheritance diagram for LoggingEvent:



Collaboration diagram for LoggingEvent:



Public Member Functions

- [LoggingEvent](#) ()
Default constructor.
- [~LoggingEvent](#) ()
Virtual destructor.
- virtual void [OnLogEvent](#) ([LoggingEventDataPtr](#) eventPtr)=0
The callback for the log event.

Protected Member Functions

- [LoggingEvent](#) & [operator=](#) (const [LoggingEvent](#) &)
Assignment operator.

Additional Inherited Members

10.122.1 Detailed Description

An event handler for capturing the device logging event.

10.122.2 Constructor & Destructor Documentation

10.122.2.1 [LoggingEvent](#)()

```
LoggingEvent ( )
```

Default constructor.

10.122.2.2 [~LoggingEvent](#)()

```
~LoggingEvent ( )
```

Virtual destructor.

10.122.3 Member Function Documentation

10.122.3.1 [OnLogEvent](#)()

```
virtual void OnLogEvent (  
    LoggingEventDataPtr eventPtr ) [pure virtual]
```

The callback for the log event.

Parameters

| | |
|-----------------|---------------------------|
| <i>eventPtr</i> | The logging event pointer |
|-----------------|---------------------------|

Implements [ILoggingEvent](#).

10.122.3.2 operator=()

```
LoggingEvent& operator= (
    const LoggingEvent & ) [protected]
```

Assignment operator.

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

- include/[LoggingEvent.h](#)

10.123 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](#)

10.123.1 Detailed Description

The [LoggingEventData](#) object.

10.123.2 Constructor & Destructor Documentation

10.123.2.1 ~LoggingEventData()

```
~LoggingEventData ( )
```

Default Destructor.

10.123.2.2 LoggingEventData()

```
LoggingEventData (
    void * data ) [protected]
```

Default Constructor.

10.123.3 Member Function Documentation

10.123.3.1 GetCategoryName()

```
const char* GetCategoryName ( )
```

Gets the logging event category name.

Returns

The category name

10.123.3.2 GetLogMessage()

```
const char* GetLogMessage ( )
```

Gets the logging event message.

Returns

The log message

10.123.3.3 GetNDC()

```
const char* GetNDC ( )
```

Gets the logging event's Nested Diagnostic Context (NDC).

Returns

The log event's NDC

10.123.3.4 GetPriority()

```
const int GetPriority ( )
```

Gets the logging event priority.

Returns

The log priority

10.123.3.5 GetPriorityName()

```
const char* GetPriorityName ( )
```

Gets the logging event priority name.

Returns

The priority name of the log

10.123.3.6 GetThreadName()

```
const char* GetThreadName ( )
```

Gets the logging event thread name.

Returns

The thread name

10.123.3.7 GetTimestamp()

```
const char* GetTimestamp ( )
```

Gets the logging event time stamp.

Returns

The time stamp of the log

10.123.4 Friends And Related Function Documentation

10.123.4.1 SystemImpl

```
friend class SystemImpl [friend]
```

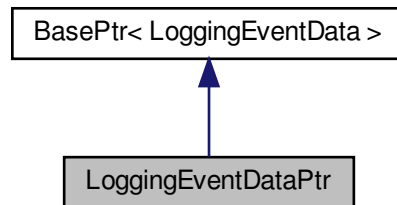
The documentation for this class was generated from the following file:

- [include/LoggingEventData.h](#)

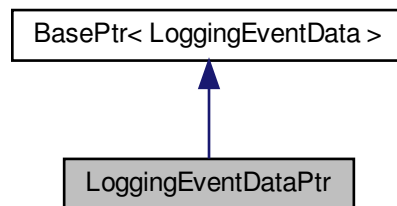
10.124 LoggingEventDataPtr Class Reference

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

10.124.1 Detailed Description

A reference tracked pointer to the [LoggingEvent](#) object.

10.124.2 Constructor & Destructor Documentation

10.124.2.1 LoggingEventDataPtr() [1/4]

```
LoggingEventDataPtr ( ) throw ) [inline]
```

Default Constructor.

10.124.2.2 LoggingEventDataPtr() [2/4]

```
LoggingEventDataPtr (
    const int ) throw ) [inline]
```

Default Constructor with argument.

10.124.2.3 LoggingEventDataPtr() [3/4]

```
LoggingEventDataPtr (
    const long ) throw ) [inline]
```

Default Constructor with argument.

10.124.2.4 LoggingEventDataPtr() [4/4]

```
LoggingEventDataPtr (
    const std::nullptr_t ) throw ) [inline]
```

Default Constructor with argument.

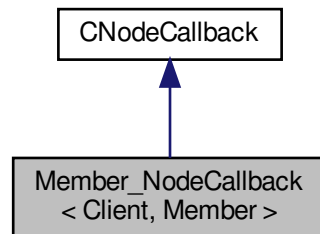
The documentation for this class was generated from the following file:

- [include/LoggingEventDataPtr.h](#)

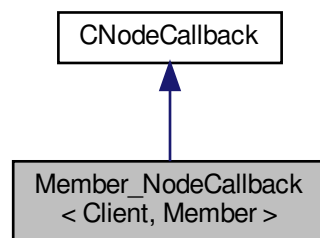
10.125 Member_NodeCallback< Client, Member > Class Template Reference

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

10.125.1 Detailed Description

```
template<class Client, class Member>
class Spinnaker::GenApi::Member_NodeCallback< Client, Member >
```

Container for a member function pointer.

10.125.2 Member Typedef Documentation

10.125.2.1 PMEMBERFUNC

```
typedef void(Client::* PMEMBERFUNC) (INode *)
```

Member function type.

10.125.3 Constructor & Destructor Documentation

10.125.3.1 Member_NodeCallback()

```
Member_NodeCallback (
    INode * pNode,
    Client & client,
    Member member,
    ECallbackType CallbackType ) [inline]
```

Constructor.

10.125.4 Member Function Documentation

10.125.4.1 Destroy()

```
virtual void Destroy ( ) [inline], [virtual]
```

destroys the object

Implements [CNodeCallback](#).

10.125.4.2 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](#)

10.126 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]

10.126.1 Detailed Description

Options for saving MJPG files.

10.126.2 Constructor & Destructor Documentation

10.126.2.1 MJPGOption()

```
MJPGOption ( ) [inline]
```

10.126.3 Member Data Documentation

10.126.3.1 frameRate

```
float frameRate
```

Frame rate of the stream.

10.126.3.2 quality

```
unsigned int quality
```

[Image](#) quality (1-100)

10.126.3.3 reserved

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

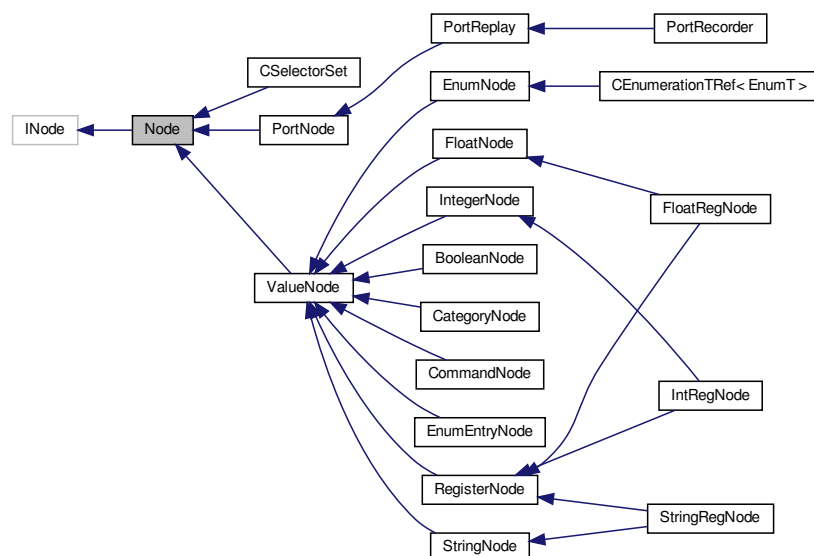
The documentation for this struct was generated from the following file:

- [include/SpinVideoDefs.h](#)

10.127 Node Class Reference

class common to all nodes

Inheritance diagram for Node:



Collaboration diagram for Node:



Public Member Functions

- [Node](#) ()
Constructor.
- [Node](#) (std::shared_ptr< Node::NodeImpl > pNodeHandle)
Constructor.
- [~Node](#) ()
Destructor.
- virtual [GenICam::gcstring](#) [GetName](#) (bool FullQualified=false) const
Get node name.
- virtual [GenApi::ENamespace](#) [GetNameSpace](#) () const
Get name space.
- virtual [EVisibility](#) [GetVisibility](#) () const
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const
Is the node value cacheable.
- virtual [EYesNo](#) [IsAccessModeCacheable](#) () const
True if the AccessMode can be cached.
- virtual [ECachingMode](#) [GetCachingMode](#) () const
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const
recommended polling time (for not cacheable nodes)
- virtual [GenICam::gcstring](#) [GetToolTip](#) () const
Get a short description of the node.
- virtual [GenICam::gcstring](#) [GetDescription](#) () const
Get a long description of the node.
- virtual [GenICam::gcstring](#) [GetDisplayName](#) () const
Get a name string for display.
- virtual [GenICam::gcstring](#) [GetDeviceName](#) () const
Get a name of the device.
- virtual void [GetChildren](#) (GenApi::NodeList_t &Children, [ELinkType](#) LinkType=ctReadingChildren) const
Get all nodes this node directly depends on.
- virtual void [GetParents](#) (GenApi::NodeList_t &Parents) const

- Gets all nodes this node is directly depending on.*

 - virtual [CallbackHandleType RegisterCallback](#) ([CNodeCallback](#) *pCallback)

Register change callback Takes ownership of the [CNodeCallback](#) object.
- virtual bool [DeregisterCallback](#) ([CallbackHandleType](#) hCallback)

De register change callback Destroys [CNodeCallback](#) object.
- virtual [INodeMap](#) * [GetNodeMap](#) () const

Retrieves the central node map.
- virtual [GenICam::gcstring GetEventID](#) () const

Get the EventId of the node.
- virtual bool [IsStreamable](#) () const

True if the node is streamable.
- virtual void [GetPropertyNames](#) ([GenICam::gcstring_vector](#) &PropertyNames) const

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

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

Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) ([EVisibility](#) ImposedVisibility)

Imposes a visibility to the natural visibility of the node.
- virtual [INode](#) * [GetAlias](#) () const

Retrieves the a node which describes the same feature in a different way.
- virtual [INode](#) * [GetCastAlias](#) () const

Retrieves the a node which describes the same feature so that it can be casted.
- virtual [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 [\ingroup Spinnaker_GenApi_PublicImpl](#).
- virtual void [SetReference](#) ([ISelector](#) *pBase)
- void [SetNodeMap](#) ([INodeMap](#) *pNodeMap)
- virtual bool [operator==](#) (int nullPtr) const
- virtual bool [operator!=](#) (int nullPtr) const

Protected Attributes

- `std::shared_ptr< Node::NodeImpl > m_pNodeData`
- `std::list< CallbackHandleType_t * > m_Callbacks`
List of callbacks.
- `INodeMap * m_pNodeMap`

10.127.1 Detailed Description

class common to all nodes

10.127.2 Constructor & Destructor Documentation

10.127.2.1 Node() [1/2]

`Node ()`

Constructor.

10.127.2.2 Node() [2/2]

`Node (`
`std::shared_ptr< Node::NodeImpl > pNodeHandle)`

Constructor.

10.127.2.3 ~Node()

`~Node ()`

Destructor.

10.127.3 Member Function Documentation

10.127.3.1 DeregisterCallback()

```
virtual bool DeregisterCallback (
    CallbackHandleType hCallback ) [virtual]
```

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

Returns

true if the callback handle was valid

10.127.3.2 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Base interface overrides.

Get the access mode of the node

Reimplemented in [PortRecorder](#).

10.127.3.3 GetAlias()

```
virtual INode* GetAlias ( ) const [virtual]
```

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

10.127.3.4 GetCachingMode()

```
virtual ECachingMode GetCachingMode ( ) const [virtual]
```

Get Caching Mode.

10.127.3.5 GetCastAlias()

```
virtual INode* GetCastAlias ( ) const [virtual]
```

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

10.127.3.6 GetChildren()

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

Get all nodes this node directly depends on.

Parameters

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

10.127.3.7 GetDescription()

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

Get a long description of the node.

10.127.3.8 GetDeviceName()

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

Get a name of the device.

10.127.3.9 GetDisplayName()

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

Get a name string for display.

10.127.3.10 GetDocuURL()

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

Gets a URL pointing to the documentation of that feature.

10.127.3.11 GetEventID()

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

Get the EventId of the node.

10.127.3.12 GetName()

```
virtual GenICam::gcstring GetName (
    bool FullQualified = false ) const [virtual]
```

Get node name.

10.127.3.13 GetNameSpace()

```
virtual GenApi::ENamespace GetNameSpace ( ) const [virtual]
```

Get name space.

10.127.3.14 GetNodeHandle()

```
std::shared_ptr<Node::NodeImpl> GetNodeHandle ( ) const
```

Get [Node](#) handle.

10.127.3.15 GetNodeMap()

```
virtual INodeMap* GetNodeMap ( ) const [virtual]
```

Retrieves the central node map.

10.127.3.16 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 |
|-----|----------------|----------------------|

10.127.3.17 GetPollingTime()

```
virtual int64_t GetPollingTime ( ) const [virtual]
```

recommended polling time (for not cacheable nodes)

10.127.3.18 GetPrincipalInterfaceType()

```
virtual EInterfaceType GetPrincipalInterfaceType ( ) const [virtual]
```

Get the type of the main interface of a node.

10.127.3.19 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.

10.127.3.20 GetPropertyNames()

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

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

10.127.3.21 GetSelectedFeatures()

```
virtual void GetSelectedFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of selected features

10.127.3.22 GetSelectingFeatures()

```
virtual void GetSelectingFeatures (
    FeatureList_t & ) const [virtual]
```

retrieve the group of features selecting this node

10.127.3.23 GetToolTip()

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

Get a short description of the node.

10.127.3.24 GetVisibility()

```
virtual EVisibility GetVisibility ( ) const [virtual]
```

Get the recommended visibility of the node.

10.127.3.25 ImposeAccessMode()

```
virtual void ImposeAccessMode (
    EAccessMode ImposedAccessMode ) [virtual]
```

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

10.127.3.26 ImposeVisibility()

```
virtual void ImposeVisibility (
    EVisibility ImposedVisibility ) [virtual]
```

Imposes a visibility to the natural visibility of the node.

10.127.3.27 InvalidateNode()

```
virtual void InvalidateNode ( ) [virtual]
```

Indicates that the node's value may have changed.

Fires the callback on this and all dependent nodes

10.127.3.28 IsAccessModeCacheable()

```
virtual EYesNo IsAccessModeCacheable ( ) const [virtual]
```

True if the AccessMode can be cached.

10.127.3.29 IsCachable()

```
virtual bool IsCachable ( ) const [virtual]
```

Is the node value cacheable.

10.127.3.30 IsDeprecated()

```
virtual bool IsDeprecated ( ) const [virtual]
```

True if the node should not be used any more.

10.127.3.31 IsFeature()

```
virtual bool IsFeature ( ) const [virtual]
```

True if the node can be reached via category nodes from a category node named "Root".

10.127.3.32 IsSelector()

```
virtual bool IsSelector ( ) const [virtual]
```

Selector interface overrides.

true if this feature selects a group of features

10.127.3.33 IsStreamable()

```
virtual bool IsStreamable ( ) const [virtual]
```

True if the node is streamable.

10.127.3.34 operator!=()

```
virtual bool operator!= (
    int nullPtr ) const [virtual]
```

10.127.3.35 operator==()

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

10.127.3.36 RegisterCallback()

```
virtual CallbackHandleType RegisterCallback (
    CNodeCallback * pCallback ) [virtual]
```

Register change callback Takes ownership of the [CNodeCallback](#) object.

10.127.3.37 SetNodeHandle()

```
void SetNodeHandle (
    std::shared_ptr< Node::NodeImpl > pNodeHandle )
```

Set [Node](#) handle.

10.127.3.38 SetNodeMap()

```
void SetNodeMap (
    INodeMap * pNodeMap )
```

10.127.3.39 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](#).

10.127.3.40 SetReference() [2/2]

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

10.127.4 Member Data Documentation

10.127.4.1 m_Callbacks

```
std::list<CallbackHandleType_t*> m_Callbacks [protected]
```

List of callbacks.

10.127.4.2 m_pNodeData

```
std::shared_ptr<Node::NodeImpl> m_pNodeData [protected]
```

10.127.4.3 m_pNodeMap

```
INodeMap* m_pNodeMap [protected]
```

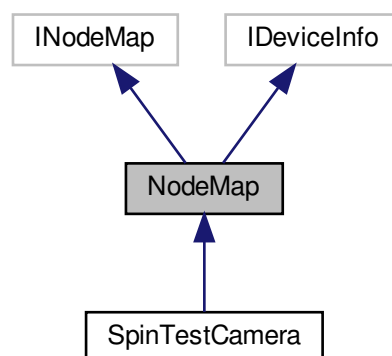
The documentation for this class was generated from the following file:

- include/SpinGenApi/Node.h

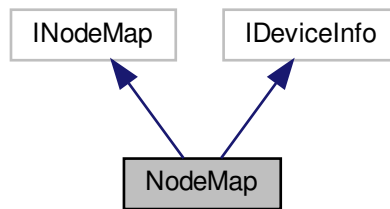
10.128 NodeMap Class Reference

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](#).

10.128.1 Detailed Description

Smart pointer template for NodeMaps with create function.

Parameters

| | |
|----------------------|---|
| <i>TCameraParams</i> | The camera specific parameter class (auto generated from camera xml file) |
|----------------------|---|

10.128.2 Constructor & Destructor Documentation

10.128.2.1 NodeMap()

```
NodeMap (
    GenICam::gcstring DeviceName = "Device" )
```

Constructor.

10.128.2.2 ~NodeMap()

```
virtual ~NodeMap ( ) [virtual]
```

Destructor.

10.128.3 Member Function Documentation

10.128.3.1 ClearXMLCache()

```
static bool ClearXMLCache ( ) [static]
```

Clears the cache of the camera description files.

10.128.3.2 Connect() [1/2]

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

Connects a port to a port node with given name.

10.128.3.3 Connect() [2/2]

```
virtual bool Connect (
    IPort * pPort ) const [virtual]
```

Connects a port to the standard port "Device".

10.128.3.4 Destroy()

```
void Destroy ( )
```

Destroys the node map.

10.128.3.5 GetDeviceName()

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

Get device name.

10.128.3.6 GetDeviceVersion()

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

Get the version of the device description file.

10.128.3.7 GetGenApiVersion()

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

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

10.128.3.8 GetLock()

```
virtual CLock& GetLock ( ) const [virtual]
```

Returns the lock which guards the node map.

10.128.3.9 GetModelName()

```
virtual GenICam::gcstring GetModelName ( ) [virtual]
```

Get the model name.

10.128.3.10 GetNode()

```
virtual INode* GetNode (
    const GenICam::gcstring & key ) const [virtual]
```

Retrieves the node from the central map by name.

10.128.3.11 GetNodeMapHandle()

```
void* GetNodeMapHandle ( ) const
```

10.128.3.12 GetNodes()

```
virtual void GetNodes (
    NodeList_t & Nodes ) const [virtual]
```

Retrieves all nodes in the node map.

10.128.3.13 GetNumNodes()

```
virtual uint64_t GetNumNodes ( ) const [virtual]
```

Get the number of nodes in the map.

10.128.3.14 GetProductGuid()

```
virtual GenICam::gcstring GetProductGuid ( ) [virtual]
```

Get the GUID describing the product.

10.128.3.15 GetSchemaVersion()

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

Get the schema version number.

10.128.3.16 GetStandardNameSpace()

```
virtual GenICam::gcstring GetStandardNameSpace ( ) [virtual]
```

Get the standard name space.

10.128.3.17 GetSupportedSchemaVersions()

```
virtual void GetSupportedSchemaVersions (
    GenICam::gcstring\_vector & SchemaVersions ) [virtual]
```

Gets a list of supported schema versions.

! Loads an XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromFile(const [GenICam::gcstring](#)& XMLFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Loads a Zipped XML, checks it for correctness, applies a style-sheet and outputs it void PreprocessXMLFromZIPFile(const [GenICam::gcstring](#)& ZIPFileName, const [GenICam::gcstring](#)& StyleSheetFileName, const [GenICam::gcstring](#)& OutputFileName, const uint32_t XMLValidation = xvDefault);

! Injects an XML file into a target file virtual void MergeXMLFiles(const [GenICam::gcstring](#)& TargetFileName, *< Name of the target XML file to process const [GenICam::gcstring](#)& InjectedFileName, *< Name of the Injected XML file to process const [GenICam::gcstring](#)& OutputFileName *< Name of the output file);

! Extract independent subtree virtual void ExtractIndependentSubtree(const [GenICam::gcstring](#)& XMLData, *< The XML data the subtree is extracted from. const [GenICam::gcstring](#)& InjectXMLData, *< Optional XML data that is injected before extracting the subtree. No effect if an empty string is passed. const [GenICam::gcstring](#)& SubTreeRootNodeName,*< The name of the node that represents the root of the subtree that shall be extracted. [GenICam::gcstring](#)& ExtractedSubtree *< The returned extracted subtree as string.);

Each list entry is a string with the format "{Major}.{Minor}" where {Major} and {Minor} are integers Example: {"1.1", "1.2"} indicates that the schema v1.1 and v1.2 are supported. The SubMinor version number is not given since it is for fully compatible bug fixes only

10.128.3.18 GetToolTip()

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

Get tool tip.

10.128.3.19 GetVendorName()

```
virtual GenICam::gcstring GetVendorName ( ) [virtual]
```

Get the vendor name.

10.128.3.20 GetVersionGuid()

```
virtual GenICam::gcstring GetVersionGuid ( ) [virtual]
```

Get the GUID describing the product version.

10.128.3.21 InvalidateNodes()

```
virtual void InvalidateNodes ( ) const [virtual]
```

Invalidates all nodes.

10.128.3.22 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);

10.128.3.23 LoadXMLFromFileInject()

```
void LoadXMLFromFileInject (
    GenICam::gcstring TargetFileName,
    GenICam::gcstring InjectFileName )
```

Creates the object from a XML target and an inject file with given file name.

10.128.3.24 LoadXMLFromString()

```
void LoadXMLFromString (
    const GenICam::gcstring & XMLData )
```

Creates the object from XML data given in a string.

10.128.3.25 LoadXMLFromStringInject()

```
void LoadXMLFromStringInject (
    const GenICam::gcstring & TargetXMLDataconst,
    const GenICam::gcstring & InjectXMLData )
```

Creates the object from XML data given in a string with injection.

10.128.3.26 LoadXMLFromZIPData()

```
void LoadXMLFromZIPData (
    const void * zipData,
    size_t zipSize )
```

Creates the object from a ZIP'd XML file given in a string.

10.128.3.27 LoadXMLFromZIPFile()

```
void LoadXMLFromZIPFile (
    GenICam::gcstring ZipFileName )
```

Creates the object from a ZIP'd XML file with given file name.

10.128.3.28 Poll()

```
virtual void Poll (
    int64_t ElapsedTime ) [virtual]
```

Fires nodes which have a polling time.

10.128.4 Member Data Documentation

10.128.4.1 _Ptr

[INodeMap](#)* _Ptr

Pointer to the [NodeMap](#).

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

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

10.129 CNodeMapFactory::NodeStatistics_t Struct Reference

Public Attributes

- uint32_t [NumNodes](#)
- uint32_t [NumProperties](#)
- uint32_t [NumLinks](#)
- uint32_t [NumStrings](#)

10.129.1 Member Data Documentation

10.129.1.1 NumLinks

uint32_t NumLinks

10.129.1.2 NumNodes

uint32_t NumNodes

10.129.1.3 NumProperties

uint32_t NumProperties

10.129.1.4 NumStrings

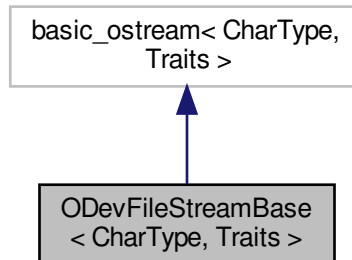
uint32_t NumStrings

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

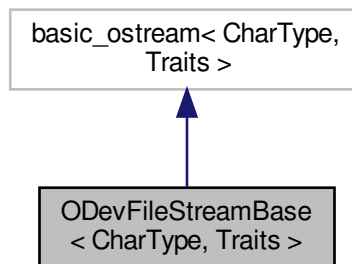
- include/SpinGenApi/[NodeMapFactory.h](#)

10.130 ODevFileStreamBase< CharType, Traits > Class Template Reference

Inheritance diagram for ODevFileStreamBase< CharType, Traits >:



Collaboration diagram for ODevFileStreamBase< CharType, Traits >:



Public Types

- typedef [ODevFileStreamBuf< CharType, Traits >](#) [filebuf_type](#)
- typedef std::basic_ios< CharType, Traits > [ios_type](#)
- typedef std::basic_ostream< CharType, Traits > [ostream_type](#)

Public Member Functions

- [filebuf_type](#) * [rdbuf](#) () const
- bool [is_open](#) () const
- void [open](#) (INodeMap *pInterface, const char *pFileName, std::ios_base::openmode mode=std::ios_base::out|std::ios_base::trunc)
Open file on device in write mode.
- void [close](#) ()
Close the file on device.

10.130.1 Member Typedef Documentation

10.130.1.1 filebuf_type

```
typedef ODevFileStreamBuf<CharType, Traits> filebuf_type
```

10.130.1.2 ios_type

```
typedef std::basic_ios<CharType, Traits> ios_type
```

10.130.1.3 ostream_type

```
typedef std::basic_ostream<CharType, Traits> ostream_type
```

10.130.2 Member Function Documentation

10.130.2.1 close()

```
void close ( ) [inline]
```

Close the file on device.

10.130.2.2 is_open()

```
bool is_open ( ) const [inline]
```

10.130.2.3 open()

```
void open (
    INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode = std::ios_base::out | std::ios_base::trunc ) [inline]
```

Open file on device in write mode.

Parameters

| | |
|-------------------|--|
| <i>pInterface</i> | NodeMap of the device to which the FileProtocolAdapter is attached |
| <i>pFileName</i> | Name of the file to open |
| <i>mode</i> | open mode |

10.130.2.4 rdbuf()

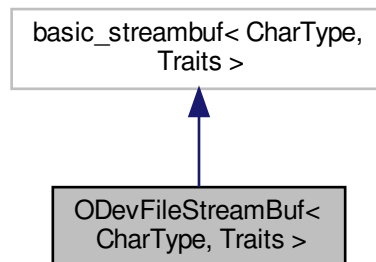
```
filebuf_type* rdbuf ( ) const [inline]
```

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

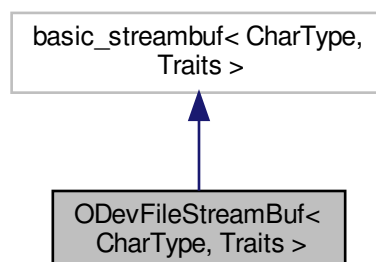
- include/SpinGenApi/[Filestream.h](#)

10.131 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 [xspu](#)tn (const char_type *s, std::streamsize n)
- int_type [overflow](#) (int_type c=traits_type::eof())
- int [sync](#) ()

10.131.1 Constructor & Destructor Documentation

10.131.1.1 ODevFileStreamBuf()

```
ODevFileStreamBuf ( ) [inline]
```

10.131.1.2 ~ODevFileStreamBuf()

```
~ODevFileStreamBuf ( ) [inline]
```

10.131.2 Member Function Documentation

10.131.2.1 close()

```
filebuf_type* close ( ) [inline]
```

10.131.2.2 is_open()

```
bool is_open ( ) const [inline]
```

10.131.2.3 open()

```
filebuf_type* open (
    Spinnaker::GenApi::INodeMap * pInterface,
    const char * pFileName,
    std::ios_base::openmode mode ) [inline]
```

10.131.2.4 overflow()

```
int_type overflow (
    int_type c = traits_type::eof() ) [inline], [protected]
```

10.131.2.5 sync()

```
int sync ( ) [inline], [protected]
```

10.131.2.6 xspn()

```
std::streamsize xspn (
    const char_type * s,
    std::streamsize n ) [inline], [protected]
```

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

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

10.132 PGMOption Struct Reference

Options for saving PGM images.

Public Member Functions

- [PGMOption](#) ()

Public Attributes

- bool [binaryFile](#)
Whether to save the PPM as a binary file.
- unsigned int [reserved](#) [16]
Reserved for future use.

10.132.1 Detailed Description

Options for saving PGM images.

10.132.2 Constructor & Destructor Documentation

10.132.2.1 PGMOption()

```
PGMOption ( ) [inline]
```

10.132.3 Member Data Documentation

10.132.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

10.132.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

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

- [include/SpinnakerDefs.h](#)

10.133 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.

10.133.1 Detailed Description

Options for saving PNG images.

10.133.2 Constructor & Destructor Documentation

10.133.2.1 PNGOption()

```
PNGOption ( ) [inline]
```

10.133.3 Member Data Documentation

10.133.3.1 [compressionLevel](#)

```
unsigned int compressionLevel
```

Compression level (0-9).

0 is no compression, 9 is best compression.

10.133.3.2 [interlaced](#)

```
bool interlaced
```

Whether to save the PNG as interlaced.

10.133.3.3 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

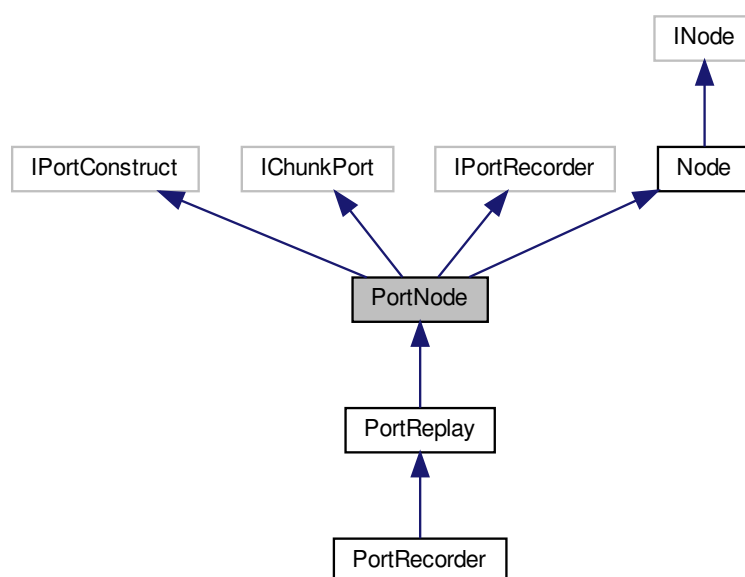
The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

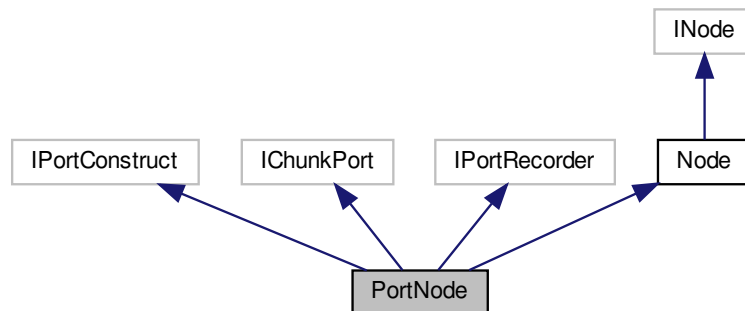
10.134 PortNode Class Reference

[Interface](#) for value properties.

Inheritance diagram for PortNode:



Collaboration diagram for PortNode:



Public Member Functions

- [PortNode](#) ()
Constructor.
- [PortNode](#) (std::shared_ptr< Node::NodeImpl > pValue)
constructor with [GenICam](#) IValue
- [~PortNode](#) ()
Destructor.
- virtual void [Read](#) (void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Reads a chunk of bytes from the port.
- virtual void [Write](#) (const void *pBuffer, int64_t [Address](#), int64_t [Length](#))
Writes a chunk of bytes to the port.
- void [SetPortImpl](#) (IPort *pPort)
Sets pointer the real port implementation; this function may called only once.
- virtual [EYesNo](#) [GetSwapEndianness](#) ()
Determines if the port adapter must perform an endianness swap.
- virtual [Spinnaker::GenICam::gcstring](#) [GetChunkID](#) () const
Get the Id of the chunk the port should be attached to.
- virtual [EYesNo](#) [CacheChunkData](#) () const
Indicates if the chunk a adapter must hold a cached version of the chunk data.
- virtual void [StartRecording](#) (IPortWriteList *pPortRecorder)
Starts logging all WriteRegister commands to a list.
- virtual void [StopRecording](#) ()
Stops recording.
- virtual void [Replay](#) (IPortWriteList *pPortRecorder, bool [Invalidate](#)=true)
Sends the commands to the camera.
- virtual void [SetReference](#) (INode *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IPort *pBase)
overload SetReference for Value
- virtual void [SetReference](#) (IChunkPort *pBase)
overload SetReference for Value
- std::shared_ptr< Node::NodeImpl > [GetPortHandle](#) ()

Additional Inherited Members

10.134.1 Detailed Description

[Interface](#) for value properties.

10.134.2 Constructor & Destructor Documentation

10.134.2.1 PortNode() [1/2]

```
PortNode ( )
```

Constructor.

10.134.2.2 PortNode() [2/2]

```
PortNode (
    std::shared_ptr< Node::NodeImpl > pValue )
```

constructor with [GenICam](#) IValue

10.134.2.3 ~PortNode()

```
~PortNode ( )
```

Destructor.

10.134.3 Member Function Documentation

10.134.3.1 CacheChunkData()

```
virtual EYesNo CacheChunkData ( ) const [virtual]
```

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

10.134.3.2 GetChunkID()

```
virtual Spinnaker::GenICam::gcstring GetChunkID ( ) const [virtual]
```

Get the Id of the chunk the port should be attached to.

10.134.3.3 GetPortHandle()

```
std::shared_ptr<Node::NodeImpl> GetPortHandle ( ) [inline]
```

10.134.3.4 GetSwapEndianness()

```
virtual EYesNo GetSwapEndianness ( ) [virtual]
```

Determines if the port adapter must perform an endianness swap.

10.134.3.5 Read()

```
virtual void Read (  
    void * pBuffer,  
    int64_t Address,  
    int64_t Length ) [virtual]
```

Reads a chunk of bytes from the port.

10.134.3.6 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 [PortReplay](#).

10.134.3.7 SetPortImpl()

```
void SetPortImpl (
    IPort * pPort )
```

Sets pointer the real port implementation; this function may called only once.

10.134.3.8 SetReference() [1/3]

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

overload SetReference for Value

Reimplemented from [Node](#).

10.134.3.9 SetReference() [2/3]

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

overload SetReference for Value

Reimplemented in [PortRecorder](#), and [PortReplay](#).

10.134.3.10 SetReference() [3/3]

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

overload SetReference for Value

10.134.3.11 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

Starts logging all WriteRegister commands to a list.

Reimplemented in [PortRecorder](#).

10.134.3.12 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

Stops recording.

Reimplemented in [PortRecorder](#).

10.134.3.13 Write()

```
virtual void Write (
    const void * pBuffer,
    int64_t Address,
    int64_t Length ) [virtual]
```

Writes a chunk of bytes to the port.

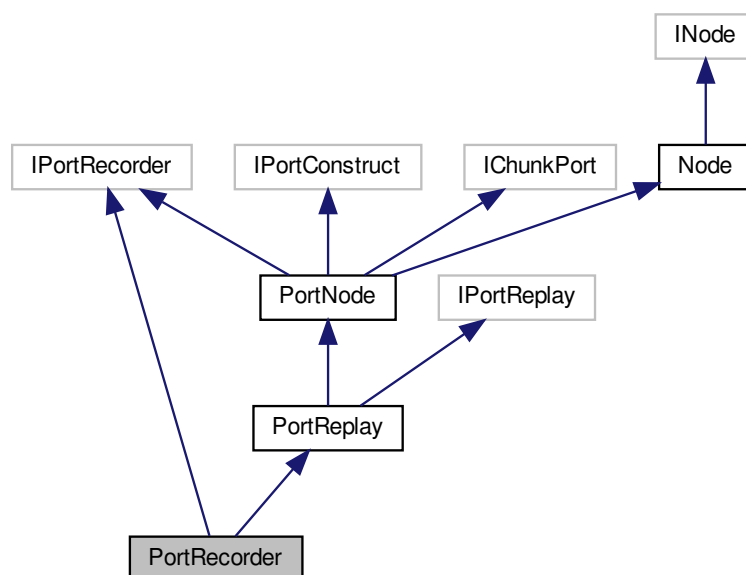
The documentation for this class was generated from the following file:

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

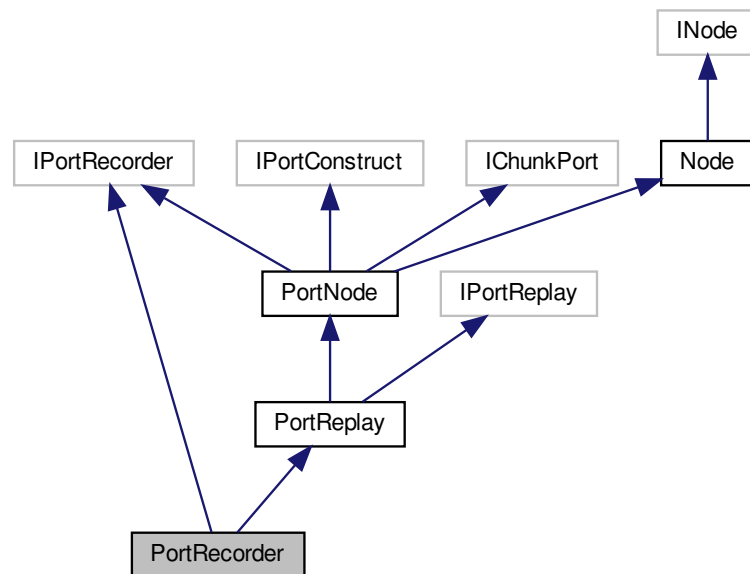
10.135 PortRecorder Class Reference

[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

Additional Inherited Members

10.135.1 Detailed Description

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

10.135.2 Constructor & Destructor Documentation

10.135.2.1 PortRecorder()

```
PortRecorder ( )
```

10.135.2.2 ~PortRecorder()

```
virtual ~PortRecorder ( ) [virtual]
```

10.135.3 Member Function Documentation

10.135.3.1 GetAccessMode()

```
virtual EAccessMode GetAccessMode ( ) const [virtual]
```

Get the access mode of the node.

Reimplemented from [Node](#).

10.135.3.2 SetReference()

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

overload SetReference for Value

Reimplemented from [PortReplay](#).

10.135.3.3 StartRecording()

```
virtual void StartRecording (
    IPortWriteList * pPortRecorder ) [virtual]
```

starts logging all WriteRegister commands to a list

Reimplemented from [PortNode](#).

10.135.3.4 StopRecording()

```
virtual void StopRecording ( ) [virtual]
```

stops recording

Reimplemented from [PortNode](#).

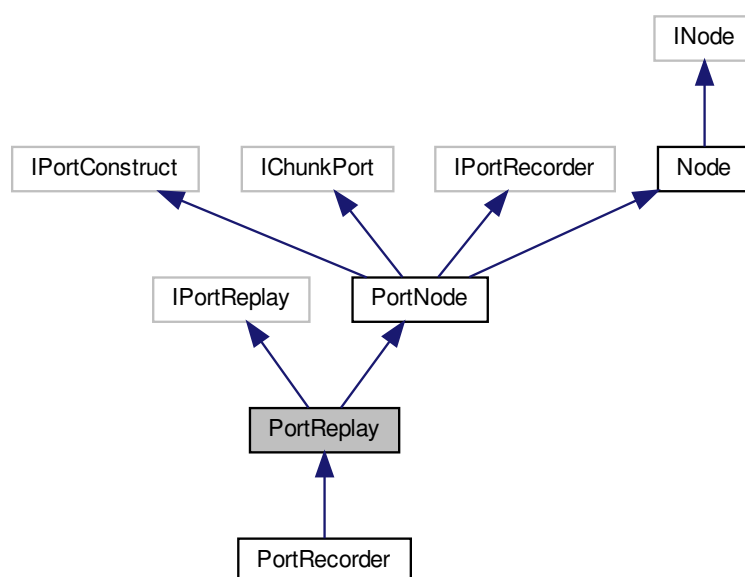
The documentation for this class was generated from the following file:

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

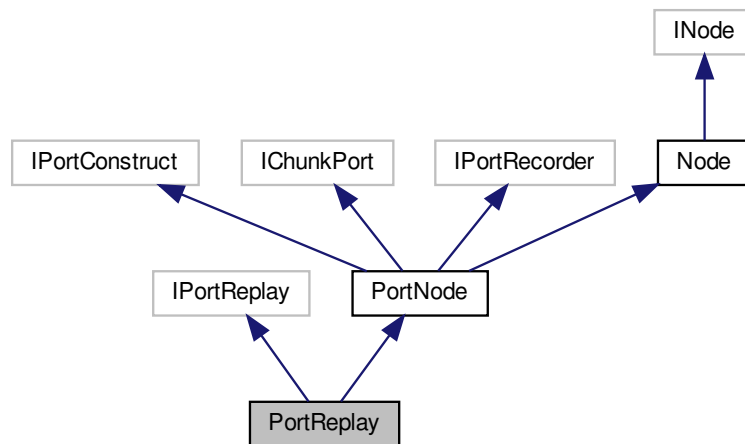
10.136 PortReplay Class Reference

[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](#) ()

Additional Inherited Members

10.136.1 Detailed Description

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

10.136.2 Constructor & Destructor Documentation

10.136.2.1 PortReplay()

[PortReplay](#) ()

10.136.2.2 ~PortReplay()

```
virtual ~PortReplay ( ) [virtual]
```

10.136.3 Member Function Documentation

10.136.3.1 GetPortReplayHandle()

```
void* GetPortReplayHandle ( )
```

10.136.3.2 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](#).

10.136.3.3 SetReference()

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

overload SetReference for Value

Reimplemented from [PortNode](#).

Reimplemented in [PortRecorder](#).

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

- include/SpinGenApi/[PortReplay.h](#)

10.137 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.

10.137.1 Detailed Description

Options for saving PPM images.

10.137.2 Constructor & Destructor Documentation

10.137.2.1 PPMOption()

```
PPMOption ( ) [inline]
```

10.137.3 Member Data Documentation

10.137.3.1 binaryFile

```
bool binaryFile
```

Whether to save the PPM as a binary file.

10.137.3.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

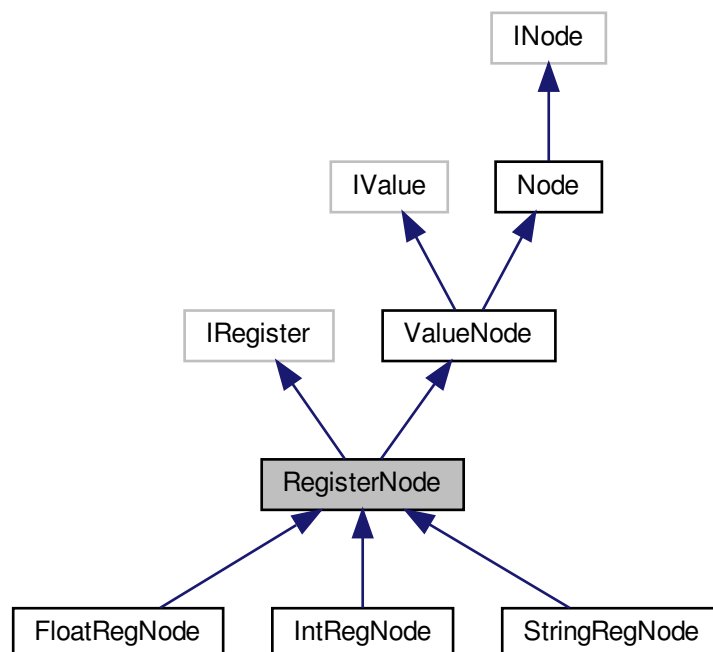
The documentation for this struct was generated from the following file:

- include/[SpinnakerDefs.h](#)

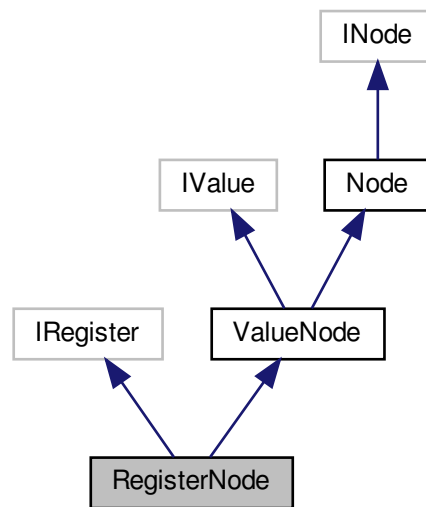
10.138 RegisterNode Class Reference

[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

10.138.1 Detailed Description

[Interface](#) for string properties.

10.138.2 Constructor & Destructor Documentation

10.138.2.1 RegisterNode() [1/2]

```
RegisterNode ( )
```

10.138.2.2 RegisterNode() [2/2]

```
RegisterNode (
    std::shared_ptr< Node::NodeImpl > pRegister )
```

10.138.2.3 ~RegisterNode()

```
virtual ~RegisterNode ( ) [virtual]
```

10.138.3 Member Function Documentation**10.138.3.1 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

10.138.3.2 GetAddress()

```
virtual int64_t GetAddress ( ) [virtual]
```

Retrieves the Address of the register.

10.138.3.3 GetLength()

```
virtual int64_t GetLength ( ) [virtual]
```

Retrieves the Length of the register [Bytes].

10.138.3.4 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) |

10.138.3.5 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Register

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#), [FloatRegNode](#), and [IntRegNode](#).

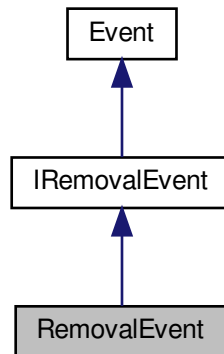
The documentation for this class was generated from the following file:

- include/SpinGenApi/[RegisterNode.h](#)

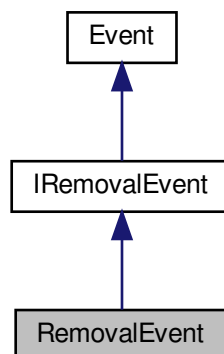
10.139 RemovalEvent Class Reference

An event handler for capturing the device removal event.

Inheritance diagram for RemovalEvent:



Collaboration diagram for RemovalEvent:



Public Member Functions

- [RemovalEvent](#) ()
Default Constructor.
- virtual [~RemovalEvent](#) ()
Virtual Destructor.
- virtual void [OnDeviceRemoval](#) (uint64_t serialNumber)=0
Device removal event callback.

Protected Member Functions

- [RemovalEvent](#) & `operator=` (const [RemovalEvent](#) &)
Assignment operator.

Additional Inherited Members

10.139.1 Detailed Description

An event handler for capturing the device removal event.

10.139.2 Constructor & Destructor Documentation

10.139.2.1 RemovalEvent()

```
RemovalEvent ( )
```

Default Constructor.

10.139.2.2 ~RemovalEvent()

```
virtual ~RemovalEvent ( ) [virtual]
```

Virtual Destructor.

10.139.3 Member Function Documentation

10.139.3.1 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 [IRemovalEvent](#).

10.139.3.2 operator=()

```
RemovalEvent& operator= (
    const RemovalEvent & ) [protected]
```

Assignment operator.

The documentation for this class was generated from the following file:

- include/[RemovalEvent.h](#)

10.140 SingleChunkData_t Struct Reference

Public Attributes

- uint64_t [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

10.140.1 Member Data Documentation

10.140.1.1 ChunkID

uint64_t ChunkID

10.140.1.2 ChunkLength

size_t ChunkLength

10.140.1.3 ChunkOffset

ptrdiff_t ChunkOffset

The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

10.141 SingleChunkDataStr_t Struct Reference

Public Attributes

- GenICam::gcstring [ChunkID](#)
- ptrdiff_t [ChunkOffset](#)
- size_t [ChunkLength](#)

10.141.1 Member Data Documentation

10.141.1.1 ChunkID

GenICam::gcstring ChunkID

10.141.1.2 ChunkLength

size_t ChunkLength

10.141.1.3 ChunkOffset

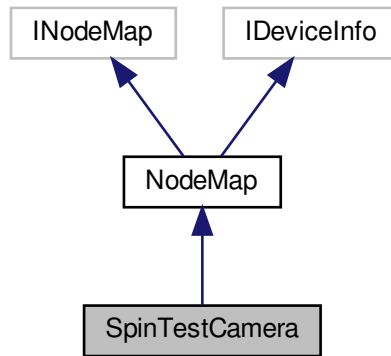
ptrdiff_t ChunkOffset

The documentation for this struct was generated from the following file:

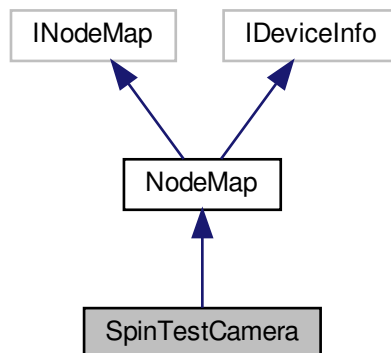
- include/SpinGenApi/[ChunkAdapterGeneric.h](#)

10.142 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](#)

10.143 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.

10.143.1 Detailed Description

Provides the functionality for the user to record images to an AVI/MP4 file.

10.143.2 Constructor & Destructor Documentation

10.143.2.1 SpinVideo()

```
SpinVideo ( )
```

Default constructor.

10.143.2.2 ~SpinVideo()

```
virtual ~SpinVideo ( ) [virtual]
```

Default destructor.

10.143.3 Member Function Documentation

10.143.3.1 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. |
|---------------|----------------------|

10.143.3.2 Close()

```
virtual void Close ( ) [virtual]
```

Close the video file.

This function will throw an exception when the H264 encoder was unable to output any encoded frames, in which case the output video should be considered invalid.

See also

[Open\(\)](#)
[Append\(ImagePtr pImage\)](#)

10.143.3.3 Open() [1/3]

```
virtual void Open (
    const char * pFileName,
    AVIOption & pOption ) [virtual]
```

Open an video file in preparation for writing Images to disk.

The size of video files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters

| | |
|------------------|-------------------------------------|
| <i>pFileName</i> | The filename of the video file. |
| <i>pOption</i> | Options to apply to the video file. |

See also

[Close\(\)](#)

10.143.3.4 Open() [2/3]

```
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](#)

10.143.3.5 **Open()** [3/3]

```
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](#)

10.143.3.6 **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\(ImagePtr plImage\)](#)

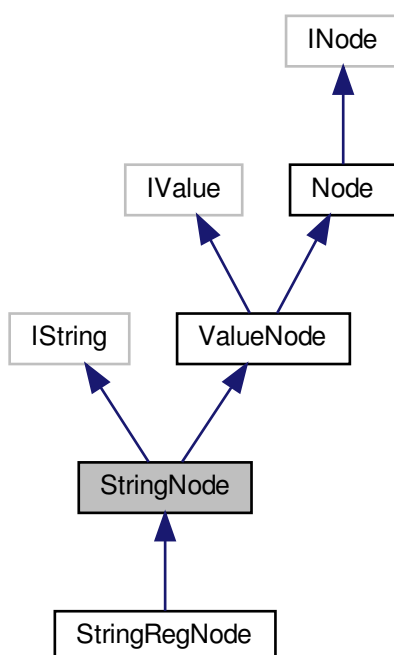
The documentation for this class was generated from the following file:

- [include/SpinVideo.h](#)

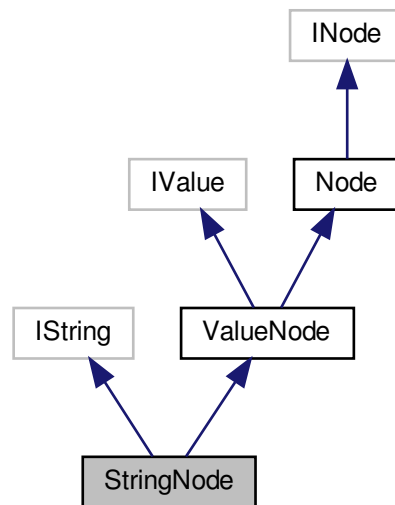
10.144 StringNode Class Reference

[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

10.144.1 Detailed Description

[Interface](#) for string properties.

10.144.2 Constructor & Destructor Documentation

10.144.2.1 StringNode() [1/2]

```
StringNode ( )
```

10.144.2.2 StringNode() [2/2]

```
StringNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

10.144.2.3 ~StringNode()

```
virtual ~StringNode ( ) [virtual]
```

10.144.3 Member Function Documentation

10.144.3.1 GetMaxLength()

```
virtual int64_t GetMaxLength ( ) [virtual]
```

Retrieves the maximum length of the string in bytes.

10.144.3.2 GetValue()

```
virtual GenICam::gcstring GetValue (
    bool Verify = false,
    bool IgnoreCache = false ) [virtual]
```

Get node value.

Parameters

| | |
|--------------------|--|
| <i>Verify</i> | Enables Range verification (default = false). The AccessMode is always checked |
| <i>IgnoreCache</i> | If true the value is read ignoring any caches (default = false) |

Returns

The value read

10.144.3.3 operator()

```
virtual GenICam::gcstring operator() ( ) [virtual]
```

Get node value.

10.144.3.4 operator*()

```
virtual GenICam::gcstring operator* ( ) [virtual]
```

Get node value.

10.144.3.5 operator=()

```
virtual IString& operator= (
    const GenICam::gcstring & Value ) [virtual]
```

Set node value.

10.144.3.6 SetReference()

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [ValueNode](#).

Reimplemented in [StringRegNode](#).

10.144.3.7 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) |

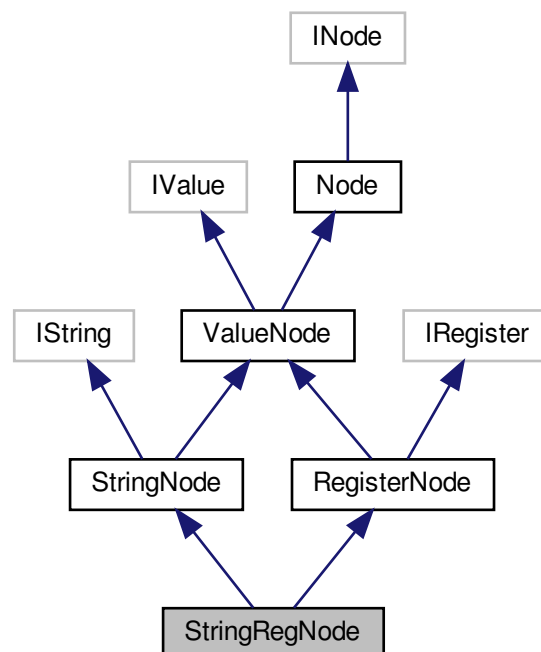
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringNode.h](#)

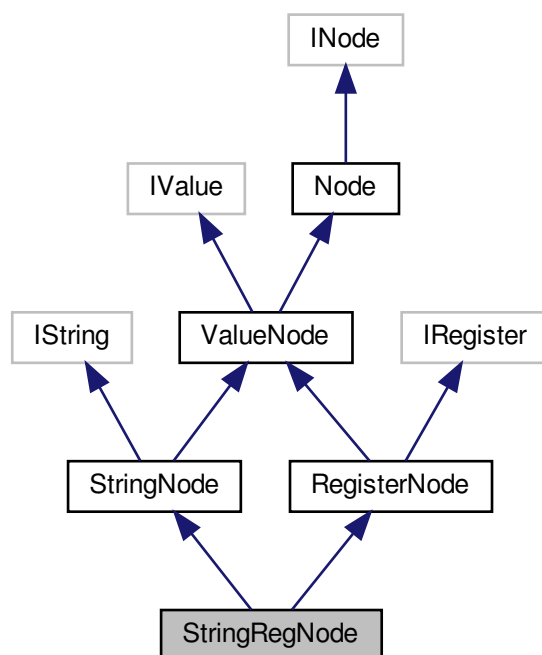
10.145 StringRegNode Class Reference

[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

10.145.1 Detailed Description

[Interface](#) for string properties.

10.145.2 Constructor & Destructor Documentation

10.145.2.1 StringRegNode() [1/2]

```
StringRegNode ( )
```

10.145.2.2 StringRegNode() [2/2]

```
StringRegNode (
    std::shared_ptr< Node::NodeImpl > pString )
```

10.145.2.3 ~StringRegNode()

```
virtual ~StringRegNode ( ) [virtual]
```

10.145.3 Member Function Documentation**10.145.3.1 SetReference()**

```
virtual void SetReference (
    INode * pBase ) [virtual]
```

overload SetReference for Value

Reimplemented from [RegisterNode](#).

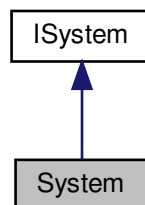
The documentation for this class was generated from the following file:

- include/SpinGenApi/[StringRegNode.h](#)

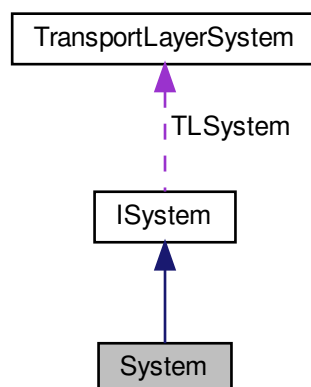
10.146 System Class Reference

The system object is used to retrieve the list of interfaces and cameras available.

Inheritance diagram for System:



Collaboration diagram for System:



Public Member Functions

- virtual `~System()`
Default destructor.
- virtual void `ReleaseInstance()`
This call releases the 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.
- virtual void [RegisterInterfaceEvent](#) ([Event](#) &evtToRegister, bool updateInterface=true)
Registers events for all available interfaces that are found on the system.
- void [UnregisterInterfaceEvent](#) ([Event](#) &evtToUnregister)
Unregisters events for all available interfaces that are found on the system.
- virtual void [RegisterLoggingEvent](#) ([LoggingEvent](#) &handler)
Registers a logging event.
- virtual void [UnregisterAllLoggingEvent](#) ()
Unregisters all previously registered logging events.
- virtual void [UnregisterLoggingEvent](#) ([LoggingEvent](#) &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

10.146.1 Detailed Description

The system object is used to retrieve the list of interfaces and cameras available.

10.146.2 Constructor & Destructor Documentation

10.146.2.1 `~System()`

```
virtual ~System ( ) [virtual]
```

Default destructor.

10.146.2.2 `System()`

```
System ( ) [protected]
```

Default constructor.

10.146.3 Member Function Documentation

10.146.3.1 `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 <code>updateInterfaceList()</code> is called before getting cameras from available interfaces on the system |
| <i>updateCameras</i> | Determines whether or not UpdateCameras() is called before getting cameras from available interfaces on the system |

Returns

An [CameraList](#) object that contains a list of all cameras.

Implements [ISystem](#).

10.146.3.2 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.

10.146.3.3 GetInterfaces()

```
virtual InterfaceList GetInterfaces (
    bool updateInterface = true ) [virtual]
```

Returns a list of interfaces available on the system.

This call returns GigE and Usb2 and Usb3 interfaces. Note that on MacOS only active GigE interfaces will be stored in the returned [InterfaceList](#).

See also

[UpdateInterfaceList\(\)](#)

Parameters

| | |
|------------------------|---|
| <i>updateInterface</i> | Determines whether or not UpdateInterfaceList() is called before getting available interfaces |
|------------------------|---|

Returns

An [InterfaceList](#) object that contains a list of all interfaces.

Implements [ISystem](#).

10.146.3.4 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](#).

10.146.3.5 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](#).

10.146.3.6 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](#).

10.146.3.7 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](#).

10.146.3.8 RegisterInterfaceEvent()

```
virtual void RegisterInterfaceEvent (
    Event & evtToRegister,
    bool updateInterface = true ) [virtual]
```

Registers events for all available interfaces that are found on the system.

Parameters

| | |
|------------------------|---|
| <i>evtToRegister</i> | The event to register for the available interfaces |
| <i>updateInterface</i> | Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system |

Implements [ISystem](#).

10.146.3.9 RegisterLoggingEvent()

```
virtual void RegisterLoggingEvent (
    LoggingEvent & handler ) [virtual]
```

Registers a logging event.

Parameters

| | |
|----------------|---------------------------------------|
| <i>handler</i> | The logging event handler to register |
|----------------|---------------------------------------|

Implements [ISystem](#).

10.146.3.10 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_RESOURCE_IN_USE.

See also

[Error](#)
[GetInstance\(\)](#)

Implements [ISystem](#).

10.146.3.11 SendActionCommand()

```
virtual void SendActionCommand (
    unsigned int deviceKey,
    unsigned int groupKey,
    unsigned int groupMask,
    unsigned long long actionTime = 0,
    unsigned int * pResultSize = 0,
    ActionCommandResult results[] = NULL ) [virtual]
```

Broadcast an Action Command to all devices on system.

Parameters

| | |
|--------------------|--|
| <i>deviceKey</i> | The Action Command's device key |
| <i>groupKey</i> | The Action Command's group key |
| <i>groupMask</i> | The Action Command's group mask |
| <i>actionTime</i> | (Optional) Time when to assert a future action. Zero means immediate action. |
| <i>pResultSize</i> | (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results. If this parameter is 0 or NULL, the function will return as soon as the command has been broadcasted. |
| <i>results</i> | (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL. |

Implements [ISystem](#).

10.146.3.12 SetLoggingEventPriorityLevel()

```
virtual void SetLoggingEventPriorityLevel (
    SpinnakerLogLevel level ) [virtual]
```

Sets a threshold priority level for logging event.

Logging events below such level will not trigger callbacks.

[Spinnaker](#) uses 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](#).

10.146.3.13 UnregisterAllLoggingEvent()

```
virtual void UnregisterAllLoggingEvent ( ) [virtual]
```

Unregisters all previously registered logging events.

Implements [ISystem](#).

10.146.3.14 UnregisterInterfaceEvent()

```
void UnregisterInterfaceEvent (
    Event & evtToUnregister ) [virtual]
```

Unregisters events for all available interfaces that are found on the system.

Parameters

| | |
|------------------------|---|
| <i>evtToUnregister</i> | The event to unregister from the available interfaces |
|------------------------|---|

Implements [ISystem](#).

10.146.3.15 UnregisterLoggingEvent()

```
virtual void UnregisterLoggingEvent (
    LoggingEvent & handler ) [virtual]
```

Unregisters a logging event.

Parameters

| | |
|----------------|---|
| <i>handler</i> | The logging event handler to unregister |
|----------------|---|

Implements [ISystem](#).

10.146.3.16 UpdateCameras()

```
virtual bool UpdateCameras (
    bool updateInterfaces = true ) [virtual]
```

Updates the list of cameras on the system.

Note that [System::GetCameras\(\)](#) internally calls [UpdateCameras\(\)](#) for each interface it enumerates. If the list changed between this call and the last time [UpdateCameras](#) was called then the return value will be true, otherwise it is false.

See also

[GetCameras\(\)](#)

Parameters

| | |
|-------------------------|--|
| <i>updateInterfaces</i> | Determines whether or not UpdateInterfaceList() is called before updating cameras for available interfaces on the system |
|-------------------------|--|

Returns

True if cameras changed on interface and false otherwise.

Implements [ISystem](#).

10.146.3.17 UpdateInterfaceList()

```
virtual void UpdateInterfaceList ( ) [virtual]
```

Updates the list of interfaces on the system.

If desired, local copies of [InterfaceList](#) should be updated by calling [GetInterfaces](#).

See also

[GetInterfaces\(\)](#)

Implements [ISystem](#).

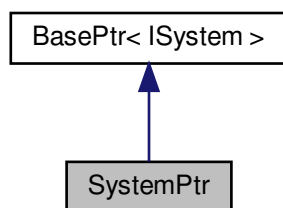
The documentation for this class was generated from the following file:

- [include/System.h](#)

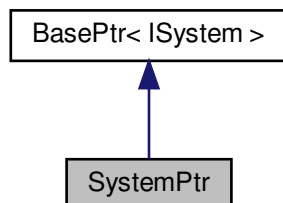
10.147 SystemPtr Class Reference

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

10.147.1 Detailed Description

A reference tracked pointer to a system object.

10.147.2 Constructor & Destructor Documentation

10.147.2.1 [SystemPtr\(\)](#) [1/4]

```
SystemPtr ( )
```

Default constructor.

10.147.2.2 [SystemPtr\(\)](#) [2/4]

```
SystemPtr (  
    const int )
```

Copy constructor.

10.147.2.3 [SystemPtr\(\)](#) [3/4]

```
SystemPtr (  
    const long )
```

Copy constructor.

10.147.2.4 SystemPtr() [4 / 4]

```
SystemPtr (
    const std::nullptr_t )
```

Copy constructor.

10.147.2.5 ~SystemPtr()

```
virtual ~SystemPtr (
    void ) [virtual]
```

Virtual destructor.

The documentation for this class was generated from the following file:

- include/SystemPtr.h

10.148 TIFFOption Struct Reference

Options for saving TIFF images.

Public Types

- enum [CompressionMethod](#) {
 NONE = 1,
 PACKBITS,
 DEFLATE,
 ADOBE_DEFLATE,
 CCITTFAX3,
 CCITTFAX4,
 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.

10.148.1 Detailed Description

Options for saving TIFF images.

10.148.2 Member Enumeration Documentation

10.148.2.1 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. |
| CCITTFAX3 | Save using CCITT Group 3 fax encoding. This is only valid for 1-bit images only. Default to LZW for other bit depths. |
| CCITTFAX4 | 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. |

10.148.3 Constructor & Destructor Documentation

10.148.3.1 TIFFOption()

```
TIFFOption ( ) [inline]
```

10.148.4 Member Data Documentation

10.148.4.1 compression

[CompressionMethod](#) compression

Compression method to use for encoding TIFF images.

10.148.4.2 reserved

```
unsigned int reserved[16]
```

Reserved for future use.

The documentation for this struct was generated from the following file:

- [include/SpinnakerDefs.h](#)

10.149 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 & GevDeviceForceIP](#)

Description: Forces the camera to be on the same subnet as its corresponding interface.

- [GenApi::IBoolean & DeviceMulticastMonitorMode](#)

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

- [GenApi::IEnumerationT < DeviceEndiannessMechanismEnum > & DeviceEndiannessMechanism](#)

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 [CameraInternal](#)

10.149.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

10.149.2 Constructor & Destructor Documentation

10.149.2.1 [TransportLayerDevice](#)() [1/2]

```
TransportLayerDevice (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.149.2.2 [~TransportLayerDevice](#)()

```
~TransportLayerDevice ( )
```

10.149.2.3 [TransportLayerDevice](#)() [2/2]

```
TransportLayerDevice ( ) [protected]
```

10.149.3 Friends And Related Function Documentation

10.149.3.1 [CameraBase](#)

```
friend class CameraBase [friend]
```

10.149.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

10.149.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

10.149.4 Member Data Documentation

10.149.4.1 DeviceAccessStatus

```
GenApi::IEnumerationT<DeviceAccessStatusEnum>& DeviceAccessStatus
```

Description: Gets the access status the transport layer Producer has on the device.

Visibility: Beginner

10.149.4.2 DeviceCurrentSpeed

```
GenApi::IEnumerationT<DeviceCurrentSpeedEnum>& DeviceCurrentSpeed
```

Description: The USB Speed that the device is currently operating at.

Visibility: Expert

10.149.4.3 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

10.149.4.4 DeviceDriverVersion

```
GenApi::IString& DeviceDriverVersion
```

Description: Version of the device driver.

Visibility: Expert

10.149.4.5 DeviceEndiannessMechanism

`GenApi::IEnumerationT<DeviceEndiannessMechanismEnum>& DeviceEndiannessMechanism`

Description: Identifies the endianness handling mode.

Visibility: Expert

10.149.4.6 DeviceID

`GenApi::IString& DeviceID`

Description: Interface-wide unique identifier of this device.

Visibility: Expert

10.149.4.7 DeviceInstanceId

`GenApi::IString& DeviceInstanceId`

Description: Visibility: Invisible.

10.149.4.8 DeviceIsUpdater

`GenApi::IBoolean& DeviceIsUpdater`

Description: Indicates whether the device is in updater mode.

Visibility: Expert

10.149.4.9 DeviceLinkSpeed

`GenApi::IInteger& DeviceLinkSpeed`

Description: Indicates the speed of transmission negotiated by the given network interface in Mbps.

Visibility: Expert

10.149.4.10 DeviceLocation

`GenApi::IString& DeviceLocation`

Description: Device Location.

Visibility: Expert

10.149.4.11 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the remote device model.

Visibility: Beginner

10.149.4.12 DeviceMulticastMonitorMode

`GenApi::IBoolean& DeviceMulticastMonitorMode`

Description: Controls and indicates if the device is operating in as a Multicast Monitor.

Visibility: Expert

10.149.4.13 DeviceSerialNumber

`GenApi::IString& DeviceSerialNumber`

Description: Serial number of the remote device.

Visibility: Expert

10.149.4.14 DeviceType

`GenApi::IEnumerationT<DeviceTypeEnum>& DeviceType`

Description: Transport layer type of the device.

Visibility: Expert

10.149.4.15 DeviceU3VProtocol

`GenApi::IBoolean& DeviceU3VProtocol`

Description: Indicates whether the device is communicating in U3V Protocol.

Visibility: Expert

10.149.4.16 DeviceUserID

`GenApi::IString& DeviceUserID`

Description: User Defined Name.

Visibility: Expert

10.149.4.17 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the remote device vendor.

Visibility: Beginner

10.149.4.18 DeviceVersion

`GenApi::IString& DeviceVersion`

Description: Version of the device.

Visibility: Expert

10.149.4.19 GenICamXMLLocation

`GenApi::IEnumerationT<GenICamXMLLocationEnum>& GenICamXMLLocation`

Description: Sets the location to load [GenICam](#) XML.

Visibility: Beginner

10.149.4.20 GenICamXMLPath

`GenApi::IString& GenICamXMLPath`

Description: [GenICam](#) XML Path.

Visibility: Beginner

10.149.4.21 GevCCP

`GenApi::IEnumerationT<GevCCPEnum>& GevCCP`

Description: Controls the device access privilege of an application.

Visibility: Beginner

10.149.4.22 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

10.149.4.23 GevDeviceForceIP

`GenApi::ICommand& GevDeviceForceIP`

Description: Forces the camera to be on the same subnet as its corresponding interface.

Visibility: Expert

10.149.4.24 GevDeviceGateway

`GenApi::IInteger& GevDeviceGateway`

Description: Current gateway IP address of the GVCP interface of the remote device.

Visibility: Expert

10.149.4.25 GevDeviceIPAddress

`GenApi::IInteger& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

10.149.4.26 GevDeviceIsWrongSubnet

`GenApi::IBoolean& GevDeviceIsWrongSubnet`

Description: Indicates whether the device is on the wrong subnet.

Visibility: Expert

10.149.4.27 GevDeviceMACAddress

`GenApi::IInteger& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

10.149.4.28 GevDeviceMaximumPacketSize

`GenApi::IInteger& GevDeviceMaximumPacketSize`

Description: The maximum packet size that can be safely used by the device on the current interface.

Visibility: Expert

10.149.4.29 GevDeviceMaximumRetryCount

`GenApi::Integer &GevDeviceMaximumRetryCount`

Description: Maximum number of times to retry a read/write operation.

Visibility: Expert

10.149.4.30 GevDeviceModelsBigEndian

`GenApi::Boolean &GevDeviceModeIsBigEndian`

Description: This represents the endianness of all device's registers (bootstrap registers and manufacturer-specific registers).

Visibility: Expert

10.149.4.31 GevDevicePort

`GenApi::Integer &GevDevicePort`

Description: Current IP port of the GVCP interface of the selected remote device.

Visibility: Expert

10.149.4.32 GevDeviceReadAndWriteTimeout

`GenApi::Integer &GevDeviceReadAndWriteTimeout`

Description: The timeout in us for read/write operations to the camera.

Visibility: Expert

10.149.4.33 GevDeviceSubnetMask

`GenApi::Integer &GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

10.149.4.34 GevVersionMajor

`GenApi::Integer &GevVersionMajor`

Description: Major version of the specification.

Visibility: Expert

10.149.4.35 GevVersionMinor

`GenApi::Integer& GevVersionMinor`

Description: Minor version of the specification.

Visibility: Expert

10.149.4.36 GUIXMLLocation

`GenApi::EnumerationT<GUIXMLLocationEnum>& GUIXMLLocation`

Description: Sets the location to load GUI XML.

Visibility: Beginner

10.149.4.37 GUIXMLPath

`GenApi::IString& GUIXMLPath`

Description: GUI XML Path.

Visibility: Beginner

The documentation for this class was generated from the following file:

- `include/TransportLayerDevice.h`

10.150 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::IString](#) & [InterfaceType](#)
Description: Transport layer type of the interface.
- [GenApi::Integer](#) & [GevInterfaceGateway](#)
Description: IP address of the selected gateway entry of this interface.
- [GenApi::Integer](#) & [GevInterfaceMACAddress](#)
Description: 48-bit MAC address of this interface.
- [GenApi::Integer](#) & [GevInterfaceIPAddress](#)
Description: IP address of the selected subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceSubnetMask](#)
Description: Subnet mask of the selected subnet of this interface.
- [GenApi::Integer](#) & [GevInterfaceTransmitLinkSpeed](#)
Description: Transmit link speed of this interface in bits per second.
- [GenApi::Integer](#) & [GevInterfaceReceiveLinkSpeed](#)
Description: Receive link speed of this interface in bits per second.
- [GenApi::Integer](#) & [GevInterfaceMTU](#)
Description: Maximum transmission unit of this interface.
- [GenApi::EnumerationT< POEStatusEnum >](#) & [POEStatus](#)
Description: Reports and controls the interface's power over Ethernet status.
- [GenApi::EnumerationT< FilterDriverStatusEnum >](#) & [FilterDriverStatus](#)
Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.
- [GenApi::Integer](#) & [GevActionDeviceKey](#)
Description: Key to authorize the action for the device.
- [GenApi::Integer](#) & [GevActionGroupKey](#)
Description: Provides the key that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer](#) & [GevActionGroupMask](#)
Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.
- [GenApi::Integer](#) & [GevActionTime](#)
Description: Provides the time in nanoseconds when the action is to be executed.
- [GenApi::ICommand](#) & [ActionCommand](#)
Description: Issues an Action Command to attached GEV devices on interface.
- [GenApi::IString](#) & [DeviceUnlock](#)
Description: Unlocks devices for internal use.
- [GenApi::ICommand](#) & [DeviceUpdateList](#)
Description: Updates the internal device list.
- [GenApi::Integer](#) & [DeviceCount](#)
Description: Number of compatible devices detected on current interface.
- [GenApi::Integer](#) & [DeviceSelector](#)
Description: Selector for the different devices on this interface.
- [GenApi::IString](#) & [DeviceID](#)
Description: [Interface](#) wide unique identifier of the selected device.
- [GenApi::IString](#) & [DeviceVendorName](#)
Description: Name of the device vendor.
- [GenApi::IString](#) & [DeviceModelName](#)

Description: Name of the device model.

- [GenApi::IEnumerationT](#) < [DeviceAccessStatusEnum](#) > & [DeviceAccessStatus](#)

Description: Gives the device's access status at the moment of the last execution of "DeviceUpdateList".

- [GenApi::Integer](#) & [GevDeviceIPAddress](#)

Description: Current IP address of the GVCP interface of the selected remote device.

- [GenApi::Integer](#) & [GevDeviceSubnetMask](#)

Description: Current subnet mask of the GVCP interface of the selected remote device.

- [GenApi::Integer](#) & [GevDeviceMACAddress](#)

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

- [GenApi::ICommand](#) & [AutoForceIP](#)

Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.

- [GenApi::Integer](#) & [IncompatibleDeviceCount](#)

Description: Number of incompatible devices detected on current interface.

- [GenApi::Integer](#) & [IncompatibleDeviceSelector](#)

Description: Selector for the devices that are not compatible with [Spinnaker](#) on this interface.

- [GenApi::IString](#) & [IncompatibleDeviceID](#)

Description: [Interface](#) wide unique identifier of the selected incompatible device.

- [GenApi::IString](#) & [IncompatibleDeviceVendorName](#)

Description: Name of the incompatible device vendor.

- [GenApi::IString](#) & [IncompatibleDeviceModelName](#)

Description: Name of the incompatible device model.

- [GenApi::Integer](#) & [IncompatibleGevDeviceIPAddress](#)

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

- [GenApi::Integer](#) & [IncompatibleGevDeviceSubnetMask](#)

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

- [GenApi::Integer](#) & [IncompatibleGevDeviceMACAddress](#)

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

- [GenApi::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](#)

10.150.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

10.150.2 Constructor & Destructor Documentation

10.150.2.1 TransportLayerInterface() [1/2]

```
TransportLayerInterface (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.150.2.2 ~TransportLayerInterface()

```
~TransportLayerInterface ( )
```

10.150.2.3 TransportLayerInterface() [2/2]

```
TransportLayerInterface ( ) [protected]
```

10.150.3 Friends And Related Function Documentation

10.150.3.1 IInterface

```
friend class IInterface [friend]
```

10.150.3.2 Interface

```
friend class Interface [friend]
```

10.150.3.3 InterfaceInternal

```
friend class InterfaceInternal [friend]
```

10.150.4 Member Data Documentation

10.150.4.1 ActionCommand

`GenApi::ICommand& ActionCommand`

Description: Issues an Action Command to attached GEV devices on interface.

Visibility: Expert

10.150.4.2 AutoForceIP

`GenApi::ICommand& AutoForceIP`

Description: Automatically forces any cameras on interface to an IP Address on the same subnet as the interface.

Visibility: Expert

10.150.4.3 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

10.150.4.4 DeviceCount

`GenApi::IInteger& DeviceCount`

Description: Number of compatible devices detected on current interface.

Visibility: Expert

10.150.4.5 DeviceID

`GenApi::IString& DeviceID`

Description: [Interface](#) wide unique identifier of the selected device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.6 DeviceModelName

`GenApi::IString& DeviceModelName`

Description: Name of the device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.7 DeviceSelector

`GenApi::Integer& 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

10.150.4.8 DeviceUnlock

`GenApi::IString& DeviceUnlock`

Description: Unlocks devices for internal use.

Visibility: Expert

10.150.4.9 DeviceUpdateList

`GenApi::ICommand& DeviceUpdateList`

Description: Updates the internal device list.

Visibility: Expert

10.150.4.10 DeviceVendorName

`GenApi::IString& DeviceVendorName`

Description: Name of the device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.11 FilterDriverStatus

`GenApi::IEnumerationT<FilterDriverStatusEnum>& FilterDriverStatus`

Description: Reports whether FLIR Light Weight Filter Driver is enabled or not.

Visibility: Expert

10.150.4.12 GevActionDeviceKey

`GenApi::Integer& GevActionDeviceKey`

Description: Key to authorize the action for the device.

Visibility: Expert

10.150.4.13 GevActionGroupKey

`GenApi::Integer& GevActionGroupKey`

Description: Provides the key that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

10.150.4.14 GevActionGroupMask

`GenApi::Integer& GevActionGroupMask`

Description: Provides the mask that the device will use to validate the action on reception of the action protocol message.

Visibility: Expert

10.150.4.15 GevActionTime

`GenApi::Integer& GevActionTime`

Description: Provides the time in nanoseconds when the action is to be executed.

Visibility: Expert

10.150.4.16 GevDeviceIPAddress

`GenApi::Integer& GevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote device.

Visibility: Expert

10.150.4.17 GevDeviceMACAddress

`GenApi::Integer& GevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote device.

Visibility: Expert

10.150.4.18 GevDeviceSubnetMask

`GenApi::Integer& GevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote device.

Visibility: Expert

10.150.4.19 GevInterfaceGateway

`GenApi::Integer& GevInterfaceGateway`

Description: IP address of the selected gateway entry of this interface.

Visibility: Expert

10.150.4.20 GevInterfaceIPAddress

`GenApi::Integer& GevInterfaceIPAddress`

Description: IP address of the selected subnet of this interface.

Visibility: Expert

10.150.4.21 GevInterfaceMACAddress

`GenApi::Integer& GevInterfaceMACAddress`

Description: 48-bit MAC address of this interface.

Visibility: Expert

10.150.4.22 GevInterfaceMTU

`GenApi::Integer& GevInterfaceMTU`

Description: Maximum transmission unit of this interface.

Visibility: Expert

10.150.4.23 GevInterfaceReceiveLinkSpeed

`GenApi::Integer& GevInterfaceReceiveLinkSpeed`

Description: Receive link speed of this interface in bits per second.

Visibility: Expert

10.150.4.24 GevInterfaceSubnetMask

`GenApi::Integer& GevInterfaceSubnetMask`

Description: Subnet mask of the selected subnet of this interface.

Visibility: Expert

10.150.4.25 `GevInterfaceTransmitLinkSpeed`

`GenApi::Integer& GevInterfaceTransmitLinkSpeed`

Description: Transmit link speed of this interface in bits per second.

Visibility: Expert

10.150.4.26 `HostAdapterDriverVersion`

`GenApi::String& HostAdapterDriverVersion`

Description: Driver version of the interface's host adapter.

Visibility: Expert

10.150.4.27 `HostAdapterName`

`GenApi::String& HostAdapterName`

Description: User readable name of the interface's host adapter.

Visibility: Expert

10.150.4.28 `HostAdapterVendor`

`GenApi::String& HostAdapterVendor`

Description: User readable name of the host adapter's vendor.

Visibility: Expert

10.150.4.29 `IncompatibleDeviceCount`

`GenApi::Integer& IncompatibleDeviceCount`

Description: Number of incompatible devices detected on current interface.

Visibility: Expert

10.150.4.30 `IncompatibleDeviceID`

`GenApi::String& IncompatibleDeviceID`

Description: [Interface](#) wide unique identifier of the selected incompatible device.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.31 IncompatibleDeviceModelName

`GenApi::IString& IncompatibleDeviceModelName`

Description: Name of the incompatible device model.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.32 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

10.150.4.33 IncompatibleDeviceVendorName

`GenApi::IString& IncompatibleDeviceVendorName`

Description: Name of the incompatible device vendor.

This value only changes on execution of "DeviceUpdateList". Visibility: Expert

10.150.4.34 IncompatibleGevDeviceIPAddress

`GenApi::IInteger& IncompatibleGevDeviceIPAddress`

Description: Current IP address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.150.4.35 IncompatibleGevDeviceMACAddress

`GenApi::IInteger& IncompatibleGevDeviceMACAddress`

Description: 48-bit MAC address of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.150.4.36 IncompatibleGevDeviceSubnetMask

`GenApi::IInteger& IncompatibleGevDeviceSubnetMask`

Description: Current subnet mask of the GVCP interface of the selected remote incompatible device.

Visibility: Expert

10.150.4.37 InterfaceDisplayName

`GenApi::IString& InterfaceDisplayName`

Description: User readable name of the selected interface.

Visibility: Expert

10.150.4.38 InterfaceID

`GenApi::IString& InterfaceID`

Description: Transport layer Producer wide unique identifier of the selected interface.

Visibility: Expert

10.150.4.39 InterfaceType

`GenApi::IString& InterfaceType`

Description: Transport layer type of the interface.

Visibility: Expert

10.150.4.40 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](#)

10.151 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::Integer](#) & [StreamTotalBufferCount](#)
Description: Counts the number of image buffers that arrived since stream started.
- [GenApi::Integer](#) & [StreamDefaultBufferCount](#)
Description: DEPRECATED; Replaced by [StreamBufferCountManual](#).
- [GenApi::Integer](#) & [StreamDefaultBufferCountMax](#)
Description: DEPRECATED; Replaced by [StreamBufferCountMax](#).
- [GenApi::IEnumerationT< StreamDefaultBufferCountModeEnum >](#) & [StreamDefaultBufferCountMode](#)
Description: DEPRECATED; Replaced by [StreamBufferCountMode](#).
- [GenApi::Integer](#) & [StreamBufferCountManual](#)
Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.
- [GenApi::Integer](#) & [StreamBufferCountResult](#)
Description: Displays the number of buffers to be used on this stream upon acquisition start.
- [GenApi::Integer](#) & [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::IBoolean](#) & [StreamCRCCheckEnable](#)
Description: Enables or disables CRC checks on received images.
- [GenApi::IBoolean](#) & [GevPacketResendMode](#)
Description: Enables or disables the packet resend mechanism.
- [GenApi::Integer](#) & [GevMaximumNumberResendRequests](#)
Description: Maximum number of resend requests per image.
- [GenApi::Integer](#) & [GevPacketResendTimeout](#)
Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.
- [GenApi::Integer](#) & [GevMaximumNumberResendBuffers](#)
Description: This node is not used and has been deprecated.
- [GenApi::Integer](#) & [GevTotalPacketCount](#)
Description: Displays number of packets received on this stream.
- [GenApi::Integer](#) & [GevFailedPacketCount](#)
Description: Displays number of packets missed on this stream.
- [GenApi::Integer](#) & [GevResendPacketCount](#)
Description: Displays number of packets received after retransmit request on this stream.
- [GenApi::Integer](#) & [StreamFailedBufferCount](#)
Description: Displays number of incomplete images with missing leader/trailer information.
- [GenApi::Integer](#) & [StreamBufferUnderrunCount](#)
Description: Displays number of dropped images caused by driver running out of buffers.
- [GenApi::Integer](#) & [GevResendRequestCount](#)
Description: Displays number of packets requested to be retransmitted on this stream.
- [GenApi::Integer](#) & [StreamBlockTransferSize](#)
Description: Controls the image breakup size that should be used on this stream.

Protected Member Functions

- [TransportLayerStream\(\)](#)

Friends

- class [CameraBase](#)
- class [ICameraBase](#)
- class [CameraInternal](#)

10.151.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

10.151.2 Constructor & Destructor Documentation

10.151.2.1 TransportLayerStream() [1/2]

```
TransportLayerStream (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.151.2.2 ~TransportLayerStream()

```
~TransportLayerStream ( )
```

10.151.2.3 TransportLayerStream() [2/2]

```
TransportLayerStream ( ) [protected]
```

10.151.3 Friends And Related Function Documentation

10.151.3.1 CameraBase

```
friend class CameraBase [friend]
```

10.151.3.2 CameraInternal

```
friend class CameraInternal [friend]
```

10.151.3.3 ICameraBase

```
friend class ICameraBase [friend]
```

10.151.4 Member Data Documentation

10.151.4.1 GevFailedPacketCount

```
GenApi::Integer& GevFailedPacketCount
```

Description: Displays number of packets missed on this stream.

Visibility: Expert

10.151.4.2 GevMaximumNumberResendBuffers

```
GenApi::Integer& GevMaximumNumberResendBuffers
```

Description: This node is not used and has been deprecated.

Visibility: Invisible

10.151.4.3 GevMaximumNumberResendRequests

```
GenApi::Integer& GevMaximumNumberResendRequests
```

Description: Maximum number of resend requests per image.

Each resend request consists of a span of consecutive packet IDs. Visibility: Expert

10.151.4.4 GevPacketResendMode

```
GenApi::Boolean& GevPacketResendMode
```

Description: Enables or disables the packet resend mechanism.

Visibility: Expert

10.151.4.5 `GevPacketResendTimeout`

`GenApi::Integer & GevPacketResendTimeout`

Description: Time in milliseconds to wait after the image trailer is received and before the image is completed by the driver.

Visibility: Expert

10.151.4.6 `GevResendPacketCount`

`GenApi::Integer & GevResendPacketCount`

Description: Displays number of packets received after retransmit request on this stream.

Visibility: Expert

10.151.4.7 `GevResendRequestCount`

`GenApi::Integer & GevResendRequestCount`

Description: Displays number of packets requested to be retransmitted on this stream.

Visibility: Expert

10.151.4.8 `GevTotalPacketCount`

`GenApi::Integer & GevTotalPacketCount`

Description: Displays number of packets received on this stream.

Visibility: Expert

10.151.4.9 `StreamBlockTransferSize`

`GenApi::Integer & StreamBlockTransferSize`

Description: Controls the image breakup size that should be used on this stream.

Visibility: Expert

10.151.4.10 `StreamBufferCountManual`

`GenApi::Integer & StreamBufferCountManual`

Description: Controls the number of buffers to be used on this stream upon acquisition start when in manual mode.

Visibility: Expert

10.151.4.11 StreamBufferCountMax

`GenApi::Integer& StreamBufferCountMax`

Description: Controls the maximum number of buffers that should be used on this stream.

This value is calculated based on the available system memory. Visibility: Expert

10.151.4.12 StreamBufferCountMode

`GenApi::EnumerationT<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

10.151.4.13 StreamBufferCountResult

`GenApi::Integer& StreamBufferCountResult`

Description: Displays the number of buffers to be used on this stream upon acquisition start.

Recalculated on acquisition start if in auto mode. Visibility: Expert

10.151.4.14 StreamBufferHandlingMode

`GenApi::EnumerationT<StreamBufferHandlingModeEnum>& StreamBufferHandlingMode`

Description: Available buffer handling modes of this data stream: Visibility: Beginner.

10.151.4.15 StreamBufferUnderrunCount

`GenApi::Integer& StreamBufferUnderrunCount`

Description: Displays number of dropped images caused by driver running out of buffers.

Visibility: Expert

10.151.4.16 StreamCRCCheckEnable

`GenApi::Boolean& StreamCRCCheckEnable`

Description: Enables or disables CRC checks on received images.

Visibility: Expert

10.151.4.17 StreamDefaultBufferCount

`GenApi::Integer& StreamDefaultBufferCount`

Description: DEPRECATED; Replaced by StreamBufferCountManual.

Controls the number of buffers to be used on this stream upon acquisition start when in manual mode. Visibility: Invisible

10.151.4.18 StreamDefaultBufferCountMax

`GenApi::Integer& StreamDefaultBufferCountMax`

Description: DEPRECATED; Replaced by StreamBufferCountMax.

Controls the maximum number of buffers that should be used on this stream. This value is calculated based on the available system memory. Visibility: Invisible

10.151.4.19 StreamDefaultBufferCountMode

`GenApi::EnumerationT<StreamDefaultBufferCountModeEnum>& StreamDefaultBufferCountMode`

Description: DEPRECATED; Replaced by StreamBufferCountMode.

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: Invisible

10.151.4.20 StreamFailedBufferCount

`GenApi::Integer& StreamFailedBufferCount`

Description: Displays number of incomplete images with missing leader/trailer information.

Visibility: Expert

10.151.4.21 StreamID

`GenApi::IString& StreamID`

Description: Device unique ID for the data stream, e.g.

a GUID. Visibility: Expert

10.151.4.22 StreamTotalBufferCount

`GenApi::Integer& StreamTotalBufferCount`

Description: Counts the number of image buffers that arrived since stream started.

Visibility: Expert

10.151.4.23 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](#)

10.152 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.

Protected Member Functions

- [TransportLayerSystem](#) ()

Friends

- class [System](#)
- class [ISystem](#)
- class [SystemPtrInternal](#)

10.152.1 Detailed Description

Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

10.152.2 Constructor & Destructor Documentation

10.152.2.1 TransportLayerSystem() [1/2]

```
TransportLayerSystem (
    GenApi::INodeMap * nodeMapTLDevice )
```

10.152.2.2 ~TransportLayerSystem()

```
~TransportLayerSystem ( )
```

10.152.2.3 TransportLayerSystem() [2/2]

```
TransportLayerSystem ( ) [protected]
```

10.152.3 Friends And Related Function Documentation

10.152.3.1 ISystem

```
friend class ISystem [friend]
```

10.152.3.2 System

```
friend class System [friend]
```

10.152.3.3 SystemPtrInternal

```
friend class SystemPtrInternal [friend]
```

10.152.4 Member Data Documentation

10.152.4.1 EnumerateGEVInterfaces

`GenApi::IBoolen& EnumerateGEVInterfaces`

Description: Enables or disables enumeration of GEV Interfaces.

Visibility: Expert

The documentation for this class was generated from the following file:

- include/[TransportLayerSystem.h](#)

10.153 U3V_CHUNK_TRAILER Struct Reference

header of a GVCP request packet

Public Attributes

- uint32_t [ChunkID](#)
- uint32_t [ChunkLength](#)

10.153.1 Detailed Description

header of a GVCP request packet

10.153.2 Member Data Documentation

10.153.2.1 ChunkID

uint32_t ChunkID

10.153.2.2 ChunkLength

uint32_t ChunkLength

The documentation for this struct was generated from the following file:

- include/SpinGenApi/[ChunkAdapterU3V.h](#)

10.154 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](#)

10.154.1 Detailed Description

U3V/GenCP command header.

10.154.2 Member Data Documentation

10.154.2.1 CommandId

uint16_t CommandId

10.154.2.2 Flags

uint16_t Flags

10.154.2.3 Length

uint16_t Length

10.154.2.4 Prefix

uint32_t Prefix

10.154.2.5 ReqId

uint16_t ReqId

The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

10.155 U3V_EVENT_DATA Struct Reference

U3V/GenCP EVENT_CMD specific command data.

Public Attributes

- uint16_t [Reserved](#)
- uint16_t [EventId](#)
- uint64_t [Timestamp](#)

10.155.1 Detailed Description

U3V/GenCP EVENT_CMD specific command data.

10.155.2 Member Data Documentation

10.155.2.1 EventId

uint16_t EventId

10.155.2.2 Reserved

uint16_t Reserved

10.155.2.3 Timestamp

uint64_t Timestamp

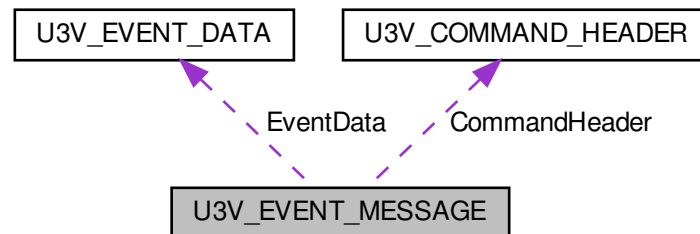
The documentation for this struct was generated from the following file:

- include/SpinGenApi/[EventAdapterU3V.h](#)

10.156 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`

10.156.1 Detailed Description

Entire event data message (without the variable-sized data field)

10.156.2 Member Data Documentation

10.156.2.1 CommandHeader

[U3V_COMMAND_HEADER](#) `CommandHeader`

10.156.2.2 EventData

[U3V_EVENT_DATA](#) `EventData`

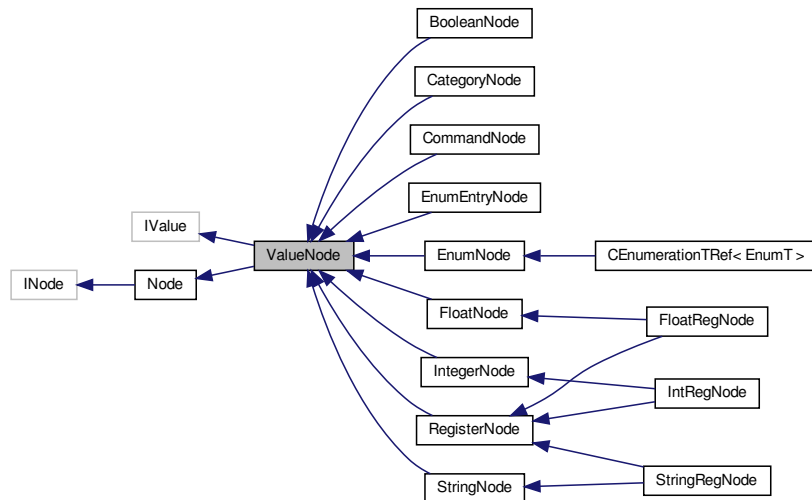
The documentation for this struct was generated from the following file:

- `include/SpinGenApi/EventAdapterU3V.h`

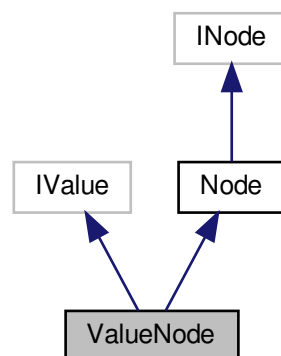
10.157 ValueNode Class Reference

[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

10.157.1 Detailed Description

[Interface](#) for value properties.

10.157.2 Constructor & Destructor Documentation

10.157.2.1 ValueNode() [1/2]

[ValueNode](#) ()

Constructor.

10.157.2.2 ValueNode() [2/2]

[ValueNode](#) (
 std::shared_ptr< [Node::NodeImpl](#) > pValue)

constructor with [GenICam IValue](#)

10.157.2.3 ~ValueNode()

[~ValueNode](#) ()

Destructor.

10.157.3 Member Function Documentation

10.157.3.1 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) |

10.157.3.2 GetNode()

```
virtual INode* GetNode ( ) [virtual]
```

10.157.3.3 IsValueCacheValid()

```
virtual bool IsValueCacheValid ( ) const [virtual]
```

Checks if the value comes from cache or is requested from another node.

10.157.3.4 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](#).

10.157.3.5 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](#)

10.158 Version_t Struct Reference

Version.

Public Attributes

- uint16_t [Major](#)
- uint16_t [Minor](#)
a is incompatible with b if $a \neq b$
- uint16_t [SubMinor](#)
a is incompatible b $a > b$

10.158.1 Detailed Description

Version.

10.158.2 Member Data Documentation

10.158.2.1 Major

uint16_t Major

10.158.2.2 Minor

uint16_t Minor

a is incompatible with b if $a \neq b$

10.158.2.3 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 11

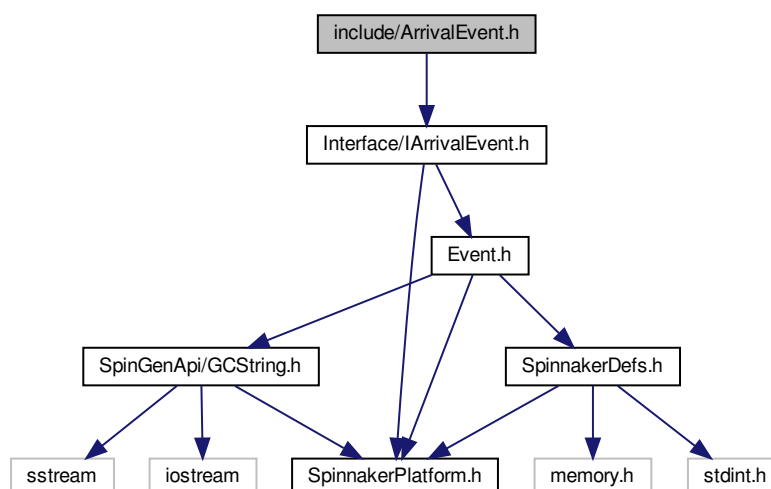
File Documentation

11.1 doc/Doxygen/spindocs/Licensing.dox File Reference

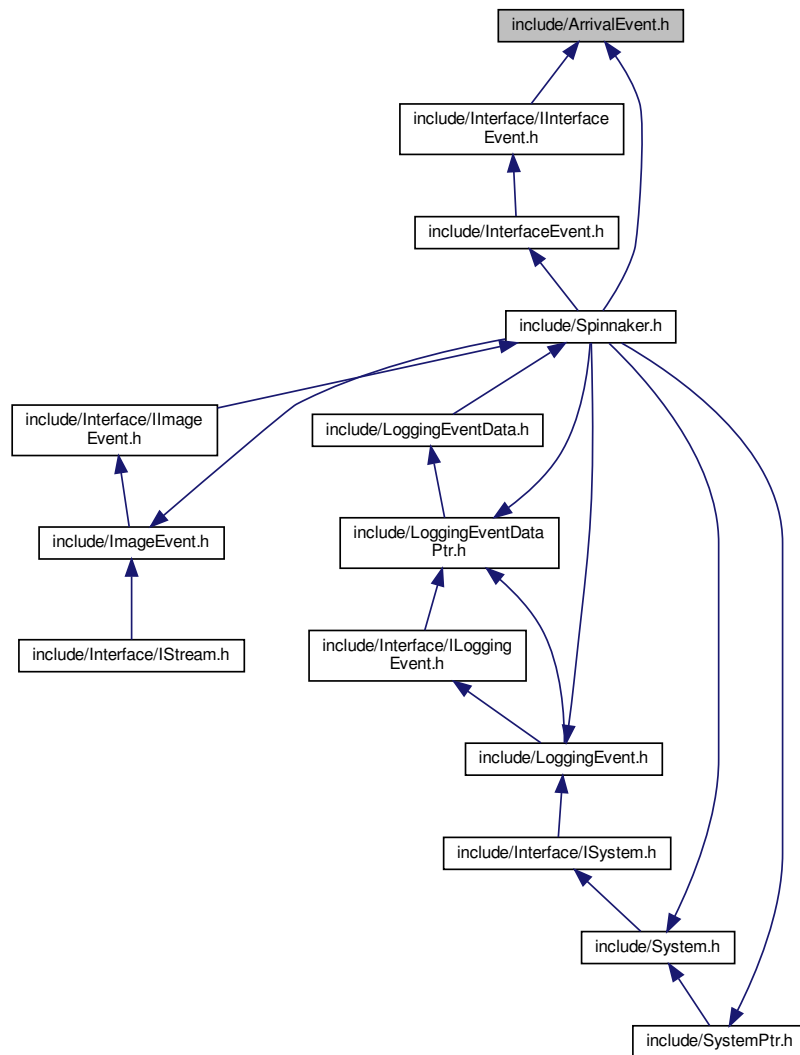
11.2 doc/Doxygen/spindocs/MainPage.dox File Reference

11.3 include/ArrivalEvent.h File Reference

Include dependency graph for ArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ArrivalEvent](#)

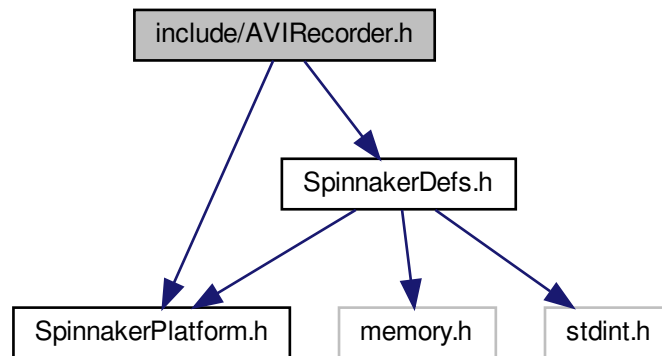
An event handler for capturing the device arrival event.

Namespaces

- [Spinnaker](#)

11.4 include/AVIRecorder.h File Reference

Include dependency graph for AVIRecorder.h:



Namespaces

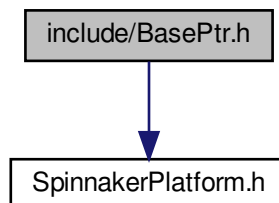
- [Spinnaker](#)

Functions

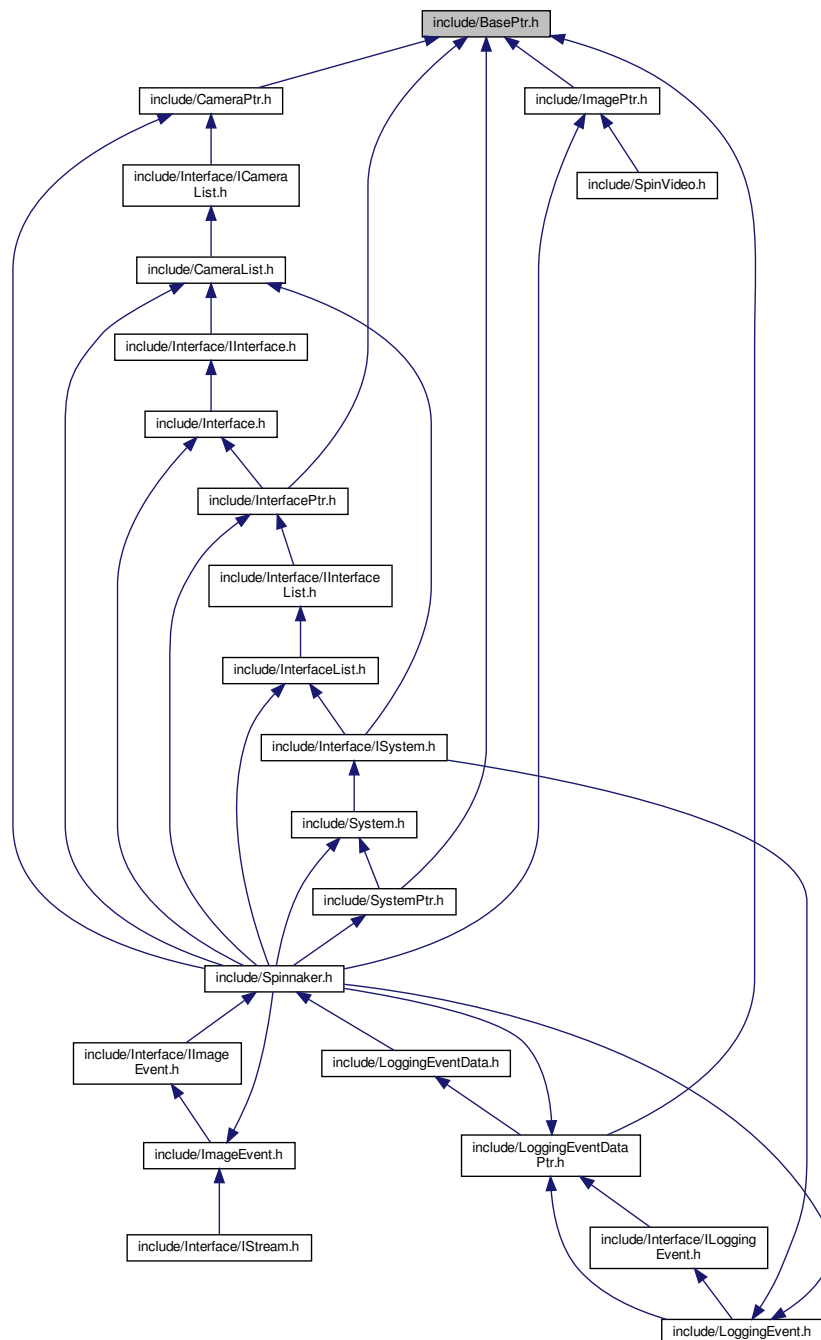
- class [DEPRECATED_CLASS](#) ("AVIRecorder is deprecated, use SpinVideo instead.") SPINNAKER_API \leftrightarrow `VIRecorder`
Provides the functionality for the user to record images to an AVI file.

11.5 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 >](#)

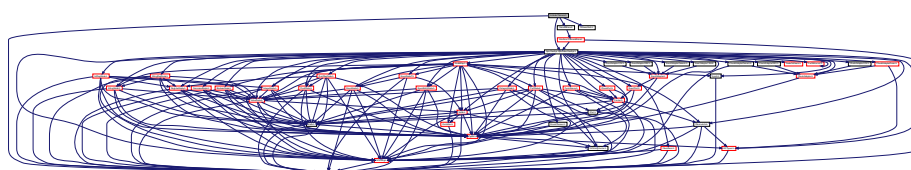
The base class of the [SystemPtr](#), [CameraPtr](#), [InterfacePtr](#), [ImagePtr](#) and [LoggingEventDataPtr](#) objects.

Namespaces

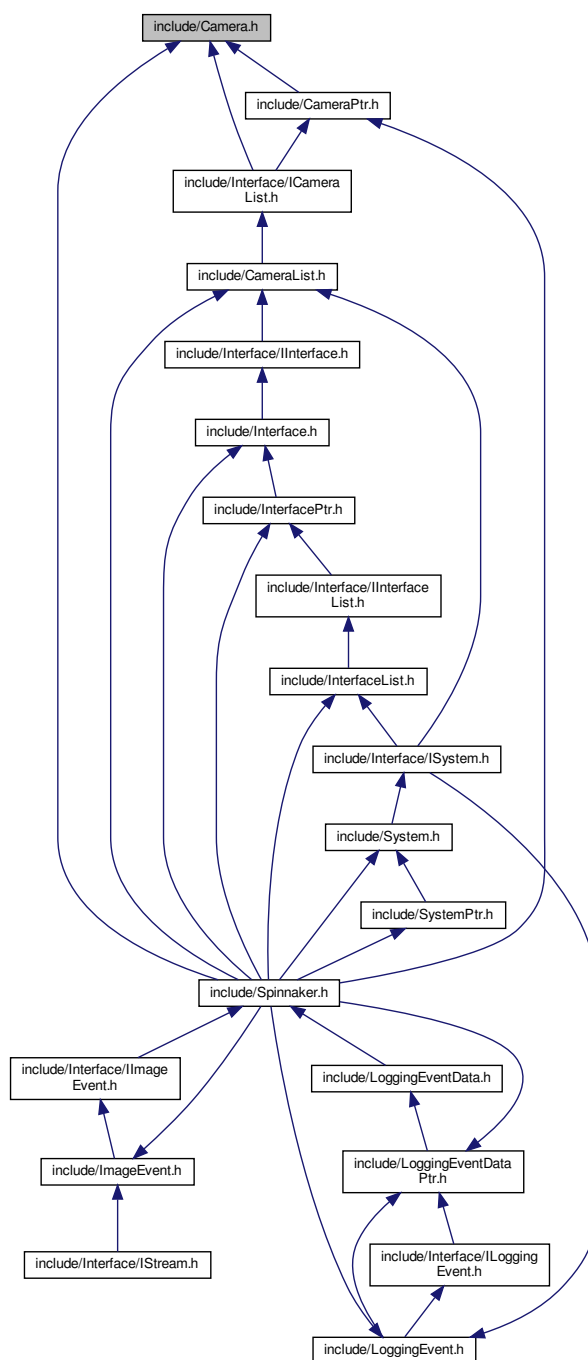
- [Spinnaker](#)

11.6 include/Camera.h File Reference

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



Classes

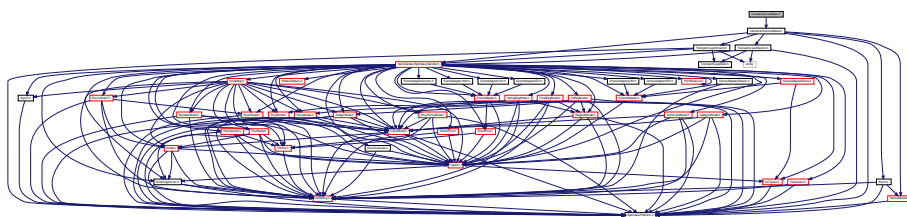
- class [Camera](#)
The camera object class.

Namespaces

- [Spinnaker](#)

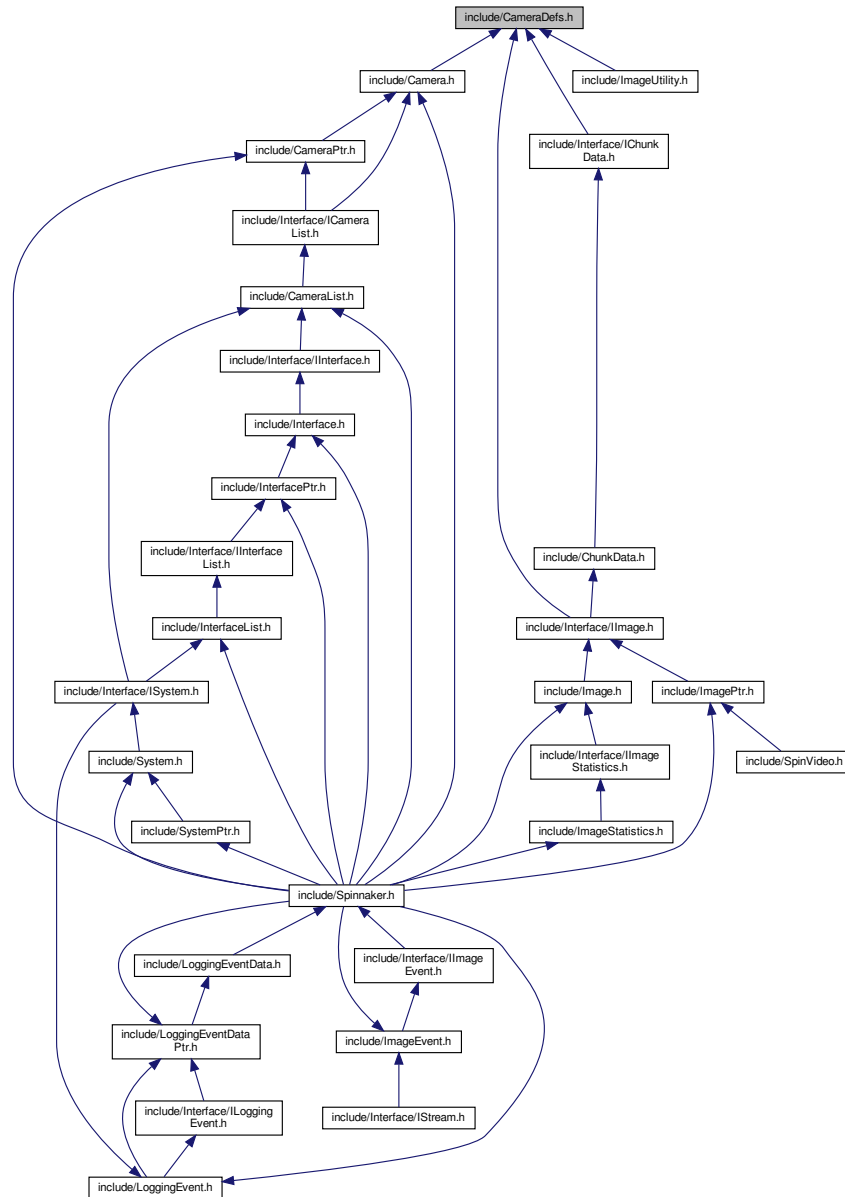
11.7 include/CameraBase.h File Reference

Include dependency graph for CameraBase.h:



11.8 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_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_Deglintch,
    LineInputFilterSelector_Debounce,
    NUM_LINEINPUTFILTERSELECTOR }

• enum UserOutputSelectorEnums {
    UserOutputSelector_UserOutput0,
    UserOutputSelector_UserOutput1,
    UserOutputSelector_UserOutput2,
    UserOutputSelector_UserOutput3,
    NUM_USEROUTPUTSELECTOR }

• enum LineFormatEnums {
    LineFormat_NoConnect,
    LineFormat_TriState,
    LineFormat_TTL,
    LineFormat_LVDS,
    LineFormat_RS422,
    LineFormat_OptoCoupled,
    LineFormat_OpenDrain,
    NUM_LINEFORMAT }

• enum LineSelectorEnums {
    LineSelector_Line0,
    LineSelector_Line1,
    LineSelector_Line2,
    LineSelector_Line3,
    NUM_LINESELECTOR }

• enum ExposureActiveModeEnums {
    ExposureActiveMode_Line1,
    ExposureActiveMode_AnyPixels,
    ExposureActiveMode_AllPixels,
    NUM_EXPOSUREACTIVEMODE }

• enum CounterTriggerActivationEnums {
    CounterTriggerActivation_LevelLow,
    CounterTriggerActivation_LevelHigh,
    CounterTriggerActivation_FallingEdge,
    CounterTriggerActivation_RisingEdge,
    CounterTriggerActivation_AnyEdge,
    NUM_COUNTERTRIGGERACTIVATION }

• enum CounterSelectorEnums {
    CounterSelector_Counter0,
    CounterSelector_Counter1,
    NUM_COUNTERSELECTOR }

• enum CounterStatusEnums {
    CounterStatus_CounterIdle,
    CounterStatus_CounterTriggerWait,

```

```
CounterStatus_CounterActive,  
CounterStatus_CounterCompleted,  
CounterStatus_CounterOverflow,  
NUM_COUNTERSTATUS }  
  
• enum CounterTriggerSourceEnums {  
CounterTriggerSource_Off,  
CounterTriggerSource_Line0,  
CounterTriggerSource_Line1,  
CounterTriggerSource_Line2,  
CounterTriggerSource_Line3,  
CounterTriggerSource_UserOutput0,  
CounterTriggerSource_UserOutput1,  
CounterTriggerSource_UserOutput2,  
CounterTriggerSource_UserOutput3,  
CounterTriggerSource_Counter0Start,  
CounterTriggerSource_Counter1Start,  
CounterTriggerSource_Counter0End,  
CounterTriggerSource_Counter1End,  
CounterTriggerSource_LogicBlock0,  
CounterTriggerSource_LogicBlock1,  
CounterTriggerSource_ExposureStart,  
CounterTriggerSource_ExposureEnd,  
CounterTriggerSource_FrameTriggerWait,  
NUM_COUNTERTRIGGERSOURCE }  
  
• enum CounterResetSourceEnums {  
CounterResetSource_Off,  
CounterResetSource_Line0,  
CounterResetSource_Line1,  
CounterResetSource_Line2,  
CounterResetSource_Line3,  
CounterResetSource_UserOutput0,  
CounterResetSource_UserOutput1,  
CounterResetSource_UserOutput2,  
CounterResetSource_UserOutput3,  
CounterResetSource_Counter0Start,  
CounterResetSource_Counter1Start,  
CounterResetSource_Counter0End,  
CounterResetSource_Counter1End,  
CounterResetSource_LogicBlock0,  
CounterResetSource_LogicBlock1,  
CounterResetSource_ExposureStart,  
CounterResetSource_ExposureEnd,  
CounterResetSource_FrameTriggerWait,  
NUM_COUNTERRESETSOURCE }  
  
• enum CounterEventSourceEnums {  
CounterEventSource_Off,  
CounterEventSource_MHzTick,  
CounterEventSource_Line0,  
CounterEventSource_Line1,  
CounterEventSource_Line2,  
CounterEventSource_Line3,  
CounterEventSource_UserOutput0,  
CounterEventSource_UserOutput1,  
CounterEventSource_UserOutput2,  
CounterEventSource_UserOutput3,  
CounterEventSource_Counter0Start,  
CounterEventSource_Counter1Start,  
CounterEventSource_Counter0End,
```



```

CounterEventSource_Counter1End,
CounterEventSource_LogicBlock0,
CounterEventSource_LogicBlock1,
CounterEventSource_ExposureStart,
CounterEventSource_ExposureEnd,
CounterEventSource_FrameTriggerWait,
NUM_COUNTEREVENTSOURCE }
• enum CounterEventActivationEnums {
CounterEventActivation_LevelLow,
CounterEventActivation_LevelHigh,
CounterEventActivation_FallingEdge,
CounterEventActivation_RisingEdge,
CounterEventActivation_AnyEdge,
NUM_COUNTEREVENTACTIVATION }
• enum CounterResetActivationEnums {
CounterResetActivation_LevelLow,
CounterResetActivation_LevelHigh,
CounterResetActivation_FallingEdge,
CounterResetActivation_RisingEdge,
CounterResetActivation_AnyEdge,
NUM_COUNTERRESETACTIVATION }
• enum DeviceTypeEnums {
DeviceType_Transmitter,
DeviceType_Receiver,
DeviceType_Transceiver,
DeviceType_Peripheral,
NUM_DEVICETYPE }
• enum DeviceConnectionStatusEnums {
DeviceConnectionStatus_Active,
DeviceConnectionStatus_Inactive,
NUM_DEVICECONNECTIONSTATUS }
• enum DeviceLinkThroughputLimitModeEnums {
DeviceLinkThroughputLimitMode_On,
DeviceLinkThroughputLimitMode_Off,
NUM_DEVICELINKTHROUGHPUTLIMITMODE }
• enum DeviceLinkHeartbeatModeEnums {
DeviceLinkHeartbeatMode_On,
DeviceLinkHeartbeatMode_Off,
NUM_DEVICELINKHEARTBEATMODE }
• enum DeviceStreamChannelTypeEnums {
DeviceStreamChannelType_Transmitter,
DeviceStreamChannelType_Receiver,
NUM_DEVICESTREAMCHANNELTYPE }
• enum DeviceStreamChannelEndiannessEnums {
DeviceStreamChannelEndianness_Big,
DeviceStreamChannelEndianness_Little,
NUM_DEVICESTREAMCHANNELENDIANNESS }
• enum DeviceClockSelectorEnums {
DeviceClockSelector_Sensor,
DeviceClockSelector_SensorDigitization,
DeviceClockSelector_CameraLink,
NUM_DEVICECLOCKSELECTOR }
• enum DeviceSerialPortSelectorEnums {
DeviceSerialPortSelector_CameraLink,
NUM_DEVICESERIALPORTSELECTOR }
• enum DeviceSerialPortBaudRateEnums {
DeviceSerialPortBaudRate_Baud9600,
DeviceSerialPortBaudRate_Baud19200,

```

```
DeviceSerialPortBaudRate_Baud38400,  
DeviceSerialPortBaudRate_Baud57600,  
DeviceSerialPortBaudRate_Baud115200,  
DeviceSerialPortBaudRate_Baud230400,  
DeviceSerialPortBaudRate_Baud460800,  
DeviceSerialPortBaudRate_Baud921600,  
NUM_DEVICESERIALPORTBAUDRATE }  
  
• enum SensorTapsEnums {  
    SensorTaps_One,  
    SensorTaps_Two,  
    SensorTaps_Three,  
    SensorTaps_Four,  
    SensorTaps_Eight,  
    SensorTaps_Ten,  
    NUM_SENSORTAPS }  
  
• enum SensorDigitizationTapsEnums {  
    SensorDigitizationTaps_One,  
    SensorDigitizationTaps_Two,  
    SensorDigitizationTaps_Three,  
    SensorDigitizationTaps_Four,  
    SensorDigitizationTaps_Eight,  
    SensorDigitizationTaps_Ten,  
    NUM_SENSORDIGITIZATIONTAPS }  
  
• enum RegionSelectorEnums {  
    RegionSelector_Region0,  
    RegionSelector_Region1,  
    RegionSelector_Region2,  
    RegionSelector_All,  
    NUM_REGIONSELECTOR }  
  
• enum RegionModeEnums {  
    RegionMode_Off,  
    RegionMode_On,  
    NUM_REGIONMODE }  
  
• enum RegionDestinationEnums {  
    RegionDestination_Stream0,  
    RegionDestination_Stream1,  
    RegionDestination_Stream2,  
    NUM_REGIONDESTINATION }  
  
• enum ImageComponentSelectorEnums {  
    ImageComponentSelector_Intensity,  
    ImageComponentSelector_Color,  
    ImageComponentSelector_Infrared,  
    ImageComponentSelector_Ultraviolet,  
    ImageComponentSelector_Range,  
    ImageComponentSelector_Disparity,  
    ImageComponentSelector_Confidence,  
    ImageComponentSelector_Scatter,  
    NUM_IMAGECOMPONENTSELECTOR }  
  
• enum PixelFormatInfoSelectorEnums {  
    PixelFormatInfoSelector_Mono1p,  
    PixelFormatInfoSelector_Mono2p,  
    PixelFormatInfoSelector_Mono4p,  
    PixelFormatInfoSelector_Mono8,  
    PixelFormatInfoSelector_Mono8s,  
    PixelFormatInfoSelector_Mono10,  
    PixelFormatInfoSelector_Mono10p,  
    PixelFormatInfoSelector_Mono12,  
    PixelFormatInfoSelector_Mono12p,
```

[PixelFormatInfoSelector_Mono14,](#)
[PixelFormatInfoSelector_Mono16,](#)
[PixelFormatInfoSelector_Mono16s,](#)
[PixelFormatInfoSelector_Mono32f,](#)
[PixelFormatInfoSelector_BayerBG8,](#)
[PixelFormatInfoSelector_BayerBG10,](#)
[PixelFormatInfoSelector_BayerBG10p,](#)
[PixelFormatInfoSelector_BayerBG12,](#)
[PixelFormatInfoSelector_BayerBG12p,](#)
[PixelFormatInfoSelector_BayerBG16,](#)
[PixelFormatInfoSelector_BayerGB8,](#)
[PixelFormatInfoSelector_BayerGB10,](#)
[PixelFormatInfoSelector_BayerGB10p,](#)
[PixelFormatInfoSelector_BayerGB12,](#)
[PixelFormatInfoSelector_BayerGB12p,](#)
[PixelFormatInfoSelector_BayerGB16,](#)
[PixelFormatInfoSelector_BayerGR8,](#)
[PixelFormatInfoSelector_BayerGR10,](#)
[PixelFormatInfoSelector_BayerGR10p,](#)
[PixelFormatInfoSelector_BayerGR12,](#)
[PixelFormatInfoSelector_BayerGR12p,](#)
[PixelFormatInfoSelector_BayerGR16,](#)
[PixelFormatInfoSelector_BayerRG8,](#)
[PixelFormatInfoSelector_BayerRG10,](#)
[PixelFormatInfoSelector_BayerRG10p,](#)
[PixelFormatInfoSelector_BayerRG12,](#)
[PixelFormatInfoSelector_BayerRG12p,](#)
[PixelFormatInfoSelector_BayerRG16,](#)
[PixelFormatInfoSelector_RGBa8,](#)
[PixelFormatInfoSelector_RGBa10,](#)
[PixelFormatInfoSelector_RGBa10p,](#)
[PixelFormatInfoSelector_RGBa12,](#)
[PixelFormatInfoSelector_RGBa12p,](#)
[PixelFormatInfoSelector_RGBa14,](#)
[PixelFormatInfoSelector_RGBa16,](#)
[PixelFormatInfoSelector_RGB8,](#)
[PixelFormatInfoSelector_RGB8_Planar,](#)
[PixelFormatInfoSelector_RGB10,](#)
[PixelFormatInfoSelector_RGB10_Planar,](#)
[PixelFormatInfoSelector_RGB10p,](#)
[PixelFormatInfoSelector_RGB10p32,](#)
[PixelFormatInfoSelector_RGB12,](#)
[PixelFormatInfoSelector_RGB12_Planar,](#)
[PixelFormatInfoSelector_RGB12p,](#)
[PixelFormatInfoSelector_RGB14,](#)
[PixelFormatInfoSelector_RGB16,](#)
[PixelFormatInfoSelector_RGB16s,](#)
[PixelFormatInfoSelector_RGB32f,](#)
[PixelFormatInfoSelector_RGB16_Planar,](#)
[PixelFormatInfoSelector_RGB565p,](#)
[PixelFormatInfoSelector_BGRa8,](#)
[PixelFormatInfoSelector_BGRa10,](#)
[PixelFormatInfoSelector_BGRa10p,](#)
[PixelFormatInfoSelector_BGRa12,](#)
[PixelFormatInfoSelector_BGRa12p,](#)
[PixelFormatInfoSelector_BGRa14,](#)
[PixelFormatInfoSelector_BGRa16,](#)
[PixelFormatInfoSelector_RGBa32f,](#)

PixelFormatInfoSelector_BGR8,
PixelFormatInfoSelector_BGR10,
PixelFormatInfoSelector_BGR10p,
PixelFormatInfoSelector_BGR12,
PixelFormatInfoSelector_BGR12p,
PixelFormatInfoSelector_BGR14,
PixelFormatInfoSelector_BGR16,
PixelFormatInfoSelector_BGR565p,
PixelFormatInfoSelector_R8,
PixelFormatInfoSelector_R10,
PixelFormatInfoSelector_R12,
PixelFormatInfoSelector_R16,
PixelFormatInfoSelector_G8,
PixelFormatInfoSelector_G10,
PixelFormatInfoSelector_G12,
PixelFormatInfoSelector_G16,
PixelFormatInfoSelector_B8,
PixelFormatInfoSelector_B10,
PixelFormatInfoSelector_B12,
PixelFormatInfoSelector_B16,
PixelFormatInfoSelector_Coord3D_ABC8,
PixelFormatInfoSelector_Coord3D_ABC8_Planar,
PixelFormatInfoSelector_Coord3D_ABC10p,
PixelFormatInfoSelector_Coord3D_ABC10p_Planar,
PixelFormatInfoSelector_Coord3D_ABC12p,
PixelFormatInfoSelector_Coord3D_ABC12p_Planar,
PixelFormatInfoSelector_Coord3D_ABC16,
PixelFormatInfoSelector_Coord3D_ABC16_Planar,
PixelFormatInfoSelector_Coord3D_ABC32f,
PixelFormatInfoSelector_Coord3D_ABC32f_Planar,
PixelFormatInfoSelector_Coord3D_AC8,
PixelFormatInfoSelector_Coord3D_AC8_Planar,
PixelFormatInfoSelector_Coord3D_AC10p,
PixelFormatInfoSelector_Coord3D_AC10p_Planar,
PixelFormatInfoSelector_Coord3D_AC12p,
PixelFormatInfoSelector_Coord3D_AC12p_Planar,
PixelFormatInfoSelector_Coord3D_AC16,
PixelFormatInfoSelector_Coord3D_AC16_Planar,
PixelFormatInfoSelector_Coord3D_AC32f,
PixelFormatInfoSelector_Coord3D_AC32f_Planar,
PixelFormatInfoSelector_Coord3D_A8,
PixelFormatInfoSelector_Coord3D_A10p,
PixelFormatInfoSelector_Coord3D_A12p,
PixelFormatInfoSelector_Coord3D_A16,
PixelFormatInfoSelector_Coord3D_A32f,
PixelFormatInfoSelector_Coord3D_B8,
PixelFormatInfoSelector_Coord3D_B10p,
PixelFormatInfoSelector_Coord3D_B12p,
PixelFormatInfoSelector_Coord3D_B16,
PixelFormatInfoSelector_Coord3D_B32f,
PixelFormatInfoSelector_Coord3D_C8,
PixelFormatInfoSelector_Coord3D_C10p,
PixelFormatInfoSelector_Coord3D_C12p,
PixelFormatInfoSelector_Coord3D_C16,
PixelFormatInfoSelector_Coord3D_C32f,
PixelFormatInfoSelector_Confidence1,
PixelFormatInfoSelector_Confidence1p,
PixelFormatInfoSelector_Confidence8,

[PixelFormatInfoSelector_Confidence16,](#)
[PixelFormatInfoSelector_Confidence32f,](#)
[PixelFormatInfoSelector_BiColorBGRG8,](#)
[PixelFormatInfoSelector_BiColorBGRG10,](#)
[PixelFormatInfoSelector_BiColorBGRG10p,](#)
[PixelFormatInfoSelector_BiColorBGRG12,](#)
[PixelFormatInfoSelector_BiColorBGRG12p,](#)
[PixelFormatInfoSelector_BiColorRGBG8,](#)
[PixelFormatInfoSelector_BiColorRGBG10,](#)
[PixelFormatInfoSelector_BiColorRGBG10p,](#)
[PixelFormatInfoSelector_BiColorRGBG12,](#)
[PixelFormatInfoSelector_BiColorRGBG12p,](#)
[PixelFormatInfoSelector_SCF1WBWG8,](#)
[PixelFormatInfoSelector_SCF1WBWG10,](#)
[PixelFormatInfoSelector_SCF1WBWG10p,](#)
[PixelFormatInfoSelector_SCF1WBWG12,](#)
[PixelFormatInfoSelector_SCF1WBWG12p,](#)
[PixelFormatInfoSelector_SCF1WBWG14,](#)
[PixelFormatInfoSelector_SCF1WBWG16,](#)
[PixelFormatInfoSelector_SCF1WGWB8,](#)
[PixelFormatInfoSelector_SCF1WGWB10,](#)
[PixelFormatInfoSelector_SCF1WGWB10p,](#)
[PixelFormatInfoSelector_SCF1WGWB12,](#)
[PixelFormatInfoSelector_SCF1WGWB12p,](#)
[PixelFormatInfoSelector_SCF1WGWB14,](#)
[PixelFormatInfoSelector_SCF1WGWB16,](#)
[PixelFormatInfoSelector_SCF1WGWR8,](#)
[PixelFormatInfoSelector_SCF1WGWR10,](#)
[PixelFormatInfoSelector_SCF1WGWR10p,](#)
[PixelFormatInfoSelector_SCF1WGWR12,](#)
[PixelFormatInfoSelector_SCF1WGWR12p,](#)
[PixelFormatInfoSelector_SCF1WGWR14,](#)
[PixelFormatInfoSelector_SCF1WGWR16,](#)
[PixelFormatInfoSelector_SCF1WRWG8,](#)
[PixelFormatInfoSelector_SCF1WRWG10,](#)
[PixelFormatInfoSelector_SCF1WRWG10p,](#)
[PixelFormatInfoSelector_SCF1WRWG12,](#)
[PixelFormatInfoSelector_SCF1WRWG12p,](#)
[PixelFormatInfoSelector_SCF1WRWG14,](#)
[PixelFormatInfoSelector_SCF1WRWG16,](#)
[PixelFormatInfoSelector_YCbCr8,](#)
[PixelFormatInfoSelector_YCbCr8_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr10_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr10p_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr12_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr12p_CbYCr,](#)
[PixelFormatInfoSelector_YCbCr411_8,](#)
[PixelFormatInfoSelector_YCbCr411_8_CbYYCrYY,](#)
[PixelFormatInfoSelector_YCbCr422_8,](#)
[PixelFormatInfoSelector_YCbCr422_8_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_10,](#)
[PixelFormatInfoSelector_YCbCr422_10_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_10p,](#)
[PixelFormatInfoSelector_YCbCr422_10p_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_12,](#)
[PixelFormatInfoSelector_YCbCr422_12_CbYCrY,](#)
[PixelFormatInfoSelector_YCbCr422_12p,](#)
[PixelFormatInfoSelector_YCbCr422_12p_CbYCrY,](#)

```

PixelFormatInfoSelector_YCbCr601_8_CbYCr,
PixelFormatInfoSelector_YCbCr601_10_CbYCr,
PixelFormatInfoSelector_YCbCr601_10p_CbYCr,
PixelFormatInfoSelector_YCbCr601_12_CbYCr,
PixelFormatInfoSelector_YCbCr601_12p_CbYCr,
PixelFormatInfoSelector_YCbCr601_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr601_422_8,
PixelFormatInfoSelector_YCbCr601_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10,
PixelFormatInfoSelector_YCbCr601_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_10p,
PixelFormatInfoSelector_YCbCr601_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12,
PixelFormatInfoSelector_YCbCr601_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr601_422_12p,
PixelFormatInfoSelector_YCbCr601_422_12p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_8_CbYCr,
PixelFormatInfoSelector_YCbCr709_10_CbYCr,
PixelFormatInfoSelector_YCbCr709_10p_CbYCr,
PixelFormatInfoSelector_YCbCr709_12_CbYCr,
PixelFormatInfoSelector_YCbCr709_12p_CbYCr,
PixelFormatInfoSelector_YCbCr709_411_8_CbYYCrYY,
PixelFormatInfoSelector_YCbCr709_422_8,
PixelFormatInfoSelector_YCbCr709_422_8_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10,
PixelFormatInfoSelector_YCbCr709_422_10_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_10p,
PixelFormatInfoSelector_YCbCr709_422_10p_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12,
PixelFormatInfoSelector_YCbCr709_422_12_CbYCrY,
PixelFormatInfoSelector_YCbCr709_422_12p,
PixelFormatInfoSelector_YCbCr709_422_12p_CbYCrY,
PixelFormatInfoSelector_YUV8_UYV,
PixelFormatInfoSelector_YUV411_8_UYYVYY,
PixelFormatInfoSelector_YUV422_8,
PixelFormatInfoSelector_YUV422_8_UYVY,
PixelFormatInfoSelector_Polarized8,
PixelFormatInfoSelector_Polarized10p,
PixelFormatInfoSelector_Polarized12p,
PixelFormatInfoSelector_Polarized16,
PixelFormatInfoSelector_BayerRGPolarized8,
PixelFormatInfoSelector_BayerRGPolarized10p,
PixelFormatInfoSelector_BayerRGPolarized12p,
PixelFormatInfoSelector_BayerRGPolarized16,
PixelFormatInfoSelector_LLCMono8,
PixelFormatInfoSelector_LLCBayerRG8,
PixelFormatInfoSelector_JPEGMono8,
PixelFormatInfoSelector_JPEGColor8,
NUM_PIXELFORMATINFOSELECTOR }

```

- enum DeinterlacingEnums {


```

Deinterlacing_Off,
Deinterlacing_LineDuplication,
Deinterlacing_Weave,
NUM_DEINTERLACING }

```
- enum ImageCompressionRateOptionEnums {


```

ImageCompressionRateOption_FixBitrate,
ImageCompressionRateOption_FixQuality,
NUM_IMAGECOMPRESSIONRATEOPTION }

```

- enum ImageCompressionJPEGFormatOptionEnums {
ImageCompressionJPEGFormatOption_Lossless,
ImageCompressionJPEGFormatOption_BaselineStandard,
ImageCompressionJPEGFormatOption_BaselineOptimized,
ImageCompressionJPEGFormatOption_Progressive,
NUM_IMAGECOMPRESSIONJPEGFORMATOPTION }
- enum AcquisitionStatusSelectorEnums {
AcquisitionStatusSelector_AcquisitionTriggerWait,
AcquisitionStatusSelector_AcquisitionActive,
AcquisitionStatusSelector_AcquisitionTransfer,
AcquisitionStatusSelector_FrameTriggerWait,
AcquisitionStatusSelector_FrameActive,
AcquisitionStatusSelector_ExposureActive,
NUM_ACQUISITIONSTATUSSELECTOR }
- enum ExposureTimeModeEnums {
ExposureTimeMode_Common,
ExposureTimeMode_Individual,
NUM_EXPOSURETIMEMODE }
- enum ExposureTimeSelectorEnums {
ExposureTimeSelector_Common,
ExposureTimeSelector_Red,
ExposureTimeSelector_Green,
ExposureTimeSelector_Blue,
ExposureTimeSelector_Cyan,
ExposureTimeSelector_Magenta,
ExposureTimeSelector_Yellow,
ExposureTimeSelector_Infrared,
ExposureTimeSelector_Ultraviolet,
ExposureTimeSelector_Stage1,
ExposureTimeSelector_Stage2,
NUM_EXPOSURETIMESELECTOR }
- enum GainAutoBalanceEnums {
GainAutoBalance_Off,
GainAutoBalance_Once,
GainAutoBalance_Continuous,
NUM_GAINAUTOBALANCE }
- enum BlackLevelAutoEnums {
BlackLevelAuto_Off,
BlackLevelAuto_Once,
BlackLevelAuto_Continuous,
NUM_BLACKLEVELAUTO }
- enum BlackLevelAutoBalanceEnums {
BlackLevelAutoBalance_Off,
BlackLevelAutoBalance_Once,
BlackLevelAutoBalance_Continuous,
NUM_BLACKLEVELAUTOBALANCE }
- enum WhiteClipSelectorEnums {
WhiteClipSelector_All,
WhiteClipSelector_Red,
WhiteClipSelector_Green,
WhiteClipSelector_Blue,
WhiteClipSelector_Y,
WhiteClipSelector_U,
WhiteClipSelector_V,
WhiteClipSelector_Tap1,
WhiteClipSelector_Tap2,
NUM_WHITECLIPSELECTOR }

- `enum TimerSelectorEnums {
 TimerSelector_Timer0,
 TimerSelector_Timer1,
 TimerSelector_Timer2,
 NUM_TIMERSELECTOR }`
- `enum TimerStatusEnums {
 TimerStatus_TimerIdle,
 TimerStatus_TimerTriggerWait,
 TimerStatus_TimerActive,
 TimerStatus_TimerCompleted,
 NUM_TIMERSTATUS }`
- `enum TimerTriggerSourceEnums {
 TimerTriggerSource_Off,
 TimerTriggerSource_AcquisitionTrigger,
 TimerTriggerSource_AcquisitionStart,
 TimerTriggerSource_AcquisitionEnd,
 TimerTriggerSource_FrameTrigger,
 TimerTriggerSource_FrameStart,
 TimerTriggerSource_FrameEnd,
 TimerTriggerSource_FrameBurstStart,
 TimerTriggerSource_FrameBurstEnd,
 TimerTriggerSource_LineTrigger,
 TimerTriggerSource_LineStart,
 TimerTriggerSource_LineEnd,
 TimerTriggerSource_ExposureStart,
 TimerTriggerSource_ExposureEnd,
 TimerTriggerSource_Line0,
 TimerTriggerSource_Line1,
 TimerTriggerSource_Line2,
 TimerTriggerSource_UserOutput0,
 TimerTriggerSource_UserOutput1,
 TimerTriggerSource_UserOutput2,
 TimerTriggerSource_Counter0Start,
 TimerTriggerSource_Counter1Start,
 TimerTriggerSource_Counter2Start,
 TimerTriggerSource_Counter0End,
 TimerTriggerSource_Counter1End,
 TimerTriggerSource_Counter2End,
 TimerTriggerSource_Timer0Start,
 TimerTriggerSource_Timer1Start,
 TimerTriggerSource_Timer2Start,
 TimerTriggerSource_Timer0End,
 TimerTriggerSource_Timer1End,
 TimerTriggerSource_Timer2End,
 TimerTriggerSource_Encoder0,
 TimerTriggerSource_Encoder1,
 TimerTriggerSource_Encoder2,
 TimerTriggerSource_SoftwareSignal0,
 TimerTriggerSource_SoftwareSignal1,
 TimerTriggerSource_SoftwareSignal2,
 TimerTriggerSource_Action0,
 TimerTriggerSource_Action1,
 TimerTriggerSource_Action2,
 TimerTriggerSource_LinkTrigger0,
 TimerTriggerSource_LinkTrigger1,
 TimerTriggerSource_LinkTrigger2,
 NUM_TIMERTRIGGERSOURCE }`
- `enum TimerTriggerActivationEnums {`


```

TimerTriggerActivation_RisingEdge,
TimerTriggerActivation_FallingEdge,
TimerTriggerActivation_AnyEdge,
TimerTriggerActivation_LevelHigh,
TimerTriggerActivation_LevelLow,
NUM_TIMERTRIGGERACTIVATION }

• enum EncoderSelectorEnums {
EncoderSelector_Encoder0,
EncoderSelector_Encoder1,
EncoderSelector_Encoder2,
NUM_ENCODERSELECTOR }

• enum EncoderSourceAEnums {
EncoderSourceA_Off,
EncoderSourceA_Line0,
EncoderSourceA_Line1,
EncoderSourceA_Line2,
NUM_ENCODERSOURCEA }

• enum EncoderSourceBEnums {
EncoderSourceB_Off,
EncoderSourceB_Line0,
EncoderSourceB_Line1,
EncoderSourceB_Line2,
NUM_ENCODERSOURCEB }

• enum EncoderModeEnums {
EncoderMode_FourPhase,
EncoderMode_HighResolution,
NUM_ENCODERMODE }

• enum EncoderOutputModeEnums {
EncoderOutputMode_Off,
EncoderOutputMode_PositionUp,
EncoderOutputMode_PositionDown,
EncoderOutputMode_DirectionUp,
EncoderOutputMode_DirectionDown,
EncoderOutputMode_Motion,
NUM_ENCODEROUTPUTMODE }

• enum EncoderStatusEnums {
EncoderStatus_EncoderUp,
EncoderStatus_EncoderDown,
EncoderStatus_EncoderIdle,
EncoderStatus_EncoderStatic,
NUM_ENCODERSTATUS }

• enum EncoderResetSourceEnums {
EncoderResetSource_Off,
EncoderResetSource_AcquisitionTrigger,
EncoderResetSource_AcquisitionStart,
EncoderResetSource_AcquisitionEnd,
EncoderResetSource_FrameTrigger,
EncoderResetSource_FrameStart,
EncoderResetSource_FrameEnd,
EncoderResetSource_ExposureStart,
EncoderResetSource_ExposureEnd,
EncoderResetSource_Line0,
EncoderResetSource_Line1,
EncoderResetSource_Line2,
EncoderResetSource_Counter0Start,
EncoderResetSource_Counter1Start,
EncoderResetSource_Counter2Start,
EncoderResetSource_Counter0End,

```

```

EncoderResetSource_Counter1End,
EncoderResetSource_Counter2End,
EncoderResetSource_Timer0Start,
EncoderResetSource_Timer1Start,
EncoderResetSource_Timer2Start,
EncoderResetSource_Timer0End,
EncoderResetSource_Timer1End,
EncoderResetSource_Timer2End,
EncoderResetSource_UserOutput0,
EncoderResetSource_UserOutput1,
EncoderResetSource_UserOutput2,
EncoderResetSource_SoftwareSignal0,
EncoderResetSource_SoftwareSignal1,
EncoderResetSource_SoftwareSignal2,
EncoderResetSource_Action0,
EncoderResetSource_Action1,
EncoderResetSource_Action2,
EncoderResetSource_LinkTrigger0,
EncoderResetSource_LinkTrigger1,
EncoderResetSource_LinkTrigger2,
NUM_ENCODERRESETSOURCE }
• enum EncoderResetActivationEnums {
EncoderResetActivation_RisingEdge,
EncoderResetActivation_FallingEdge,
EncoderResetActivation_AnyEdge,
EncoderResetActivation_LevelHigh,
EncoderResetActivation_LevelLow,
NUM_ENCODERRESETACTIVATION }
• enum SoftwareSignalSelectorEnums {
SoftwareSignalSelector_SoftwareSignal0,
SoftwareSignalSelector_SoftwareSignal1,
SoftwareSignalSelector_SoftwareSignal2,
NUM_SOFTWARESIGNALSELECTOR }
• enum ActionUnconditionalModeEnums {
ActionUnconditionalMode_Off,
ActionUnconditionalMode_On,
NUM_ACTIONUNCONDITIONALMODE }
• enum SourceSelectorEnums {
SourceSelector_Source0,
SourceSelector_Source1,
SourceSelector_Source2,
SourceSelector_All,
NUM_SOURCESELECTOR }
• enum TransferSelectorEnums {
TransferSelector_Stream0,
TransferSelector_Stream1,
TransferSelector_Stream2,
TransferSelector_All,
NUM_TRANSFERSELECTOR }
• enum TransferTriggerSelectorEnums {
TransferTriggerSelector_TransferStart,
TransferTriggerSelector_TransferStop,
TransferTriggerSelector_TransferAbort,
TransferTriggerSelector_TransferPause,
TransferTriggerSelector_TransferResume,
TransferTriggerSelector_TransferActive,
TransferTriggerSelector_TransferBurstStart,
TransferTriggerSelector_TransferBurstStop,

```

```
NUM_TRANSFERTRIGGERSELECTOR }

• enum TransferTriggerModeEnums {
    TransferTriggerMode_Off,
    TransferTriggerMode_On,
    NUM_TRANSFERTRIGGERMODE }

• enum TransferTriggerSourceEnums {
    TransferTriggerSource_Line0,
    TransferTriggerSource_Line1,
    TransferTriggerSource_Line2,
    TransferTriggerSource_Counter0Start,
    TransferTriggerSource_Counter1Start,
    TransferTriggerSource_Counter2Start,
    TransferTriggerSource_Counter0End,
    TransferTriggerSource_Counter1End,
    TransferTriggerSource_Counter2End,
    TransferTriggerSource_Timer0Start,
    TransferTriggerSource_Timer1Start,
    TransferTriggerSource_Timer2Start,
    TransferTriggerSource_Timer0End,
    TransferTriggerSource_Timer1End,
    TransferTriggerSource_Timer2End,
    TransferTriggerSource_SoftwareSignal0,
    TransferTriggerSource_SoftwareSignal1,
    TransferTriggerSource_SoftwareSignal2,
    TransferTriggerSource_Action0,
    TransferTriggerSource_Action1,
    TransferTriggerSource_Action2,
    NUM_TRANSFERTRIGGERSOURCE }

• enum TransferTriggerActivationEnums {
    TransferTriggerActivation_RisingEdge,
    TransferTriggerActivation_FallingEdge,
    TransferTriggerActivation_AnyEdge,
    TransferTriggerActivation_LevelHigh,
    TransferTriggerActivation_LevelLow,
    NUM_TRANSFERTRIGGERACTIVATION }

• enum TransferStatusSelectorEnums {
    TransferStatusSelector_Streaming,
    TransferStatusSelector_Paused,
    TransferStatusSelector_Stopping,
    TransferStatusSelector_Stopped,
    TransferStatusSelector_QueueOverflow,
    NUM_TRANSFERSTATUSSELECTOR }

• enum TransferComponentSelectorEnums {
    TransferComponentSelector_Red,
    TransferComponentSelector_Green,
    TransferComponentSelector_Blue,
    TransferComponentSelector_All,
    NUM_TRANSFERCOMPONENTSELECTOR }

• enum Scan3dDistanceUnitEnums {
    Scan3dDistanceUnit_Millimeter,
    Scan3dDistanceUnit_Inch,
    NUM_SCAN3DDISTANCEUNIT }

• enum Scan3dCoordinateSystemEnums {
    Scan3dCoordinateSystem_Cartesian,
    Scan3dCoordinateSystem_Spherical,
    Scan3dCoordinateSystem_Cylindrical,
    NUM_SCAN3DCOORDINATESYSTEM }
```

- enum Scan3dOutputModeEnums {
 Scan3dOutputMode_UncalibratedC,
 Scan3dOutputMode_CalibratedABC_Grid,
 Scan3dOutputMode_CalibratedABC_PointCloud,
 Scan3dOutputMode_CalibratedAC,
 Scan3dOutputMode_CalibratedAC_Linescan,
 Scan3dOutputMode_CalibratedC,
 Scan3dOutputMode_CalibratedC_Linescan,
 Scan3dOutputMode_RectifiedC,
 Scan3dOutputMode_RectifiedC_Linescan,
 Scan3dOutputMode_DisparityC,
 Scan3dOutputMode_DisparityC_Linescan,
 NUM_SCAN3DOUTPUTMODE }
- enum Scan3dCoordinateSystemReferenceEnums {
 Scan3dCoordinateSystemReference_Anchor,
 Scan3dCoordinateSystemReference_Transformed,
 NUM_SCAN3DCOORDINATESYSTEMREFERENCE }
- enum Scan3dCoordinateSelectorEnums {
 Scan3dCoordinateSelector_CoordinateA,
 Scan3dCoordinateSelector_CoordinateB,
 Scan3dCoordinateSelector_CoordinateC,
 NUM_SCAN3DCOORDINATESELECTOR }
- enum Scan3dCoordinateTransformSelectorEnums {
 Scan3dCoordinateTransformSelector_RotationX,
 Scan3dCoordinateTransformSelector_RotationY,
 Scan3dCoordinateTransformSelector_RotationZ,
 Scan3dCoordinateTransformSelector_TranslationX,
 Scan3dCoordinateTransformSelector_TranslationY,
 Scan3dCoordinateTransformSelector_TranslationZ,
 NUM_SCAN3DCOORDINATETRANSFORMSELECTOR }
- enum Scan3dCoordinateReferenceSelectorEnums {
 Scan3dCoordinateReferenceSelector_RotationX,
 Scan3dCoordinateReferenceSelector_RotationY,
 Scan3dCoordinateReferenceSelector_RotationZ,
 Scan3dCoordinateReferenceSelector_TranslationX,
 Scan3dCoordinateReferenceSelector_TranslationY,
 Scan3dCoordinateReferenceSelector_TranslationZ,
 NUM_SCAN3DCOORDINATEREFERENCESELECTOR }
- enum ChunkImageComponentEnums {
 ChunkImageComponent_Intensity,
 ChunkImageComponent_Color,
 ChunkImageComponent_Infrared,
 ChunkImageComponent_Ultraviolet,
 ChunkImageComponent_Range,
 ChunkImageComponent_Disparity,
 ChunkImageComponent_Confidence,
 ChunkImageComponent_Scatter,
 NUM_CHUNKIMAGECOMPONENT }
- enum ChunkCounterSelectorEnums {
 ChunkCounterSelector_Counter0,
 ChunkCounterSelector_Counter1,
 ChunkCounterSelector_Counter2,
 NUM_CHUNKCOUNTERSELECTOR }
- enum ChunkTimerSelectorEnums {
 ChunkTimerSelector_Timer0,
 ChunkTimerSelector_Timer1,
 ChunkTimerSelector_Timer2,
 NUM_CHUNKTIMERSELECTOR }

- enum `ChunkEncoderSelectorEnums` {
`ChunkEncoderSelector_Encoder0`,
`ChunkEncoderSelector_Encoder1`,
`ChunkEncoderSelector_Encoder2`,
`NUM_CHUNKENCODERSELECTOR` }
- enum `ChunkEncoderStatusEnums` {
`ChunkEncoderStatus_EncoderUp`,
`ChunkEncoderStatus_EncoderDown`,
`ChunkEncoderStatus_EncoderIdle`,
`ChunkEncoderStatus_EncoderStatic`,
`NUM_CHUNKENCODERSTATUS` }
- enum `ChunkExposureTimeSelectorEnums` {
`ChunkExposureTimeSelector_Common`,
`ChunkExposureTimeSelector_Red`,
`ChunkExposureTimeSelector_Green`,
`ChunkExposureTimeSelector_Blue`,
`ChunkExposureTimeSelector_Cyan`,
`ChunkExposureTimeSelector_Magenta`,
`ChunkExposureTimeSelector_Yellow`,
`ChunkExposureTimeSelector_Infrared`,
`ChunkExposureTimeSelector_Ultraviolet`,
`ChunkExposureTimeSelector_Stage1`,
`ChunkExposureTimeSelector_Stage2`,
`NUM_CHUNKEXPOSURETIMESELECTOR` }
- enum `ChunkSourceIDEnums` {
`ChunkSourceID_Source0`,
`ChunkSourceID_Source1`,
`ChunkSourceID_Source2`,
`NUM_CHUNKSOURCEID` }
- enum `ChunkRegionIDEnums` {
`ChunkRegionID_Region0`,
`ChunkRegionID_Region1`,
`ChunkRegionID_Region2`,
`NUM_CHUNKREGIONID` }
- enum `ChunkTransferStreamIDEnums` {
`ChunkTransferStreamID_Stream0`,
`ChunkTransferStreamID_Stream1`,
`ChunkTransferStreamID_Stream2`,
`ChunkTransferStreamID_Stream3`,
`NUM_CHUNKTRANSFERSTREAMID` }
- enum `ChunkScan3dDistanceUnitEnums` {
`ChunkScan3dDistanceUnit_Millimeter`,
`ChunkScan3dDistanceUnit_Inch`,
`NUM_CHUNKSCAN3DDISTANCEUNIT` }
- enum `ChunkScan3dOutputModeEnums` {
`ChunkScan3dOutputMode_UncalibratedC`,
`ChunkScan3dOutputMode_CalibratedABC_Grid`,
`ChunkScan3dOutputMode_CalibratedABC_PointCloud`,
`ChunkScan3dOutputMode_CalibratedAC`,
`ChunkScan3dOutputMode_CalibratedAC_Linescan`,
`ChunkScan3dOutputMode_CalibratedC`,
`ChunkScan3dOutputMode_CalibratedC_Linescan`,
`ChunkScan3dOutputMode_RectifiedC`,
`ChunkScan3dOutputMode_RectifiedC_Linescan`,
`ChunkScan3dOutputMode_DisparityC`,
`ChunkScan3dOutputMode_DisparityC_Linescan`,
`NUM_CHUNKSCAN3DOUTPUTMODE` }

- enum ChunkScan3dCoordinateSystemEnums {
 ChunkScan3dCoordinateSystem_Cartesian,
 ChunkScan3dCoordinateSystem_Spherical,
 ChunkScan3dCoordinateSystem_Cylindrical,
 NUM_CHUNKSCAN3DCOORDINATESYSTEM }
- enum ChunkScan3dCoordinateSystemReferenceEnums {
 ChunkScan3dCoordinateSystemReference_Anchor,
 ChunkScan3dCoordinateSystemReference_Transformed,
 NUM_CHUNKSCAN3DCOORDINATESYSTEMREFERENCE }
- enum ChunkScan3dCoordinateSelectorEnums {
 ChunkScan3dCoordinateSelector_CoordinateA,
 ChunkScan3dCoordinateSelector_CoordinateB,
 ChunkScan3dCoordinateSelector_CoordinateC,
 NUM_CHUNKSCAN3DCOORDINATESELECTOR }
- enum ChunkScan3dCoordinateTransformSelectorEnums {
 ChunkScan3dCoordinateTransformSelector_RotationX,
 ChunkScan3dCoordinateTransformSelector_RotationY,
 ChunkScan3dCoordinateTransformSelector_RotationZ,
 ChunkScan3dCoordinateTransformSelector_TranslationX,
 ChunkScan3dCoordinateTransformSelector_TranslationY,
 ChunkScan3dCoordinateTransformSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATETRANSFORMSELECTOR }
- enum ChunkScan3dCoordinateReferenceSelectorEnums {
 ChunkScan3dCoordinateReferenceSelector_RotationX,
 ChunkScan3dCoordinateReferenceSelector_RotationY,
 ChunkScan3dCoordinateReferenceSelector_RotationZ,
 ChunkScan3dCoordinateReferenceSelector_TranslationX,
 ChunkScan3dCoordinateReferenceSelector_TranslationY,
 ChunkScan3dCoordinateReferenceSelector_TranslationZ,
 NUM_CHUNKSCAN3DCOORDINATEREFERENCESELECTOR }
- enum DeviceTapGeometryEnums {
 DeviceTapGeometry_Geometry_1X_1Y,
 DeviceTapGeometry_Geometry_1X2_1Y,
 DeviceTapGeometry_Geometry_1X2_1Y2,
 DeviceTapGeometry_Geometry_2X_1Y,
 DeviceTapGeometry_Geometry_2X_1Y2Geometry_2XE_1Y,
 DeviceTapGeometry_Geometry_2XE_1Y2,
 DeviceTapGeometry_Geometry_2XM_1Y,
 DeviceTapGeometry_Geometry_2XM_1Y2,
 DeviceTapGeometry_Geometry_1X_1Y2,
 DeviceTapGeometry_Geometry_1X_2YE,
 DeviceTapGeometry_Geometry_1X3_1Y,
 DeviceTapGeometry_Geometry_3X_1Y,
 DeviceTapGeometry_Geometry_1X,
 DeviceTapGeometry_Geometry_1X2,
 DeviceTapGeometry_Geometry_2X,
 DeviceTapGeometry_Geometry_2XE,
 DeviceTapGeometry_Geometry_2XM,
 DeviceTapGeometry_Geometry_1X3,
 DeviceTapGeometry_Geometry_3X,
 DeviceTapGeometry_Geometry_1X4_1Y,
 DeviceTapGeometry_Geometry_4X_1Y,
 DeviceTapGeometry_Geometry_2X2_1Y,
 DeviceTapGeometry_Geometry_2X2E_1YGeometry_2X2M_1Y,
 DeviceTapGeometry_Geometry_1X2_2YE,
 DeviceTapGeometry_Geometry_2X_2YE,
 DeviceTapGeometry_Geometry_2XE_2YE,
 DeviceTapGeometry_Geometry_2XM_2YE,

```

DeviceTapGeometry_Geometry_1X4,
DeviceTapGeometry_Geometry_4X,
DeviceTapGeometry_Geometry_2X2,
DeviceTapGeometry_Geometry_2X2E,
DeviceTapGeometry_Geometry_2X2M,
DeviceTapGeometry_Geometry_1X8_1Y,
DeviceTapGeometry_Geometry_8X_1Y,
DeviceTapGeometry_Geometry_4X2_1Y,
DeviceTapGeometry_Geometry_2X2E_2YE,
DeviceTapGeometry_Geometry_1X8,
DeviceTapGeometry_Geometry_8X,
DeviceTapGeometry_Geometry_4X2,
DeviceTapGeometry_Geometry_4X2E,
DeviceTapGeometry_Geometry_4X2E_1Y,
DeviceTapGeometry_Geometry_1X10_1Y,
DeviceTapGeometry_Geometry_10X_1Y,
DeviceTapGeometry_Geometry_1X10,
DeviceTapGeometry_Geometry_10X,
NUM_DEVICETAPGEOMETRY }

• enum GevPhysicalLinkConfigurationEnums {
    GevPhysicalLinkConfiguration_SingleLink,
    GevPhysicalLinkConfiguration_MultiLink,
    GevPhysicalLinkConfiguration_StaticLAG,
    GevPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVPHYSICALLINKCONFIGURATION }

• enum GevCurrentPhysicalLinkConfigurationEnums {
    GevCurrentPhysicalLinkConfiguration_SingleLink,
    GevCurrentPhysicalLinkConfiguration_MultiLink,
    GevCurrentPhysicalLinkConfiguration_StaticLAG,
    GevCurrentPhysicalLinkConfiguration_DynamicLAG,
    NUM_GEVCURRENTPHYSICALLINKCONFIGURATION }

• enum GevIPConfigurationStatusEnums {
    GevIPConfigurationStatus_None,
    GevIPConfigurationStatus_PersistentIP,
    GevIPConfigurationStatus_DHCP,
    GevIPConfigurationStatus_LLA,
    GevIPConfigurationStatus_ForceIP,
    NUM_GEVIPCONFIGURATIONSTATUS }

• enum GevGVCPExtendedStatusCodesSelectorEnums {
    GevGVCPExtendedStatusCodesSelector_Version1_1,
    GevGVCPExtendedStatusCodesSelector_Version2_0,
    NUM_GEVGVCPEXTENDEDSTATUSCODESSELECTOR }

• enum GevGVSPEExtendedIDModeEnums {
    GevGVSPEExtendedIDMode_Off,
    GevGVSPEExtendedIDMode_On,
    NUM_GEVGVSPEXTENDEDIDMODE }

• enum ClConfigurationEnums {
    ClConfiguration_Base,
    ClConfiguration_Medium,
    ClConfiguration_Full,
    ClConfiguration_DualBase,
    ClConfiguration_EightyBit,
    NUM_CLCONFIGURATION }

• enum ClTimeSlotsCountEnums {
    ClTimeSlotsCount_One,
    ClTimeSlotsCount_Two,
    ClTimeSlotsCount_Three,
    NUM_CLTIMESLOTSCOUNT }

```

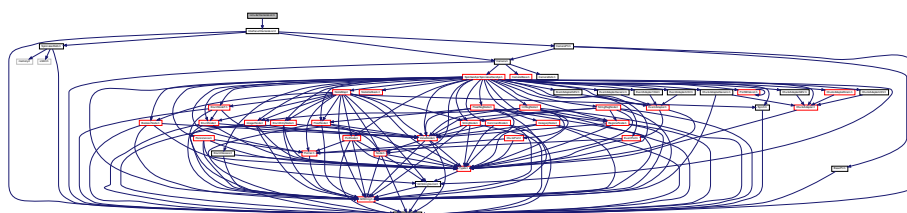
- enum CxpLinkConfigurationStatusEnums {
CxpLinkConfigurationStatus_None,
CxpLinkConfigurationStatus_Pending,
CxpLinkConfigurationStatus_CXP1_X1,
CxpLinkConfigurationStatus_CXP2_X1,
CxpLinkConfigurationStatus_CXP3_X1,
CxpLinkConfigurationStatus_CXP5_X1,
CxpLinkConfigurationStatus_CXP6_X1,
CxpLinkConfigurationStatus_CXP1_X2,
CxpLinkConfigurationStatus_CXP2_X2,
CxpLinkConfigurationStatus_CXP3_X2,
CxpLinkConfigurationStatus_CXP5_X2,
CxpLinkConfigurationStatus_CXP6_X2,
CxpLinkConfigurationStatus_CXP1_X3,
CxpLinkConfigurationStatus_CXP2_X3,
CxpLinkConfigurationStatus_CXP3_X3,
CxpLinkConfigurationStatus_CXP5_X3,
CxpLinkConfigurationStatus_CXP6_X3,
CxpLinkConfigurationStatus_CXP1_X4,
CxpLinkConfigurationStatus_CXP2_X4,
CxpLinkConfigurationStatus_CXP3_X4,
CxpLinkConfigurationStatus_CXP5_X4,
CxpLinkConfigurationStatus_CXP6_X4,
CxpLinkConfigurationStatus_CXP1_X5,
CxpLinkConfigurationStatus_CXP2_X5,
CxpLinkConfigurationStatus_CXP3_X5,
CxpLinkConfigurationStatus_CXP5_X5,
CxpLinkConfigurationStatus_CXP6_X5,
CxpLinkConfigurationStatus_CXP1_X6,
CxpLinkConfigurationStatus_CXP2_X6,
CxpLinkConfigurationStatus_CXP3_X6,
CxpLinkConfigurationStatus_CXP5_X6,
CxpLinkConfigurationStatus_CXP6_X6,
NUM_CXPLINKCONFIGURATIONSTATUS }
- enum CxpLinkConfigurationPreferredEnums {
CxpLinkConfigurationPreferred_CXP1_X1,
CxpLinkConfigurationPreferred_CXP2_X1,
CxpLinkConfigurationPreferred_CXP3_X1,
CxpLinkConfigurationPreferred_CXP5_X1,
CxpLinkConfigurationPreferred_CXP6_X1,
CxpLinkConfigurationPreferred_CXP1_X2,
CxpLinkConfigurationPreferred_CXP2_X2,
CxpLinkConfigurationPreferred_CXP3_X2,
CxpLinkConfigurationPreferred_CXP5_X2,
CxpLinkConfigurationPreferred_CXP6_X2,
CxpLinkConfigurationPreferred_CXP1_X3,
CxpLinkConfigurationPreferred_CXP2_X3,
CxpLinkConfigurationPreferred_CXP3_X3,
CxpLinkConfigurationPreferred_CXP5_X3,
CxpLinkConfigurationPreferred_CXP6_X3,
CxpLinkConfigurationPreferred_CXP1_X4,
CxpLinkConfigurationPreferred_CXP2_X4,
CxpLinkConfigurationPreferred_CXP3_X4,
CxpLinkConfigurationPreferred_CXP5_X4,
CxpLinkConfigurationPreferred_CXP6_X4,
CxpLinkConfigurationPreferred_CXP1_X5,
CxpLinkConfigurationPreferred_CXP2_X5,
CxpLinkConfigurationPreferred_CXP3_X5,


```
CxpLinkConfigurationPreferred_CXP5_X5,  
CxpLinkConfigurationPreferred_CXP6_X5,  
CxpLinkConfigurationPreferred_CXP1_X6,  
CxpLinkConfigurationPreferred_CXP2_X6,  
CxpLinkConfigurationPreferred_CXP3_X6,  
CxpLinkConfigurationPreferred_CXP5_X6,  
CxpLinkConfigurationPreferred_CXP6_X6,  
NUM_CXPLINKCONFIGURATIONPREFERRED }
```

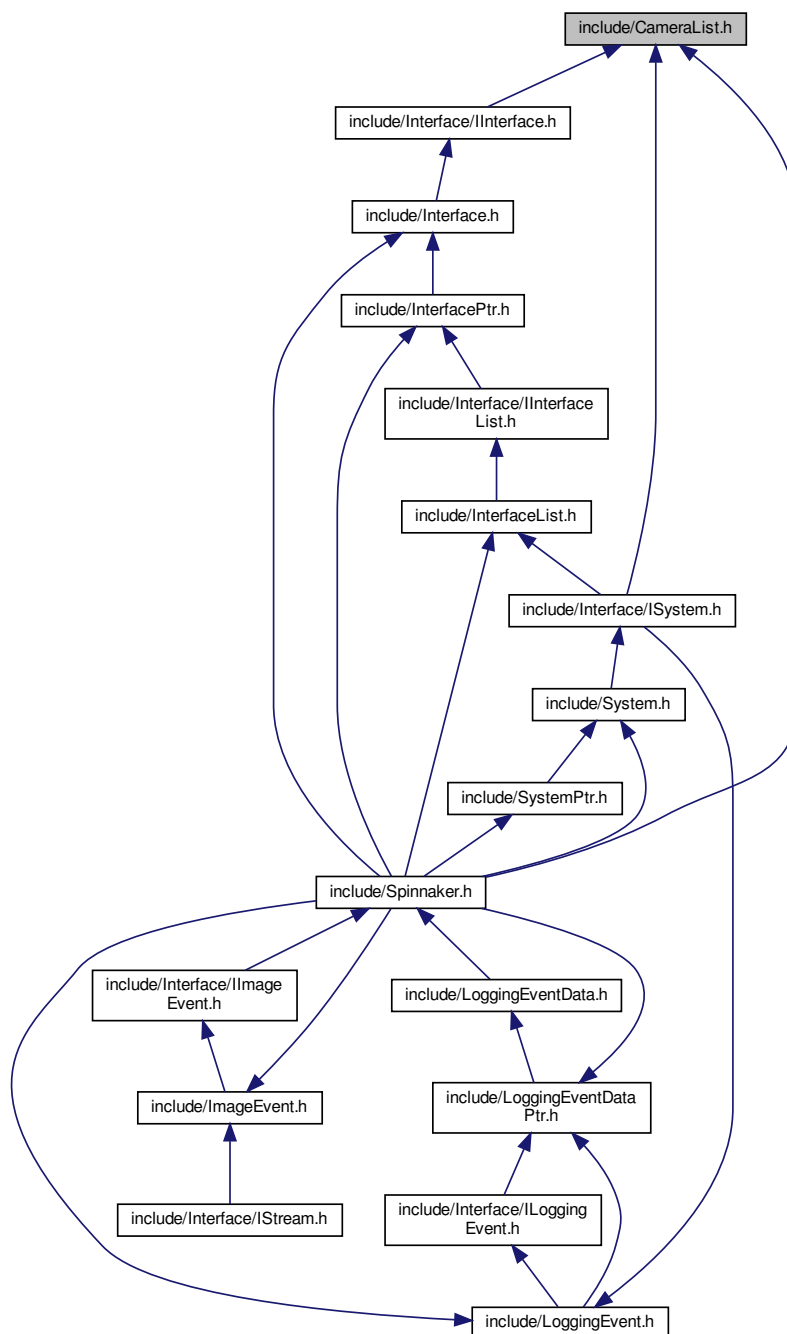
- `enum CxpLinkConfigurationEnums {`
 CxpLinkConfiguration_Auto,
 CxpLinkConfiguration_CXP1_X1,
 CxpLinkConfiguration_CXP2_X1,
 CxpLinkConfiguration_CXP3_X1,
 CxpLinkConfiguration_CXP5_X1,
 CxpLinkConfiguration_CXP6_X1,
 CxpLinkConfiguration_CXP1_X2,
 CxpLinkConfiguration_CXP2_X2,
 CxpLinkConfiguration_CXP3_X2,
 CxpLinkConfiguration_CXP5_X2,
 CxpLinkConfiguration_CXP6_X2,
 CxpLinkConfiguration_CXP1_X3,
 CxpLinkConfiguration_CXP2_X3,
 CxpLinkConfiguration_CXP3_X3,
 CxpLinkConfiguration_CXP5_X3,
 CxpLinkConfiguration_CXP6_X3,
 CxpLinkConfiguration_CXP1_X4,
 CxpLinkConfiguration_CXP2_X4,
 CxpLinkConfiguration_CXP3_X4,
 CxpLinkConfiguration_CXP5_X4,
 CxpLinkConfiguration_CXP6_X4,
 CxpLinkConfiguration_CXP1_X5,
 CxpLinkConfiguration_CXP2_X5,
 CxpLinkConfiguration_CXP3_X5,
 CxpLinkConfiguration_CXP5_X5,
 CxpLinkConfiguration_CXP6_X5,
 CxpLinkConfiguration_CXP1_X6,
 CxpLinkConfiguration_CXP2_X6,
 CxpLinkConfiguration_CXP3_X6,
 CxpLinkConfiguration_CXP5_X6,
 CxpLinkConfiguration_CXP6_X6,
 NUM_CXPLINKCONFIGURATION }
• `enum CxpConnectionTestModeEnums {`
 CxpConnectionTestMode_Off,
 CxpConnectionTestMode_Mode1,
 NUM_CXPCONNECTIONTESTMODE }
• `enum CxpPoCxpStatusEnums {`
 CxpPoCxpStatus_Auto,
 CxpPoCxpStatus_Off,
 CxpPoCxpStatus_Tripped,
 NUM_CXPPCXPSTATUS }

11.9 include/CameraList.h File Reference

Include dependency graph for CameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

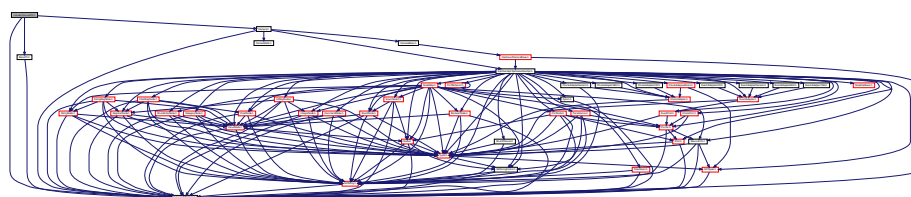
- class [CameraList](#)
Used to hold a list of camera objects.

Namespaces

- [Spinnaker](#)

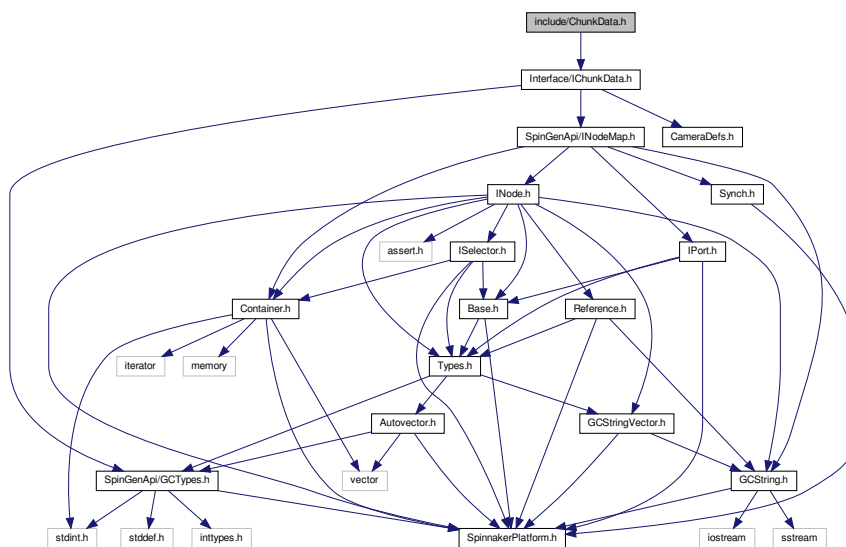
11.10 include/CameraPtr.h File Reference

Include dependency graph for CameraPtr.h:

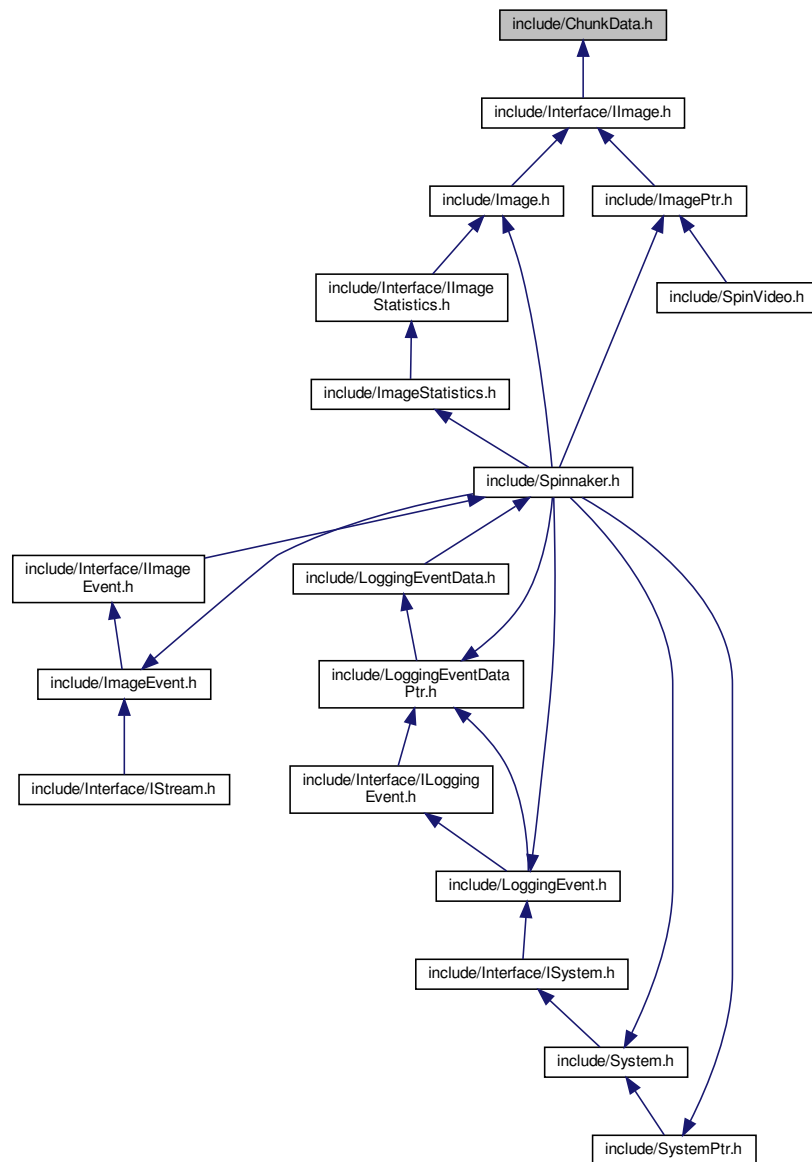


11.11 include/ChunkData.h File Reference

Include dependency graph for ChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ChunkData](#)

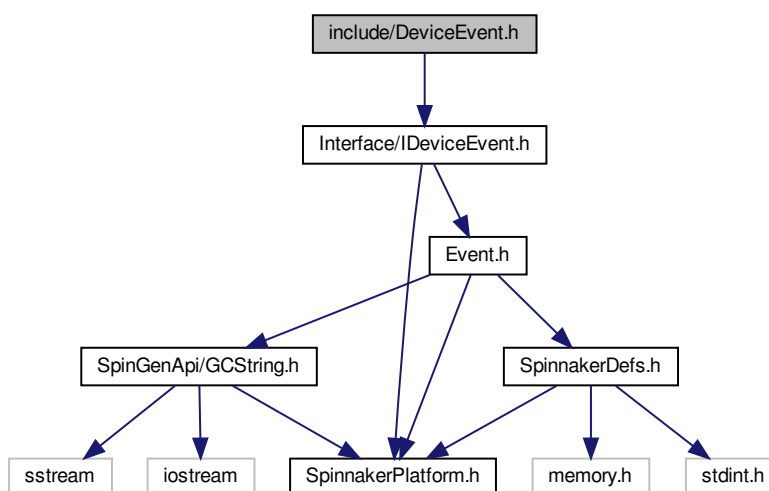
The chunk data which contains additional information about an image.

Namespaces

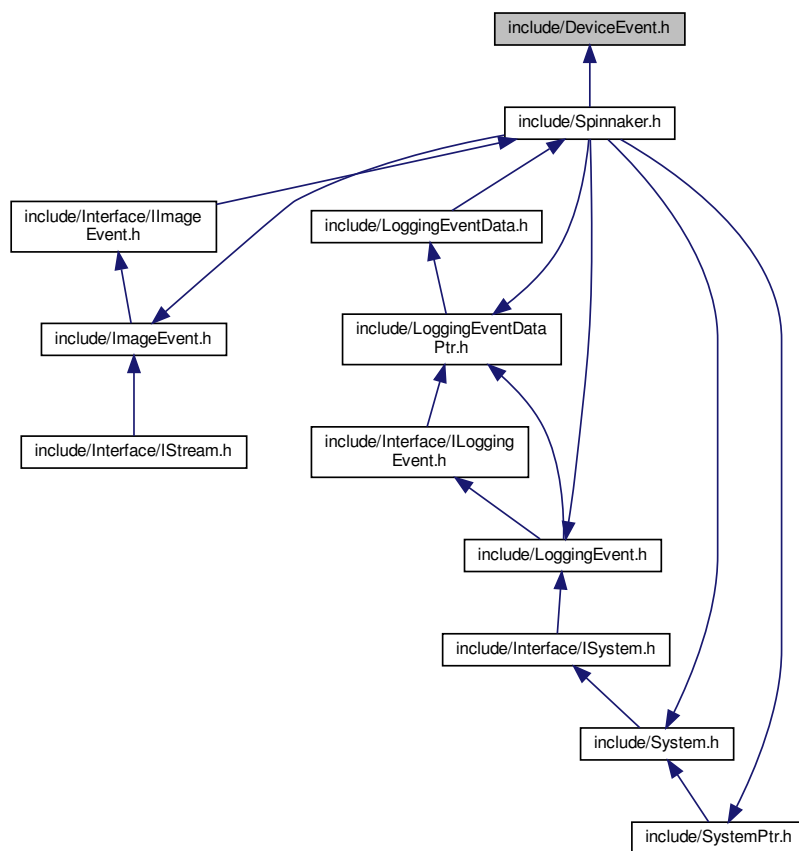
- [Spinnaker](#)

11.12 include/DeviceEvent.h File Reference

Include dependency graph for DeviceEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [DeviceEvent](#)

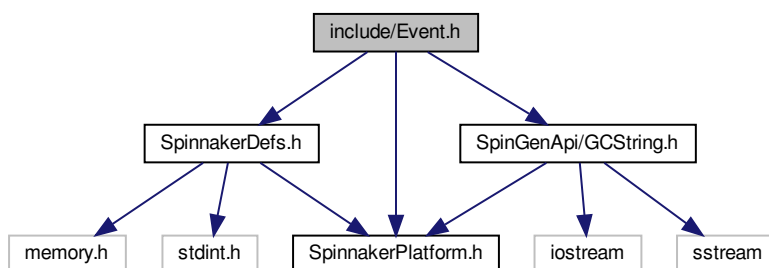
A handler to device events.

Namespaces

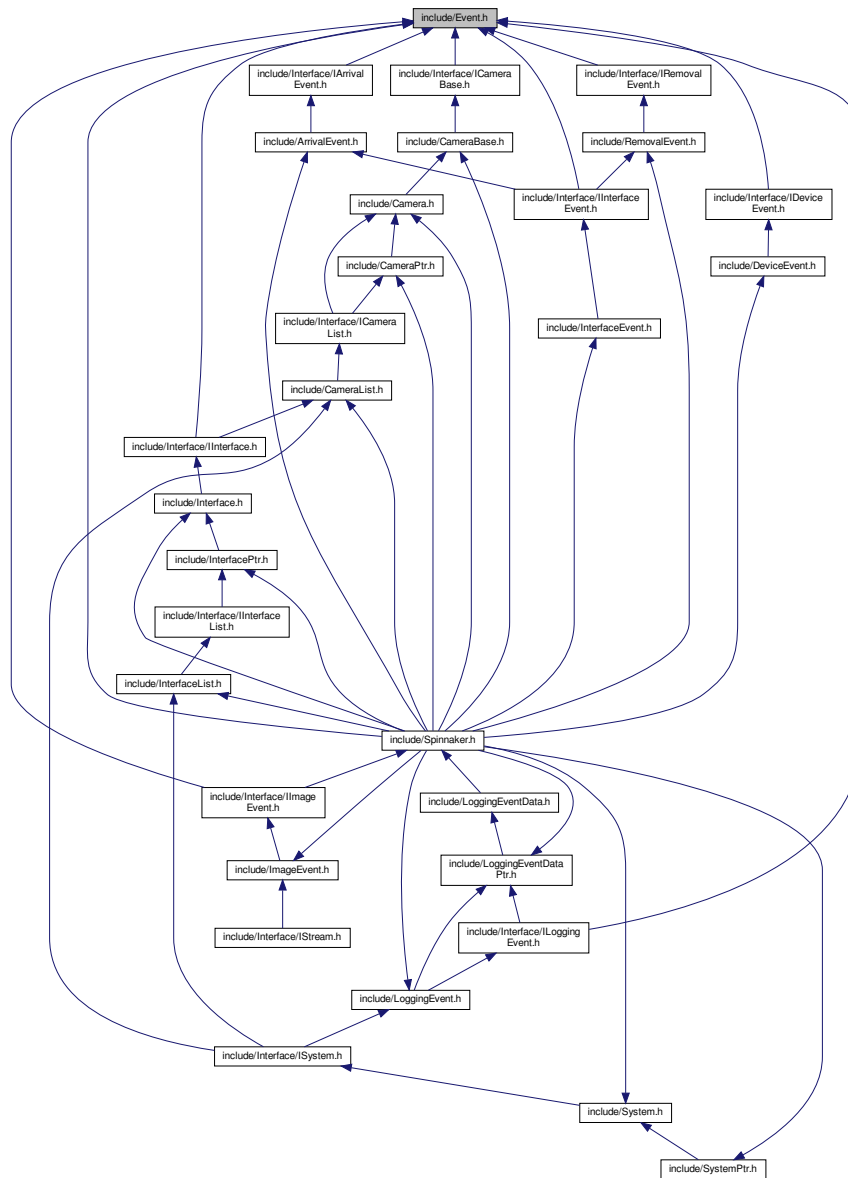
- [Spinnaker](#)

11.13 include/Event.h File Reference

Include dependency graph for Event.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Event](#)

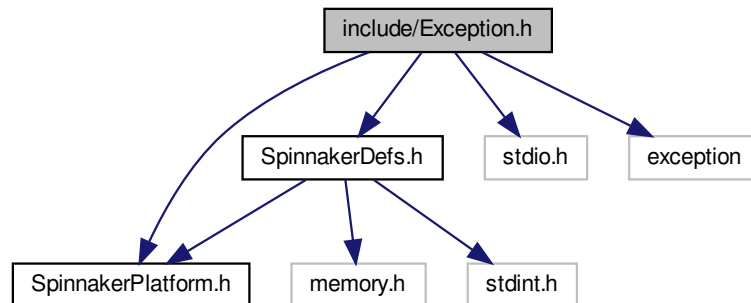
The base class for all event types.

Namespaces

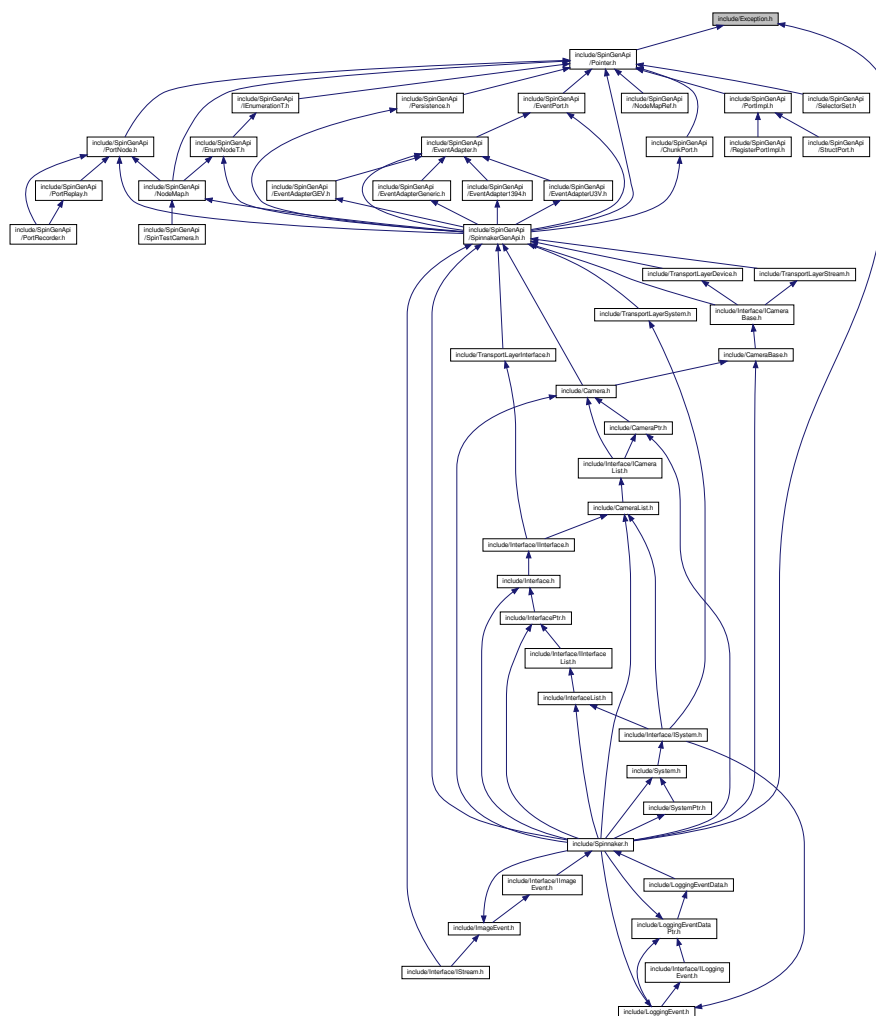
- [Spinnaker](#)

11.14 include/Exception.h File Reference

Include dependency graph for Exception.h:



This graph shows which files directly or indirectly include this file:

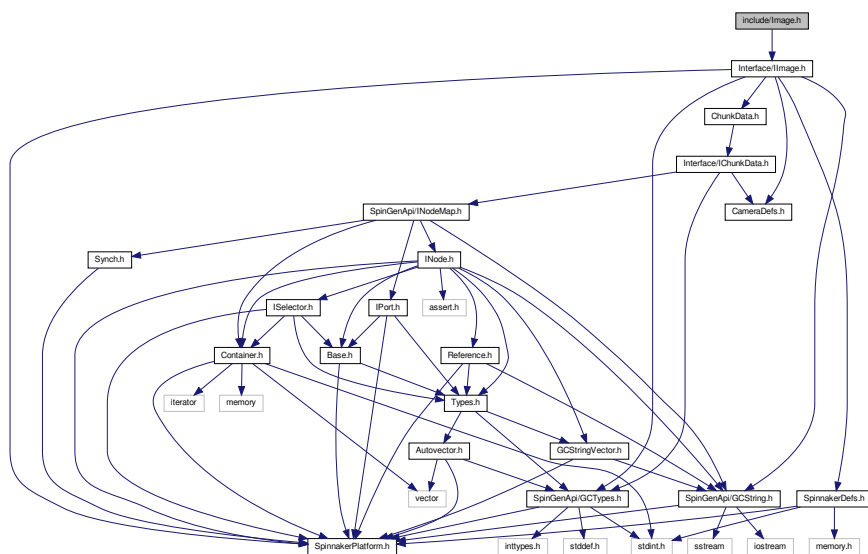


- class `Exception`

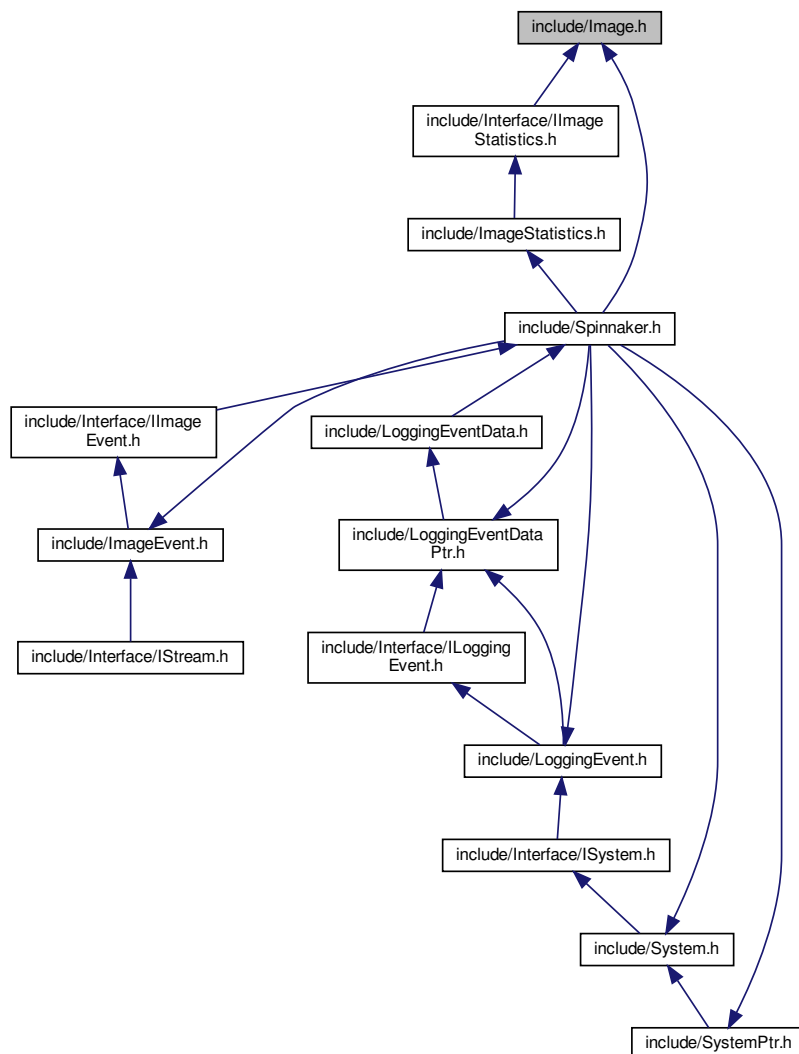
The *Exception* object represents an error that is returned from the library.

- Spinnaker

Include dependency graph for Image.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [Image](#)

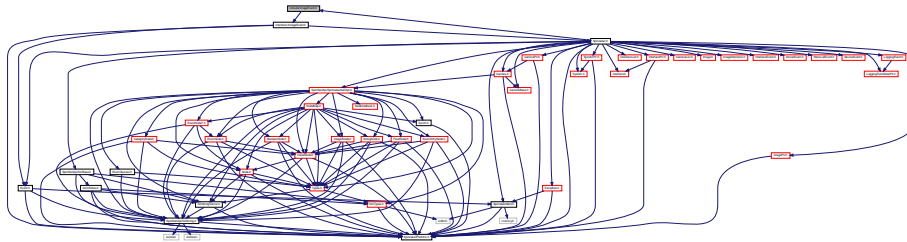
The image object class.

Namespaces

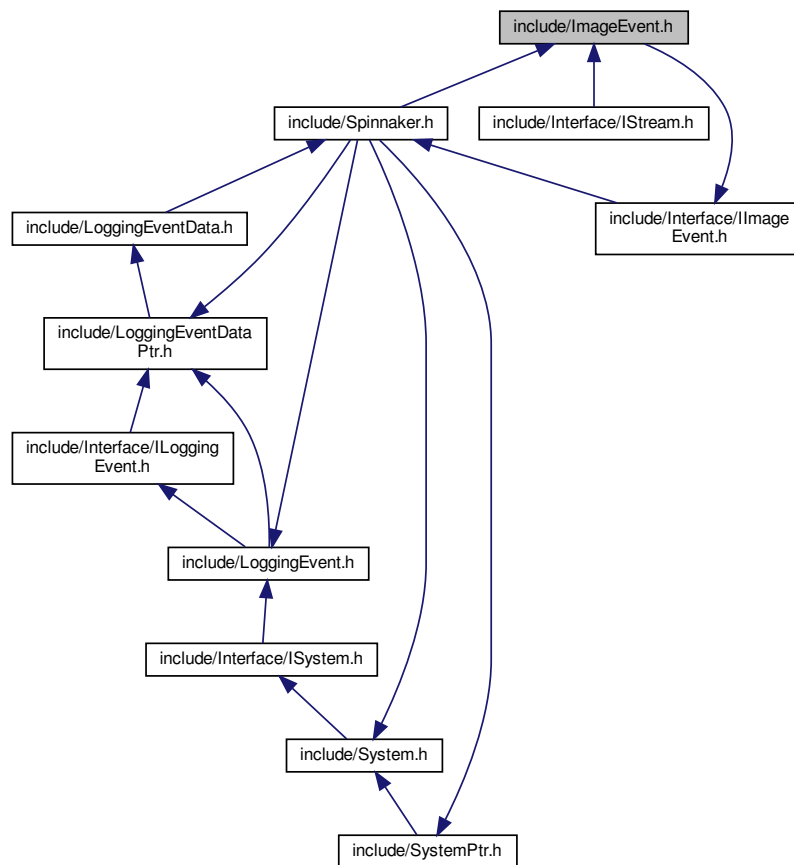
- [Spinnaker](#)

11.16 include/ImageEvent.h File Reference

Include dependency graph for ImageEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [ImageEvent](#)

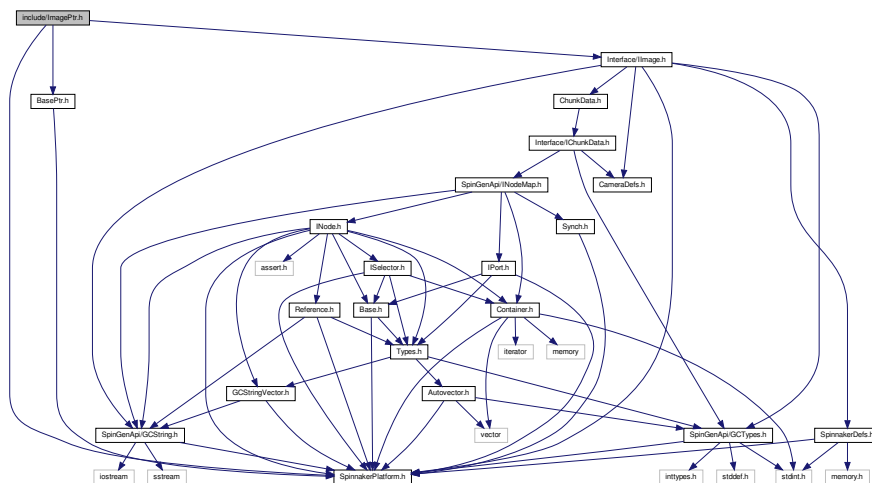
A handler for capturing image arrival events.

Namespaces

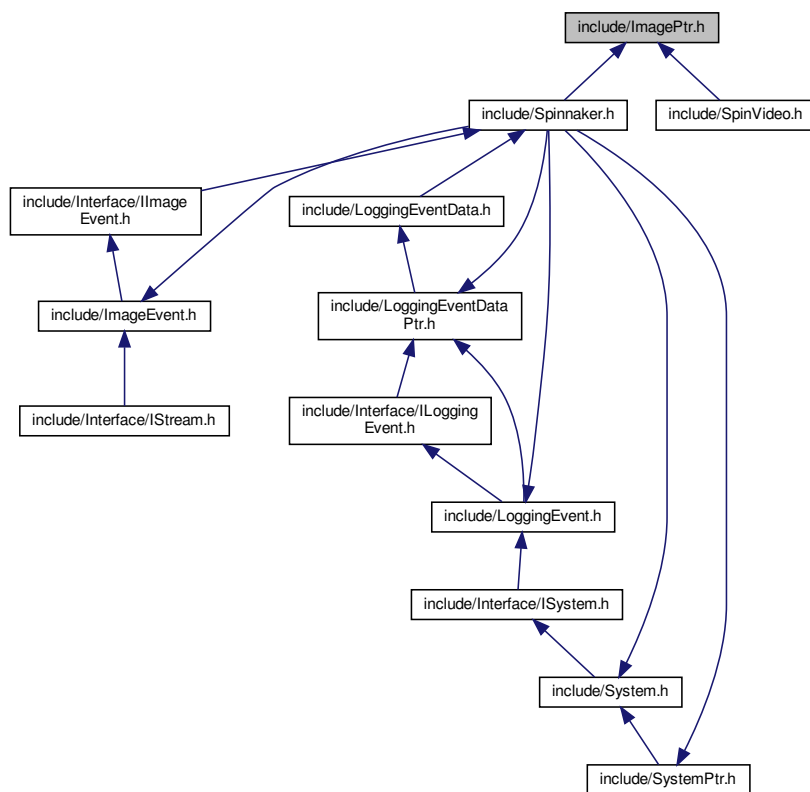
- [Spinnaker](#)

11.17 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](#)

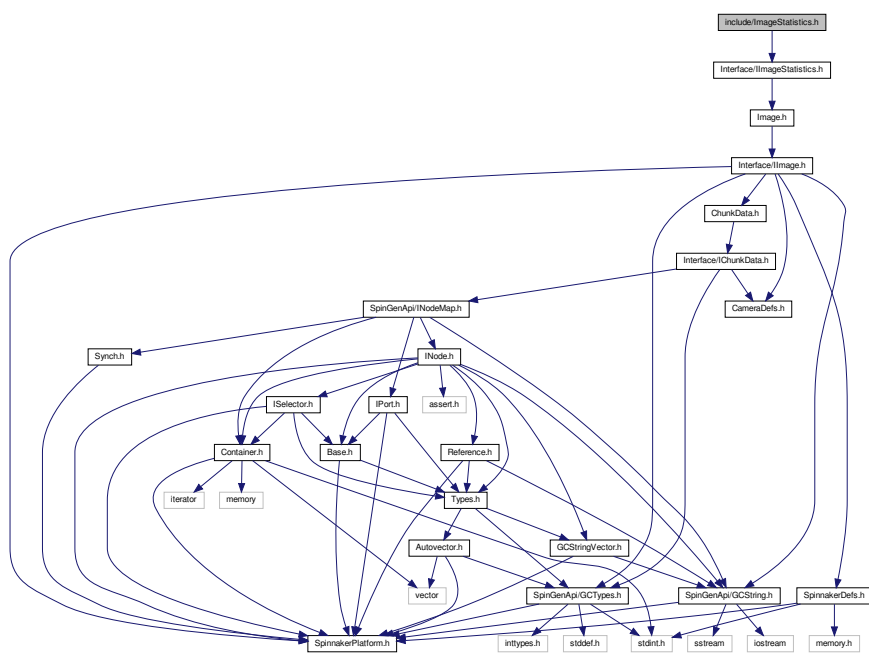
A reference tracked pointer to an image object.

Namespaces

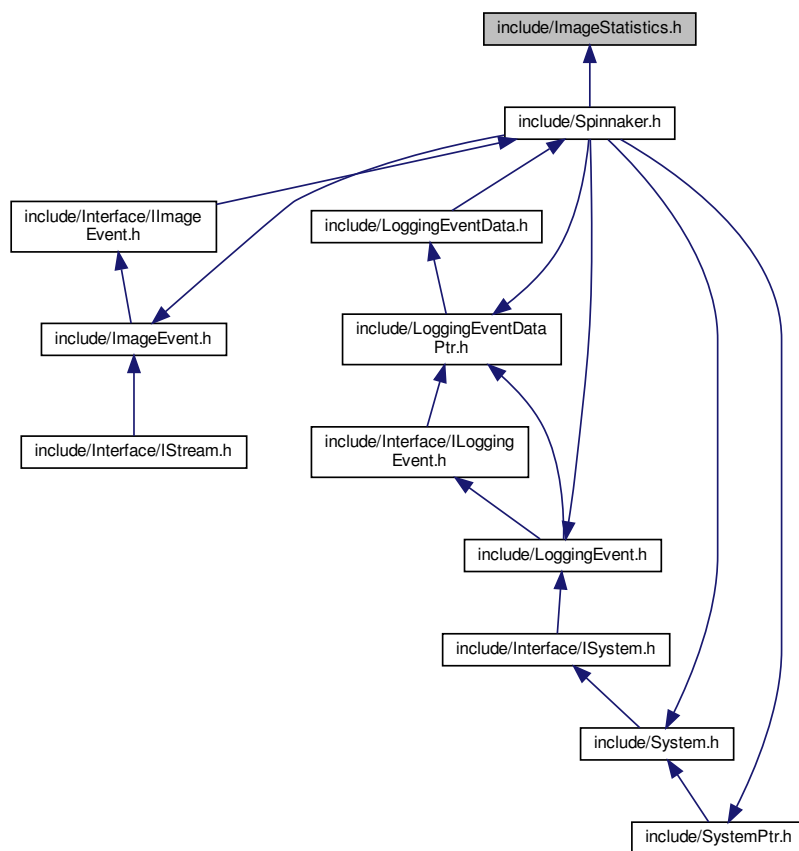
- [Spinnaker](#)

11.18 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](#)

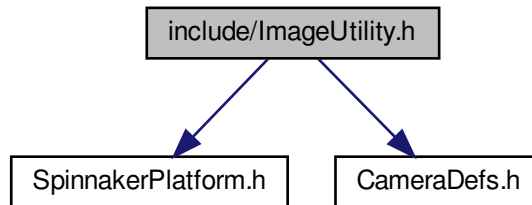
Represents image statistics for an image.

Namespaces

- [Spinnaker](#)

11.19 include/ImageUtility.h File Reference

Include dependency graph for ImageUtility.h:



Classes

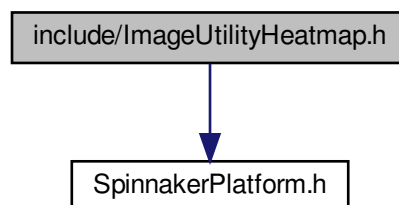
- class [ImageUtility](#)
Static helper functions for the image object class.

Namespaces

- [Spinnaker](#)

11.20 include/ImageUtilityHeatmap.h File Reference

Include dependency graph for ImageUtilityHeatmap.h:



Classes

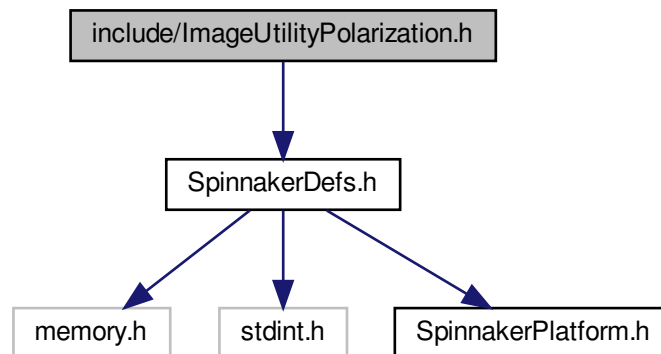
- class [ImageUtilityHeatmap](#)
Static functions to create heatmap images from image objects of pixel format Mono8 and Mono16.

Namespaces

- [Spinnaker](#)

11.21 include/ImageUtilityPolarization.h File Reference

Include dependency graph for ImageUtilityPolarization.h:



Classes

- class [ImageUtilityPolarization](#)

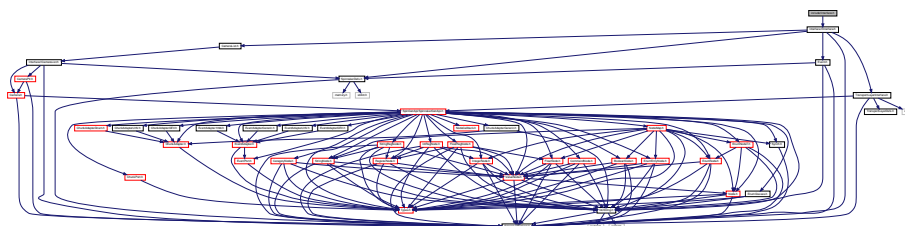
Static functions to create polarization images from image objects of pixel format `Polarized8` and `BayerRGPolarized8`.

Namespaces

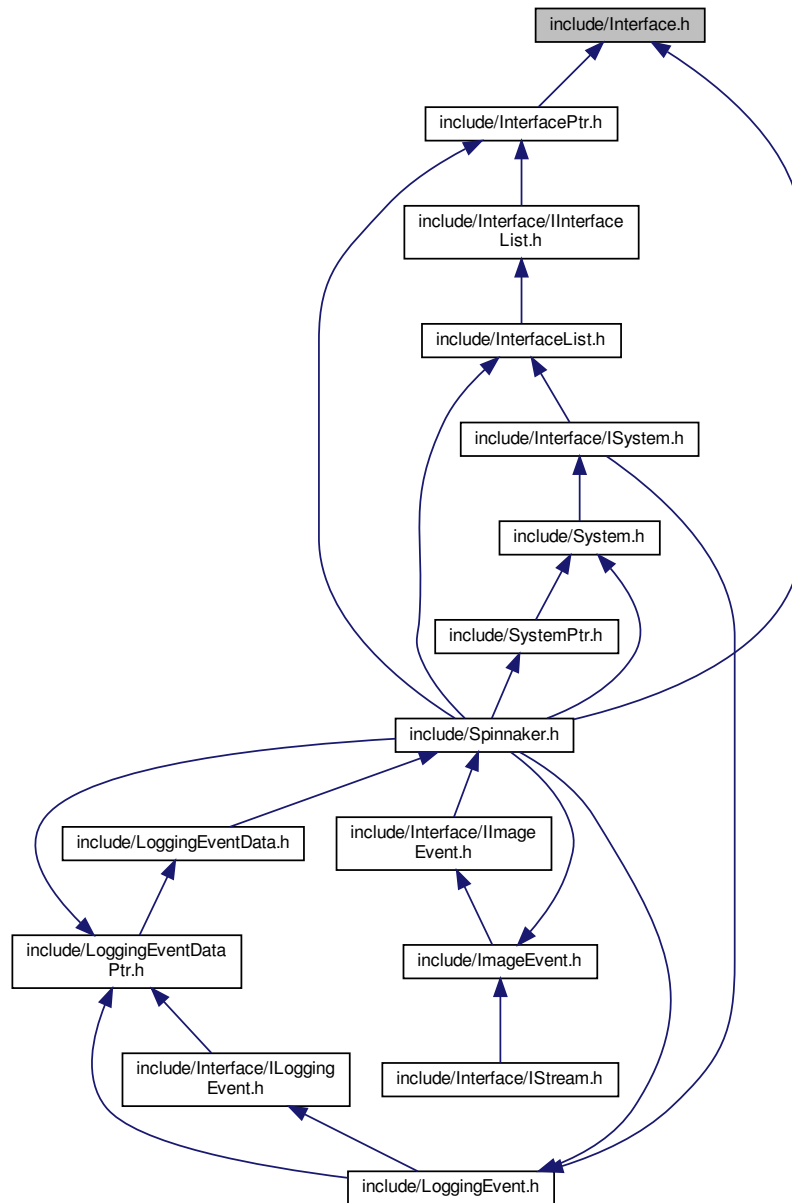
- [Spinnaker](#)

11.22 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](#)

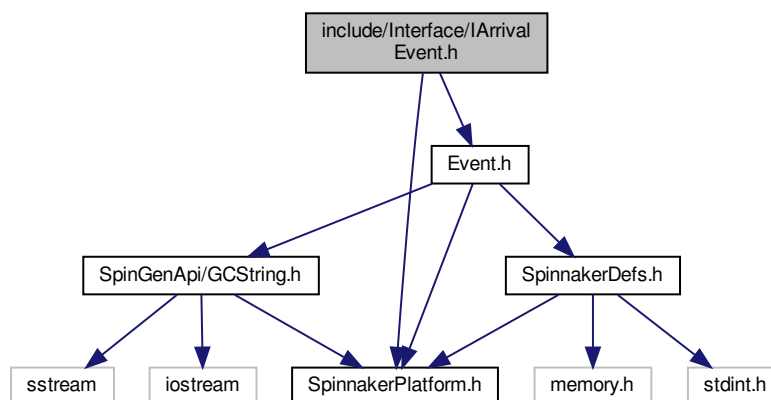
An interface object which holds a list of cameras.

Namespaces

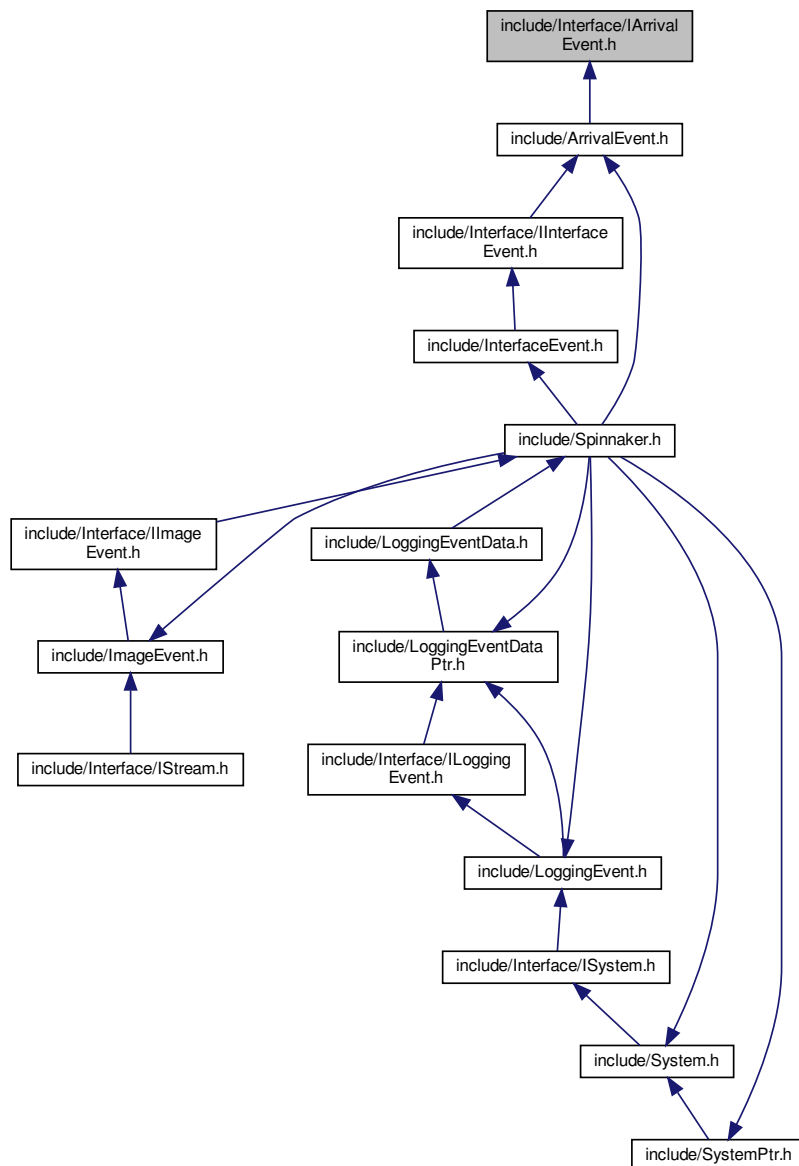
- [Spinnaker](#)

11.23 include/Interface/IArrivalEvent.h File Reference

Include dependency graph for IArrivalEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

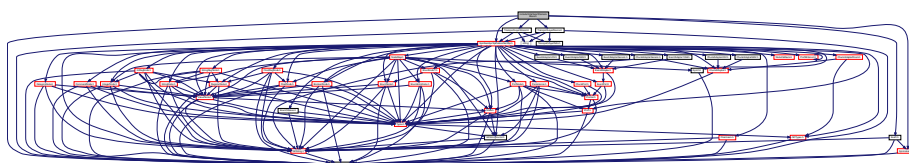
- class [IArrivalEvent](#)

Namespaces

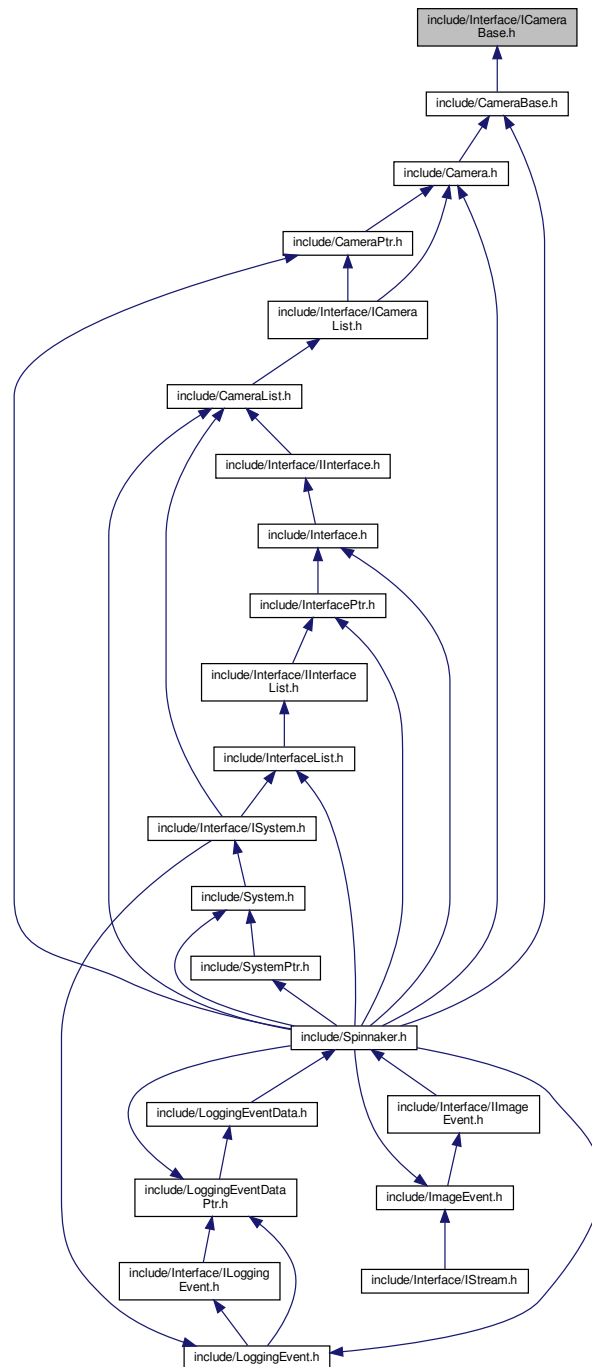
- [Spinnaker](#)

11.24 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](#)

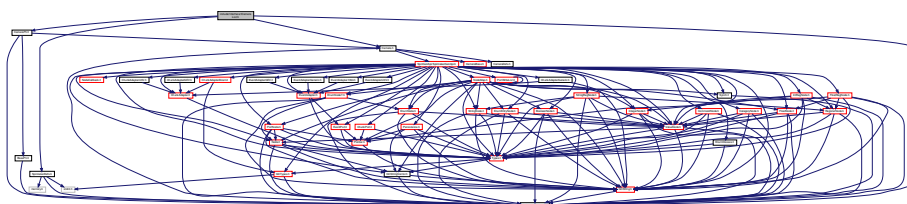
The interface file for base class for the camera object.

Namespaces

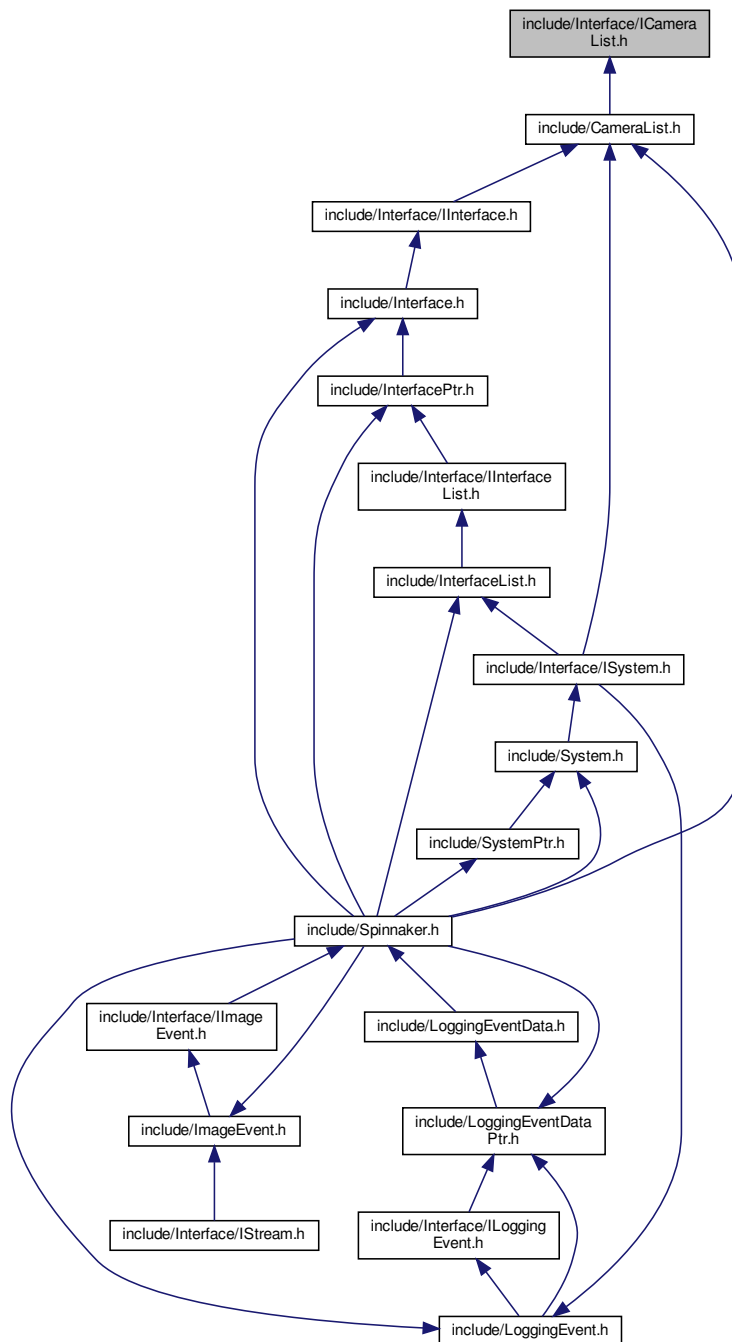
- [Spinnaker](#)

11.25 include/Interface/ICameraList.h File Reference

Include dependency graph for ICameraList.h:



This graph shows which files directly or indirectly include this file:



Classes

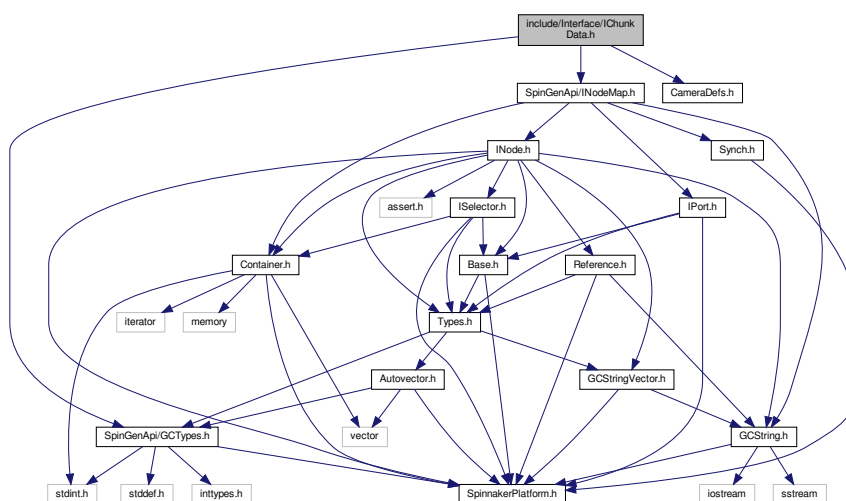
- class [ICameraList](#)
Used to hold a list of camera objects.

Namespaces

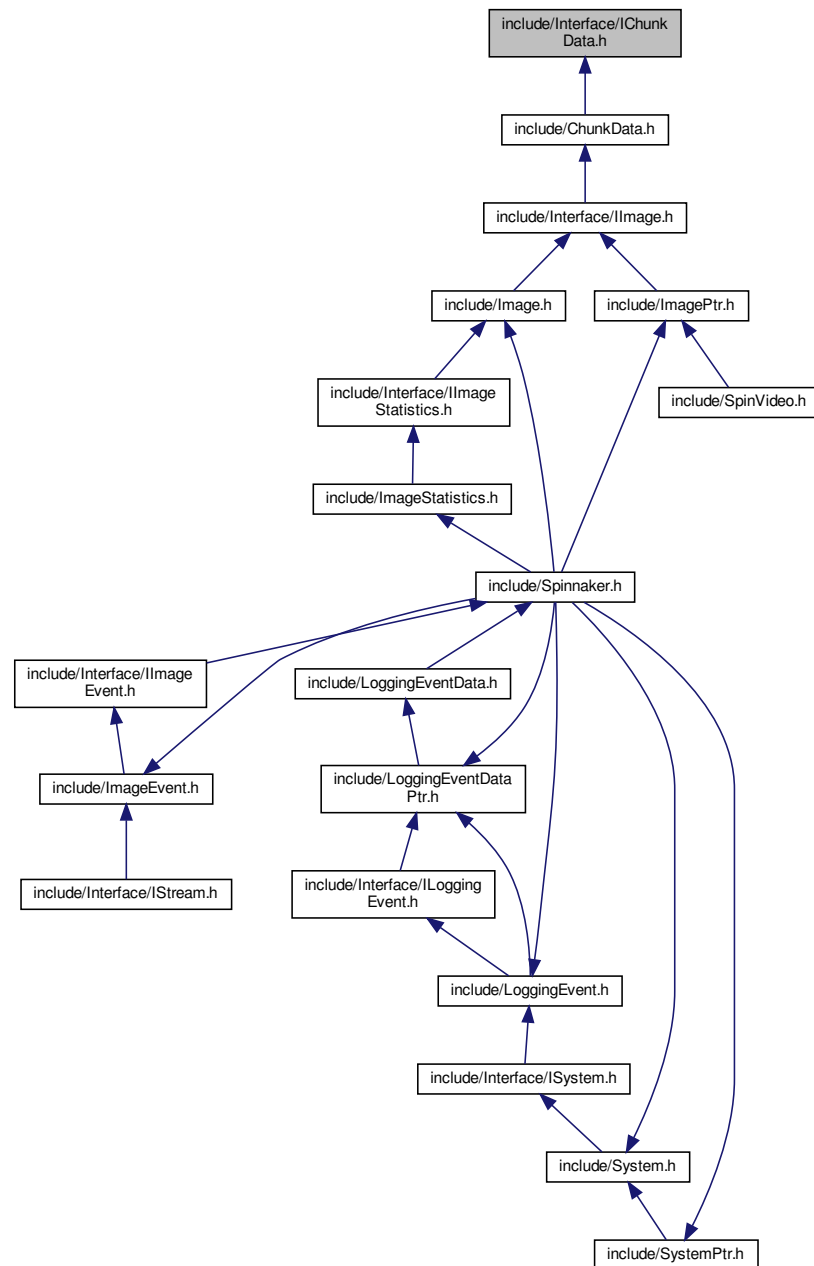
- [Spinnaker](#)

11.26 include/Interface/IChunkData.h File Reference

Include dependency graph for IChunkData.h:



This graph shows which files directly or indirectly include this file:



Classes

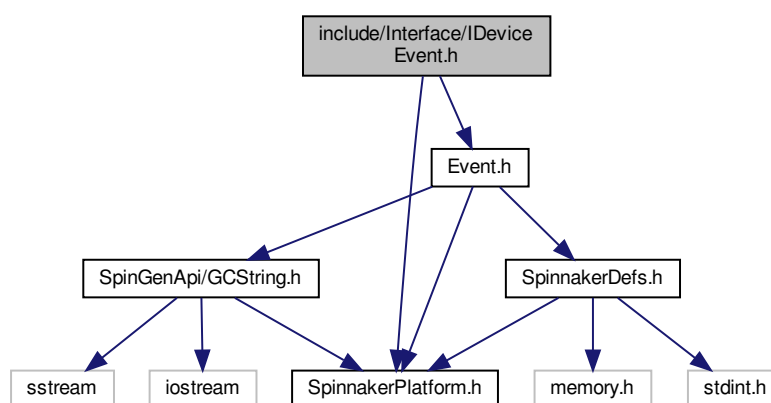
- class [IChunkData](#)
The *Interface* file for *ChunkData*.

Namespaces

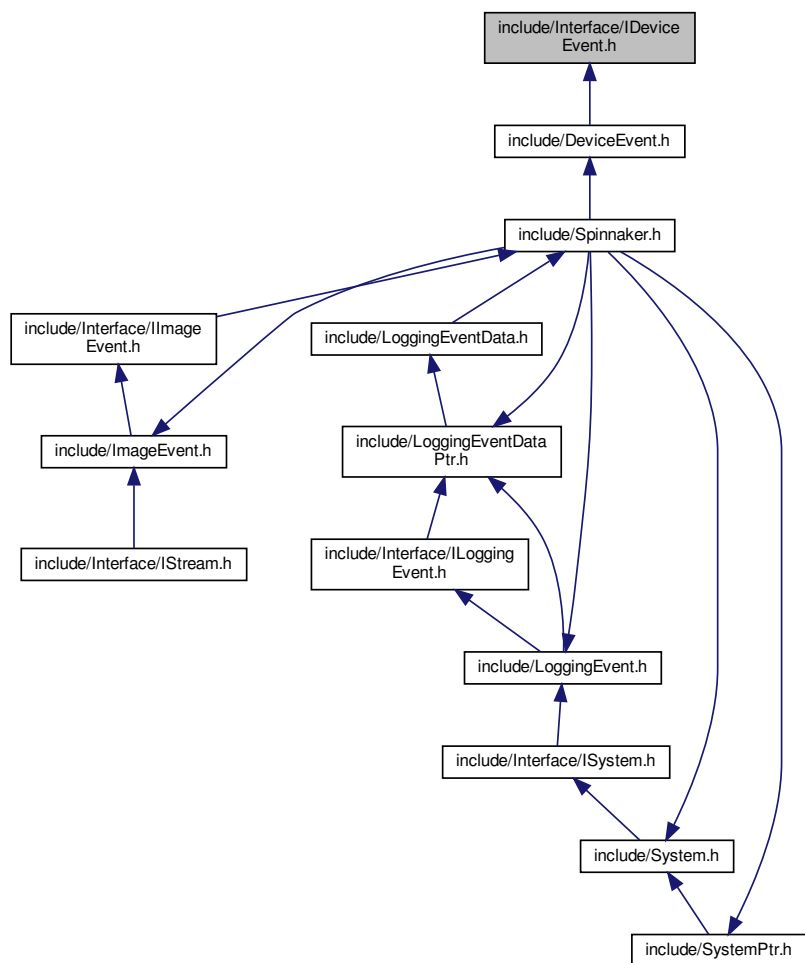
- [Spinnaker](#)

11.27 include/Interface/IDeviceEvent.h File Reference

Include dependency graph for IDeviceEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

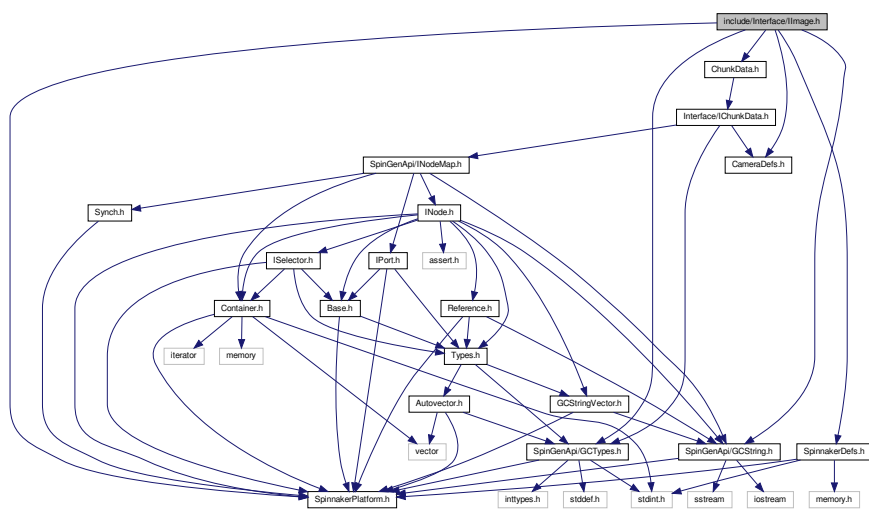
- class [IDeviceEvent](#)

Namespaces

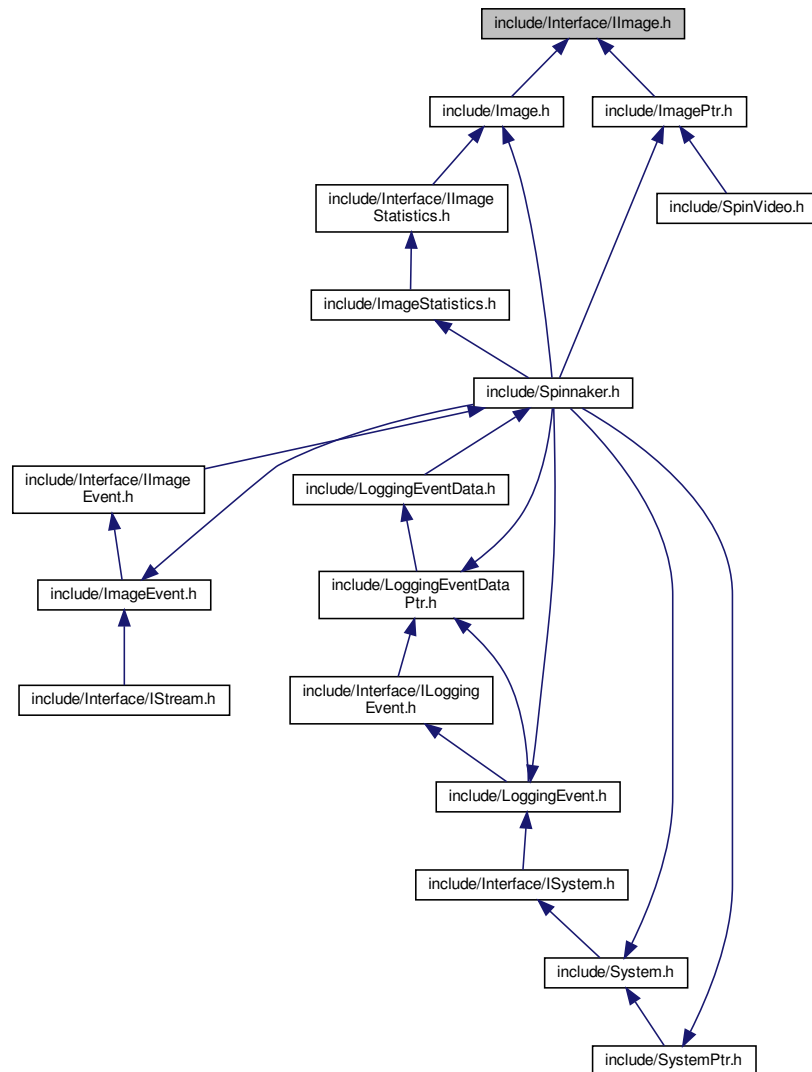
- [Spinnaker](#)

11.28 include/Interface/IImage.h File Reference

Include dependency graph for Image.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IImage](#)

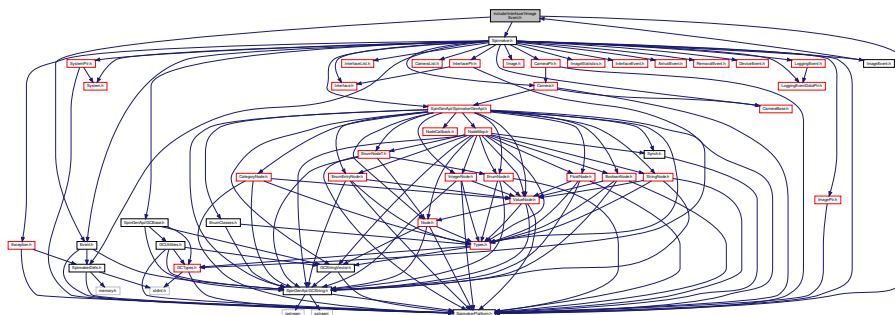
The interface file for [Image](#).

Namespaces

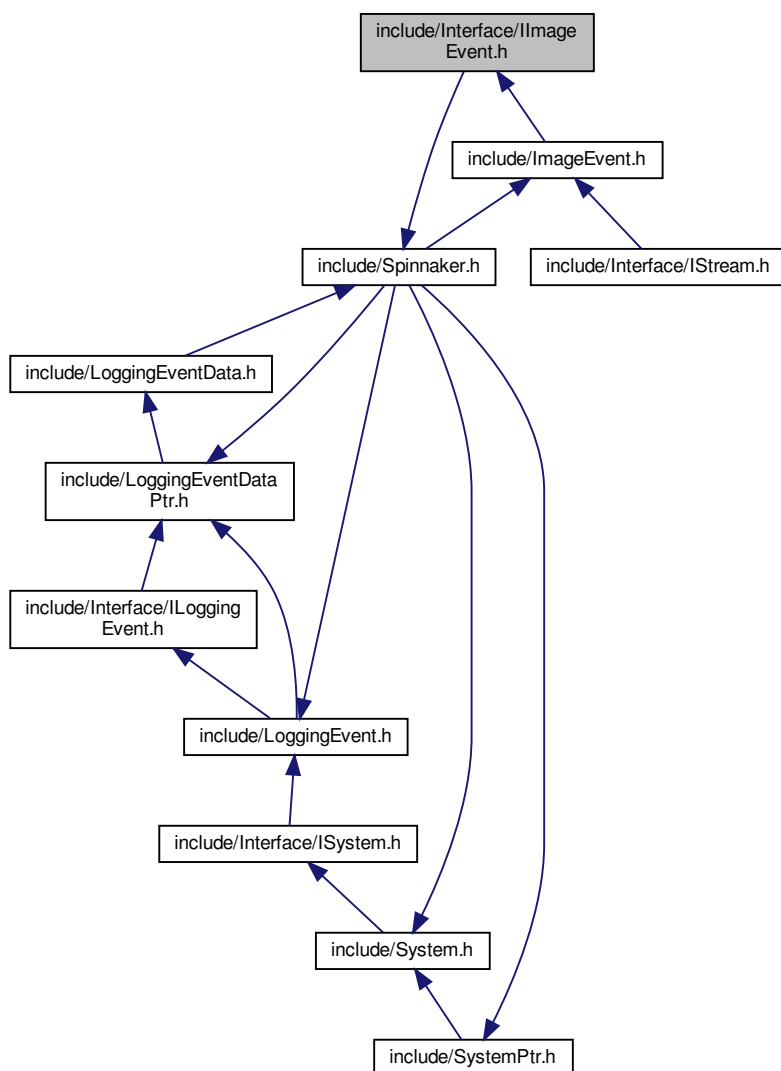
- [Spinnaker](#)

11.29 include/Interface/IImageEvent.h File Reference

Include dependency graph for IImageEvent.h:



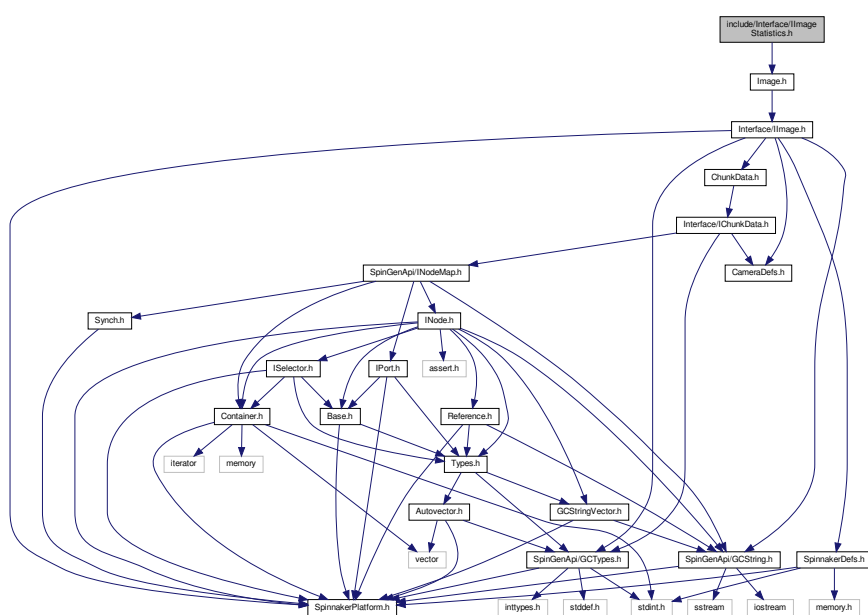
This graph shows which files directly or indirectly include this file:



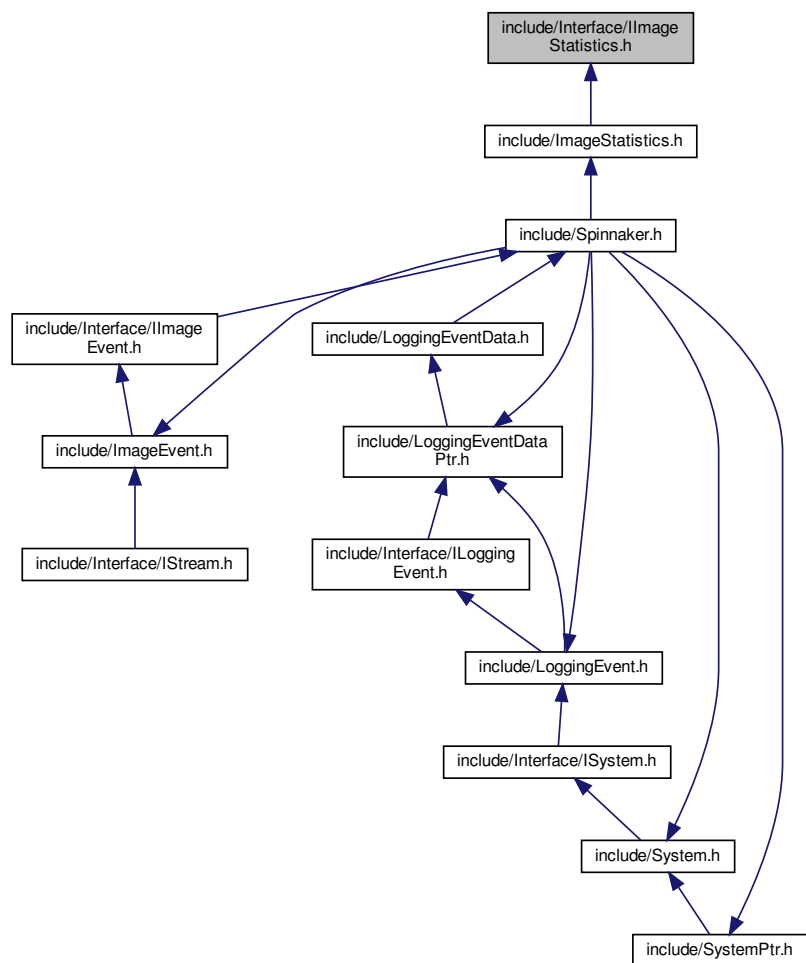
- class ImageEvent

- Spinnaker

Include dependency graph for ImageStatistics.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IImageStatistics](#)

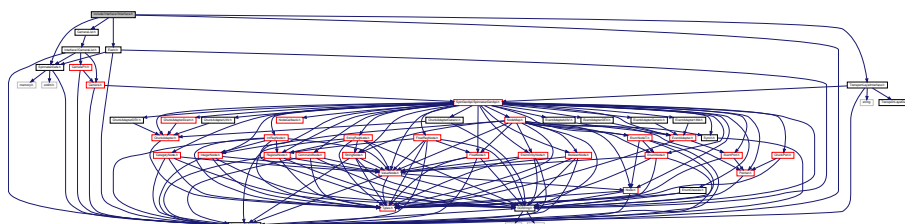
The interface file for image statistics.

Namespaces

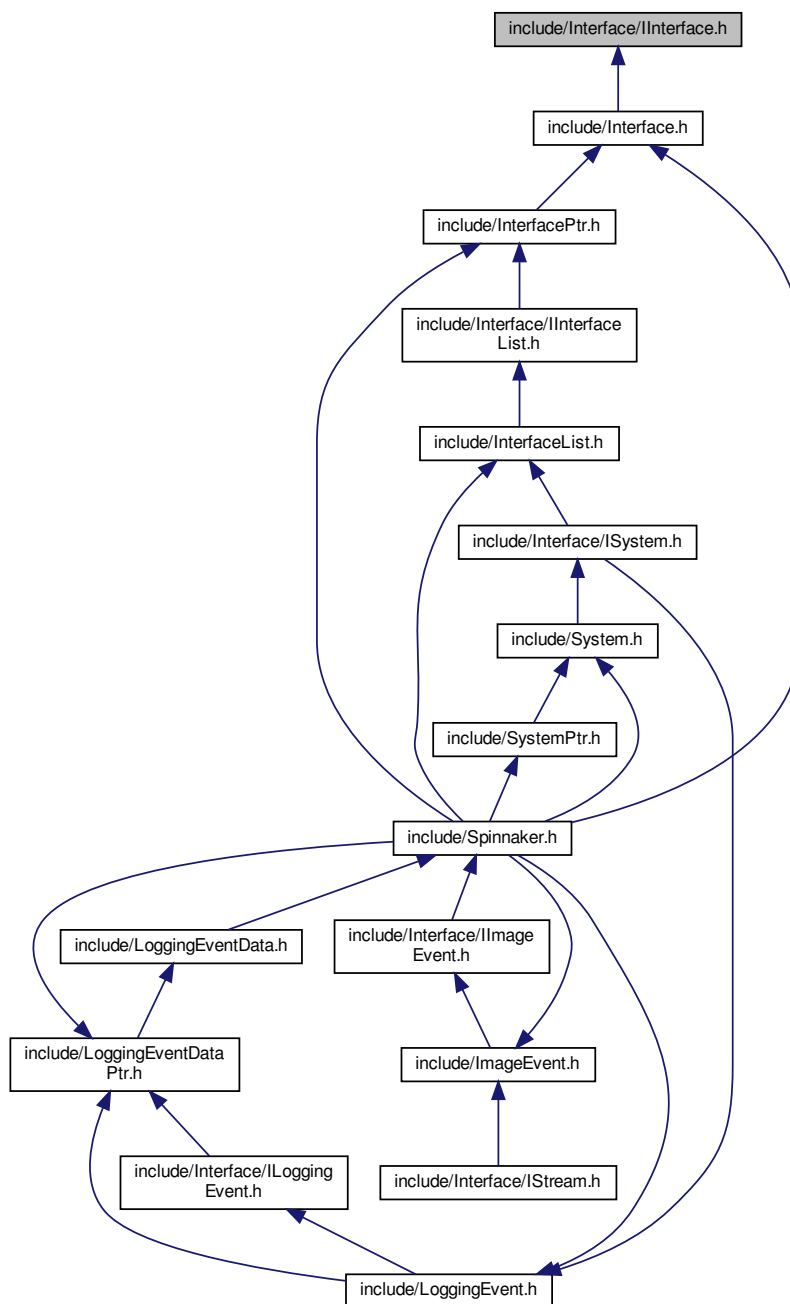
- [Spinnaker](#)

11.31 include/Interface/Interface.h File Reference

Include dependency graph for IInterface.h:



This graph shows which files directly or indirectly include this file:



Classes

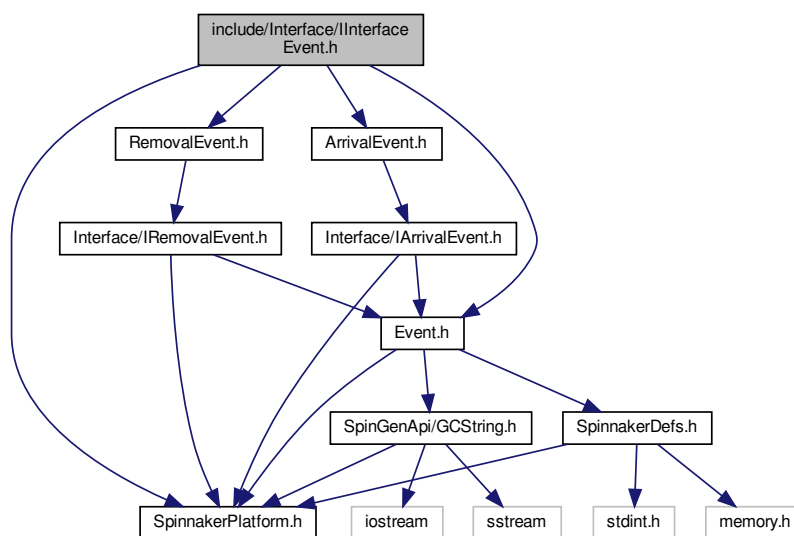
- class [IInterface](#)
The interface file for [IInterface](#).

Namespaces

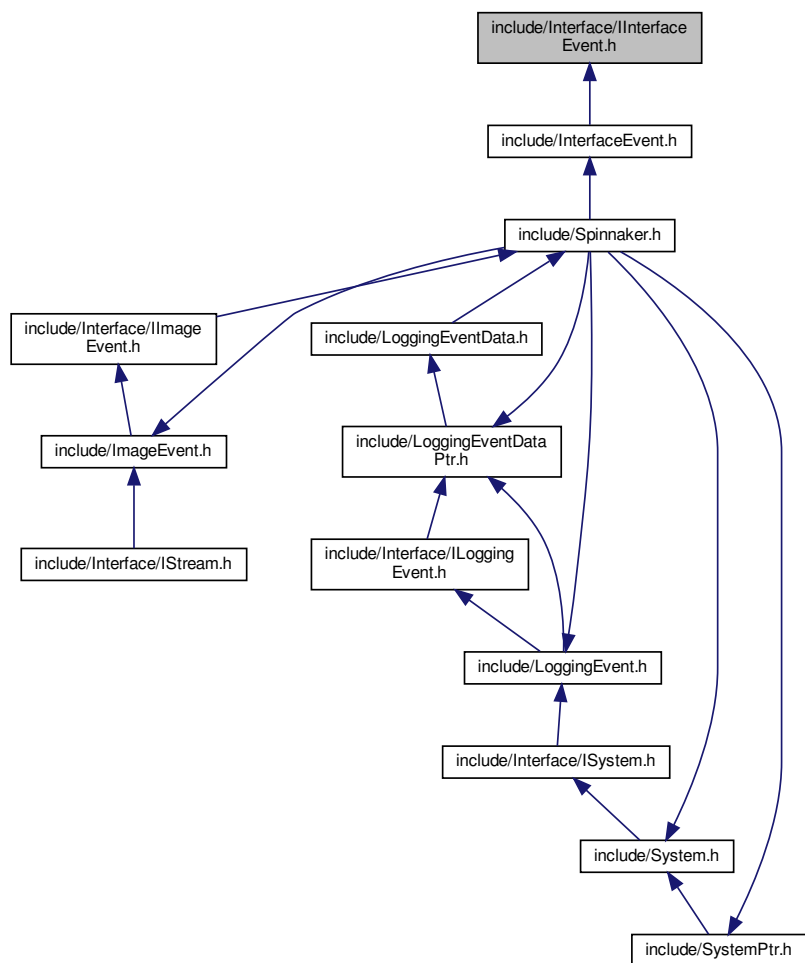
- [Spinnaker](#)

11.32 include/Interface/InterfaceEvent.h File Reference

Include dependency graph for InterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

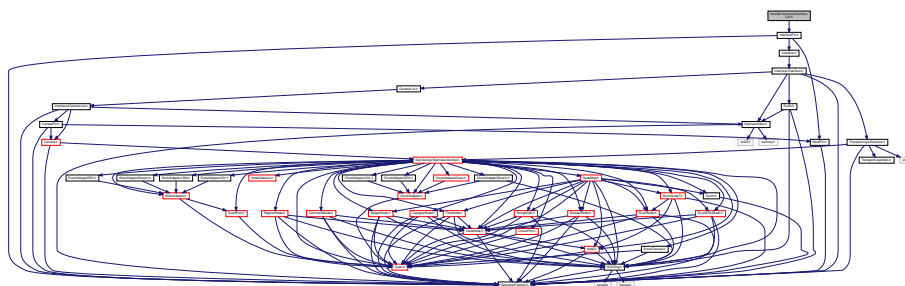
- class `InterfaceEvent`

Namespaces

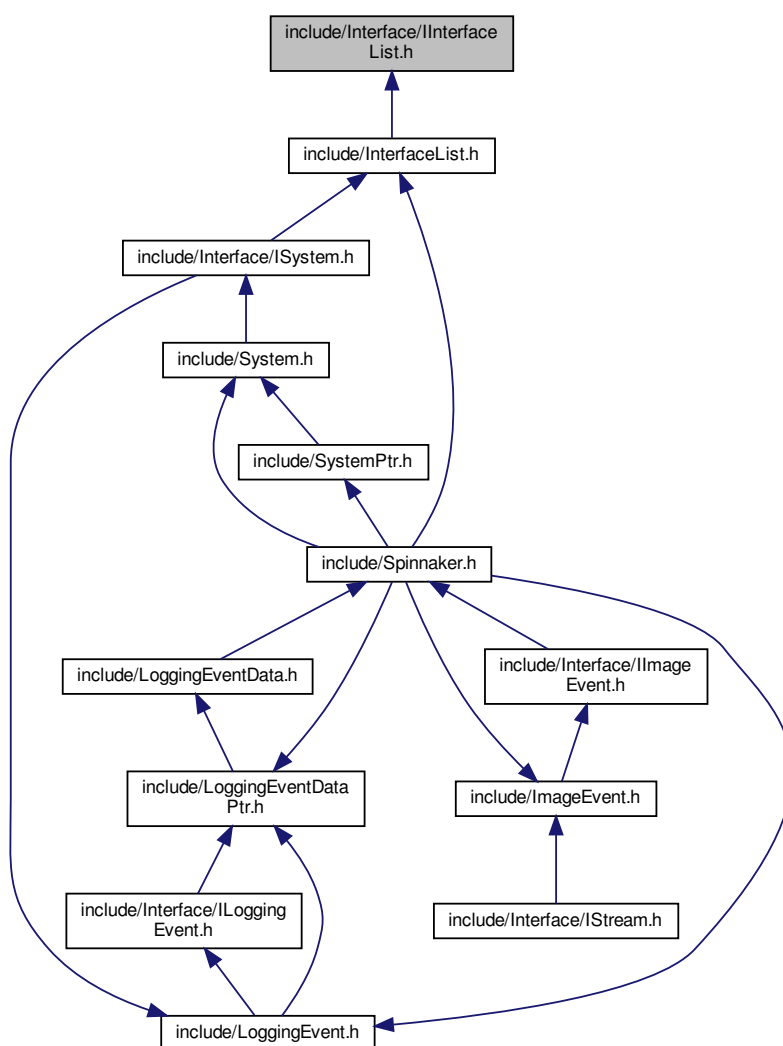
- Spinnaker

11.33 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 [ILogInterfaceList](#)

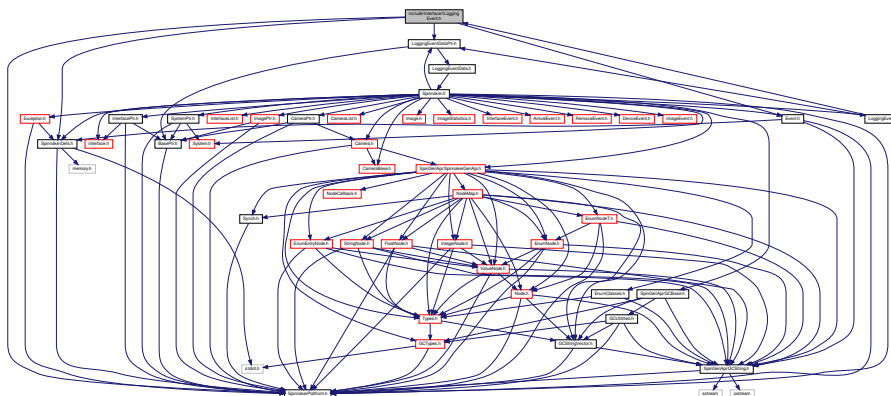
The interface file for [ILogInterfaceList](#) class.

Namespaces

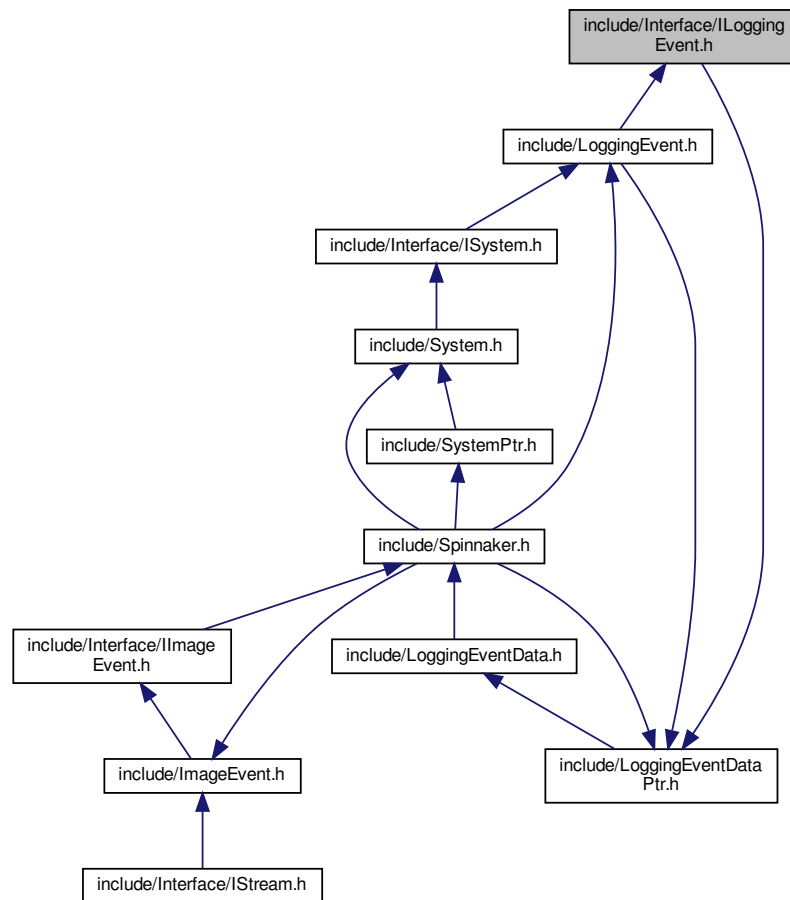
- [Spinnaker](#)

11.34 include/Interface/ILoggingEvent.h File Reference

Include dependency graph for ILoggingEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

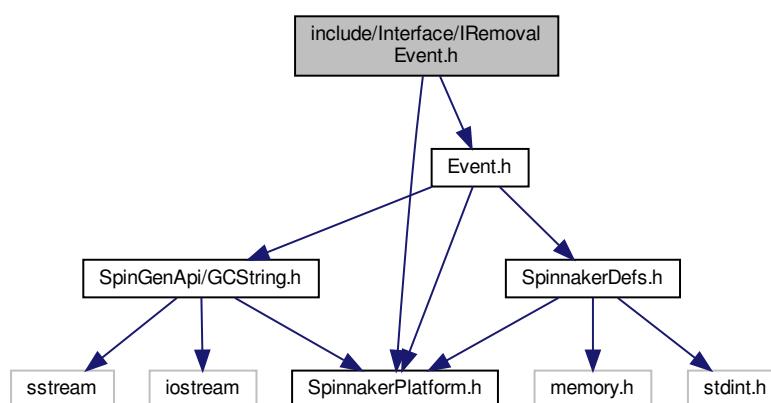
- class [ILoggingEvent](#)

Namespaces

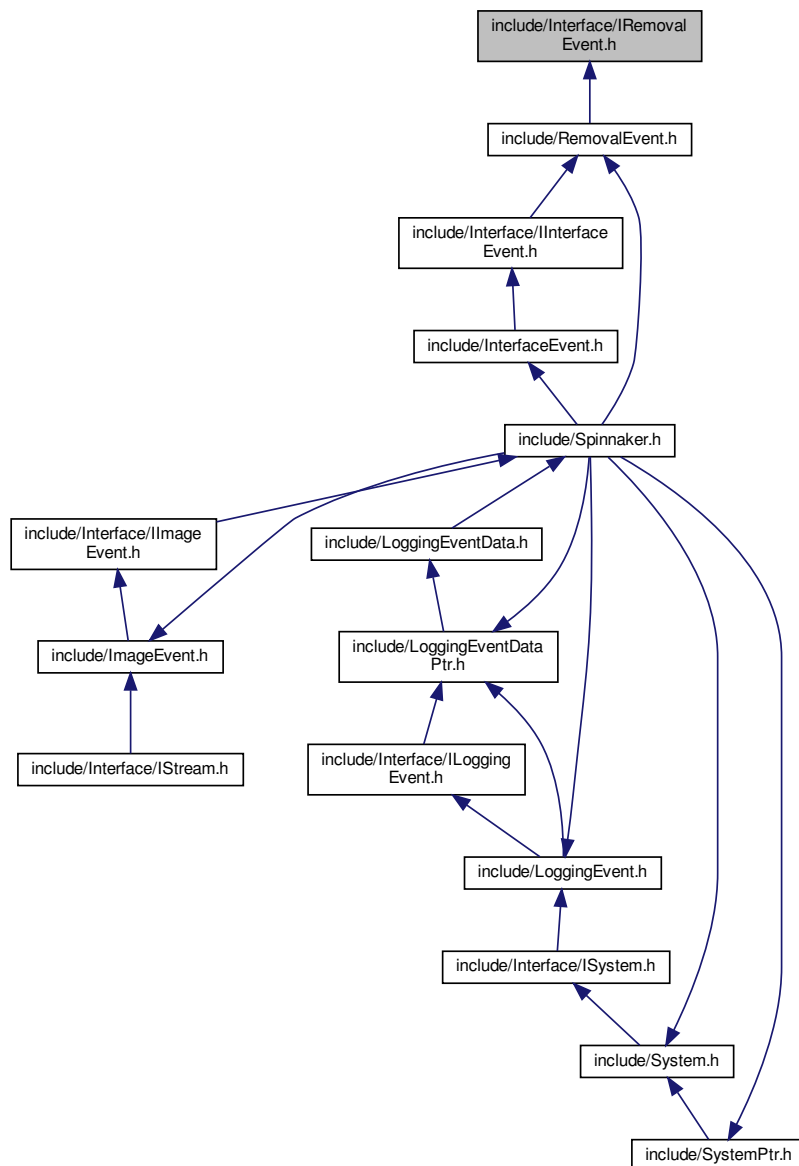
- [Spinnaker](#)

11.35 include/Interface/IRemovalEvent.h File Reference

Include dependency graph for IRemovalEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

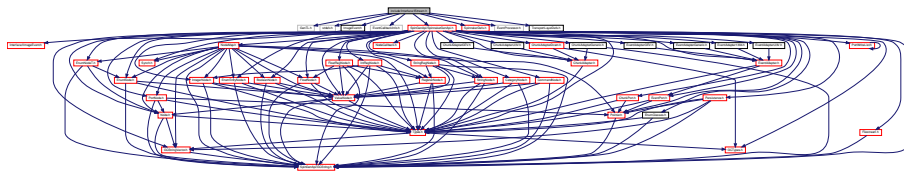
- class [IRemovalEvent](#)

Namespaces

- [Spinnaker](#)

11.36 include/Interface/IStream.h File Reference

Include dependency graph for IStream.h:



Classes

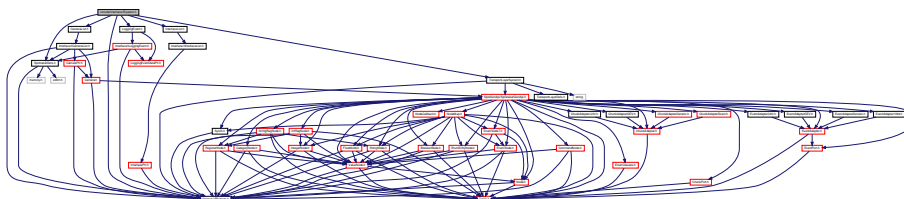
- class [IDataStream](#)

Namespaces

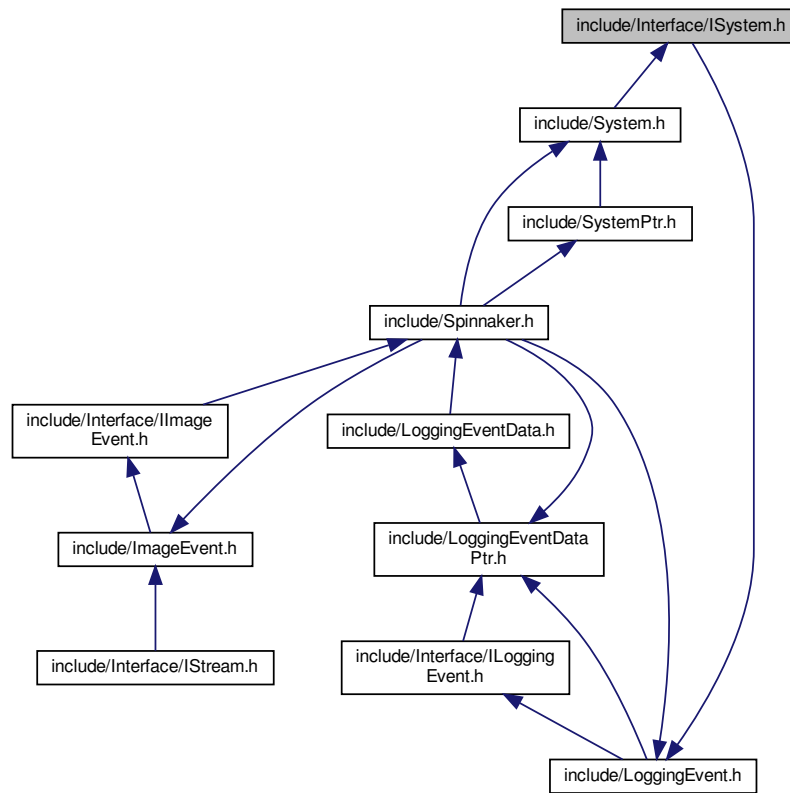
- [Spinnaker](#)

11.37 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](#)

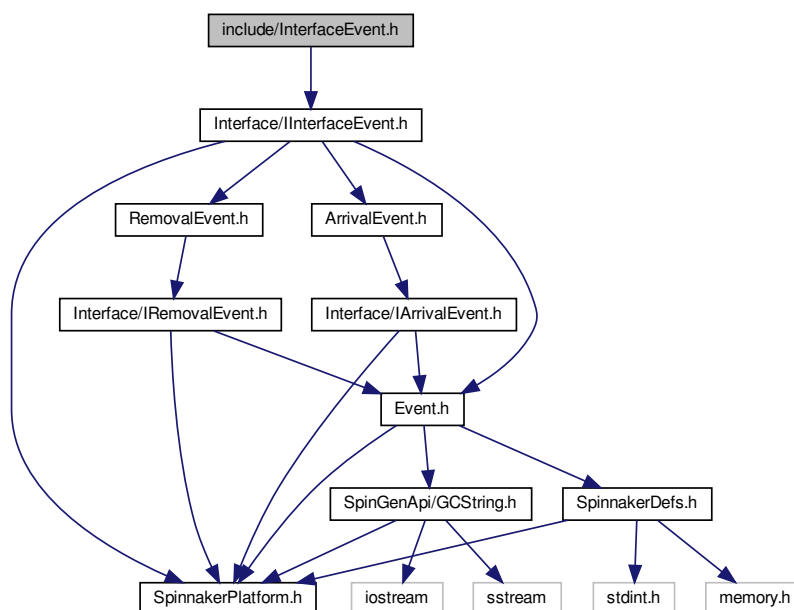
The interface file for [System](#).

Namespaces

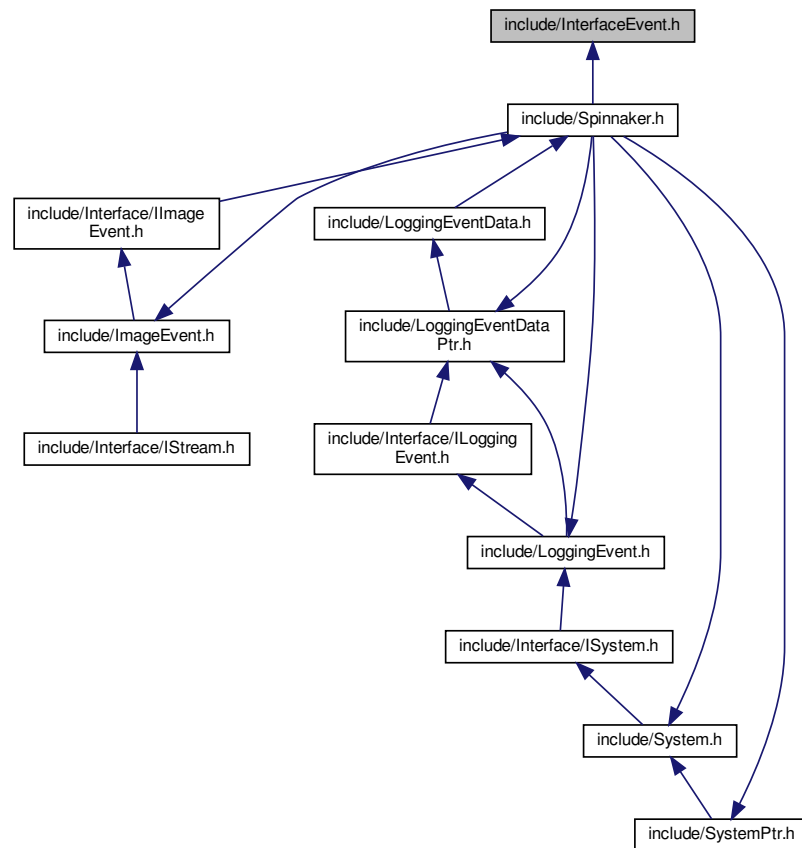
- [Spinnaker](#)

11.38 include/InterfaceEvent.h File Reference

Include dependency graph for InterfaceEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [InterfaceEvent](#)

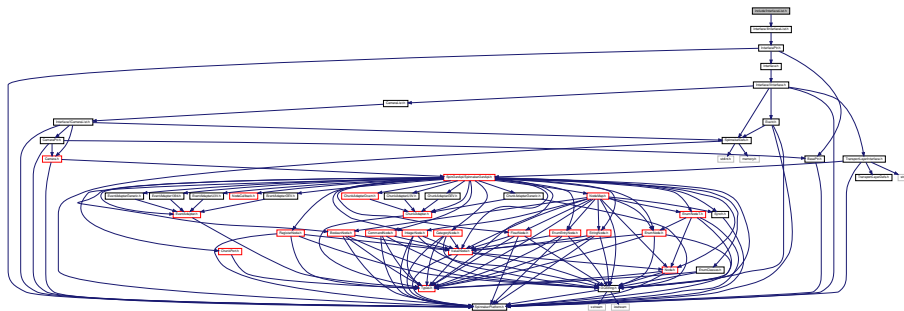
A handler to device arrival and removal events on all interfaces.

Namespaces

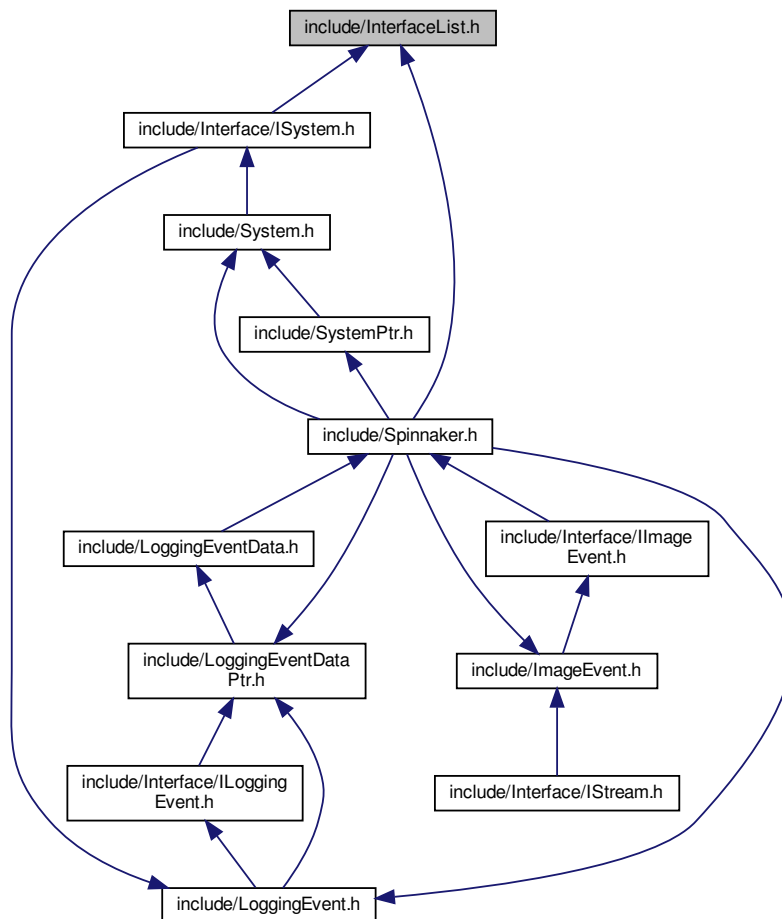
- [Spinnaker](#)

11.39 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](#)

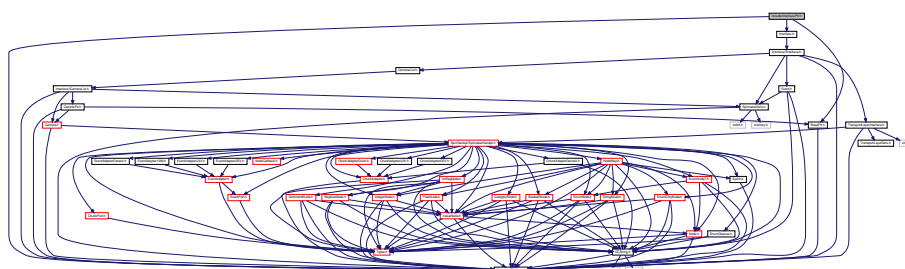
A list of the available interfaces on the system.

Namespaces

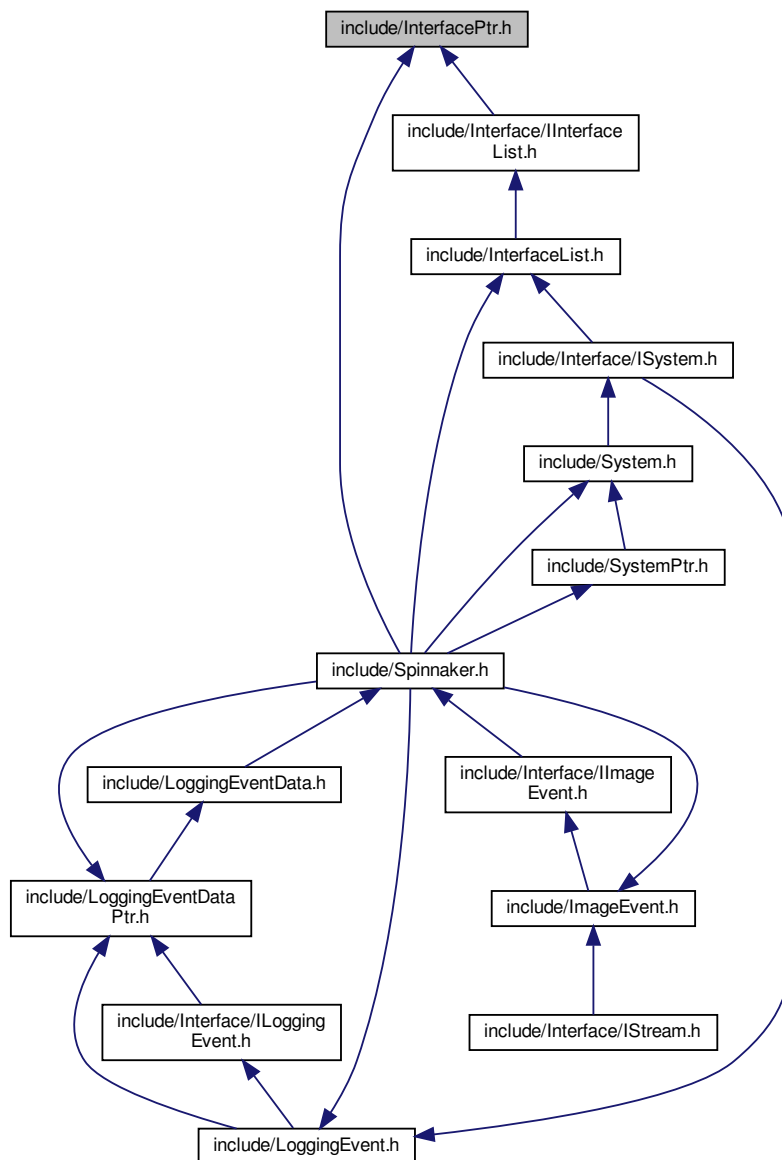
- [Spinnaker](#)

11.40 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](#)

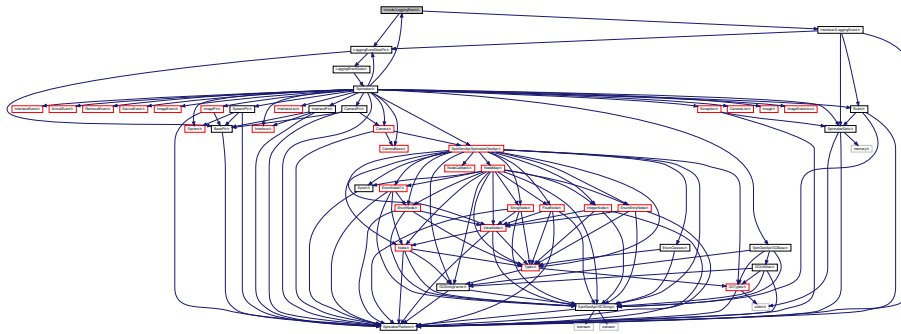
A reference tracked pointer to the interface object.

Namespaces

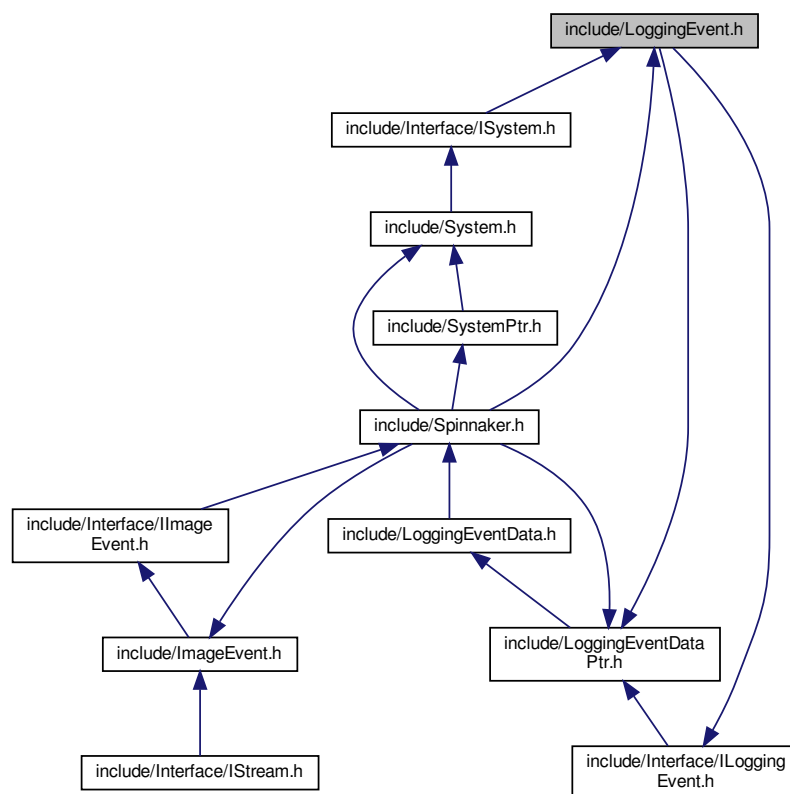
- [Spinnaker](#)

11.41 include/LoggingEvent.h File Reference

Include dependency graph for LoggingEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [LoggingEvent](#)

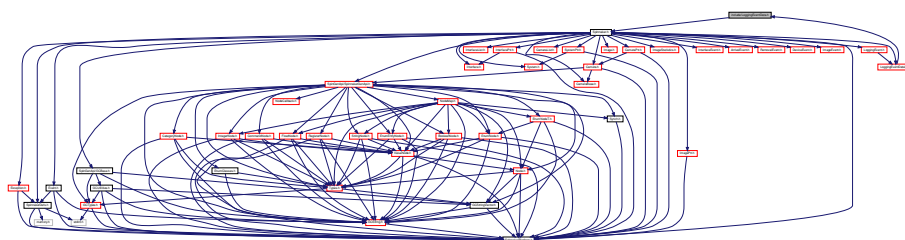
An event handler for capturing the device logging event.

Namespaces

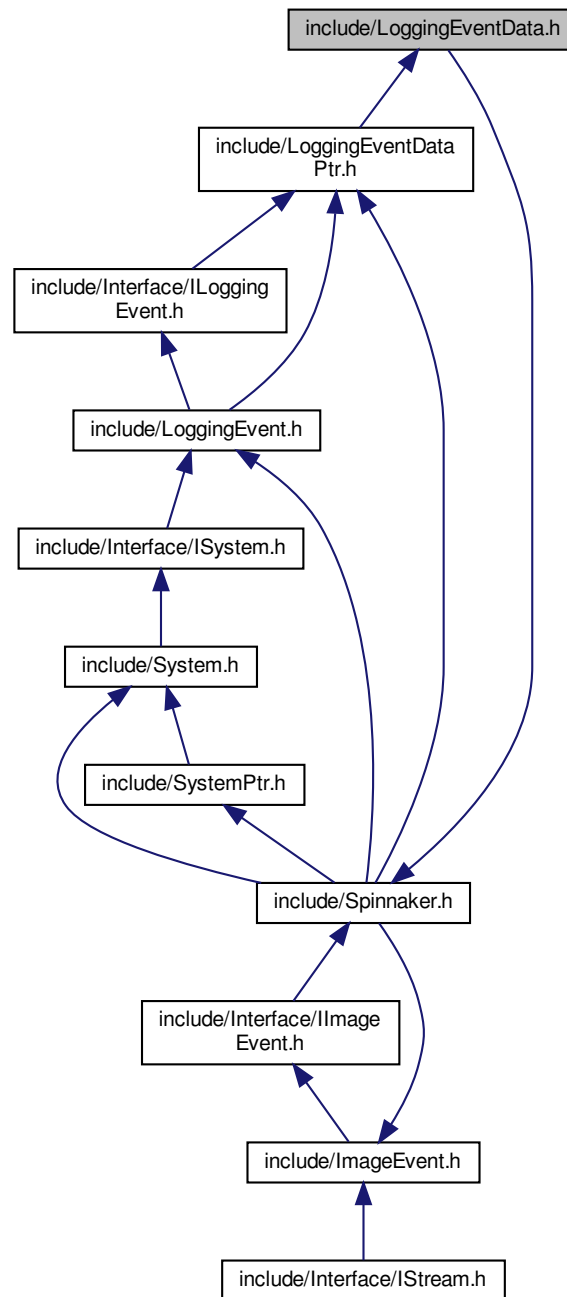
- [Spinnaker](#)

11.42 include/LoggingEventData.h File Reference

Include dependency graph for LoggingEventData.h:



This graph shows which files directly or indirectly include this file:



Classes

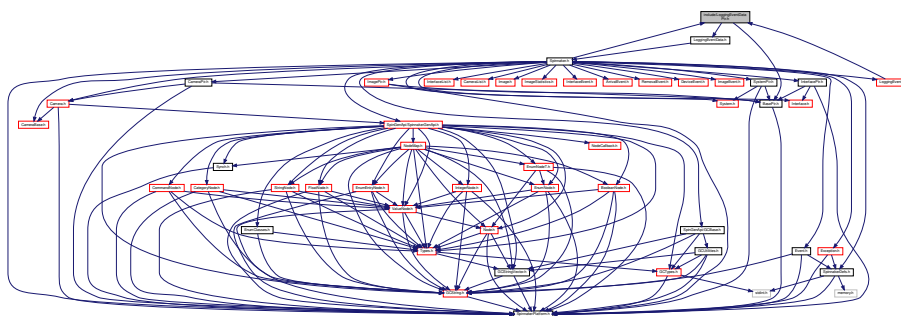
- class [LoggingEventData](#)
The *LoggingEventData* object.

Namespaces

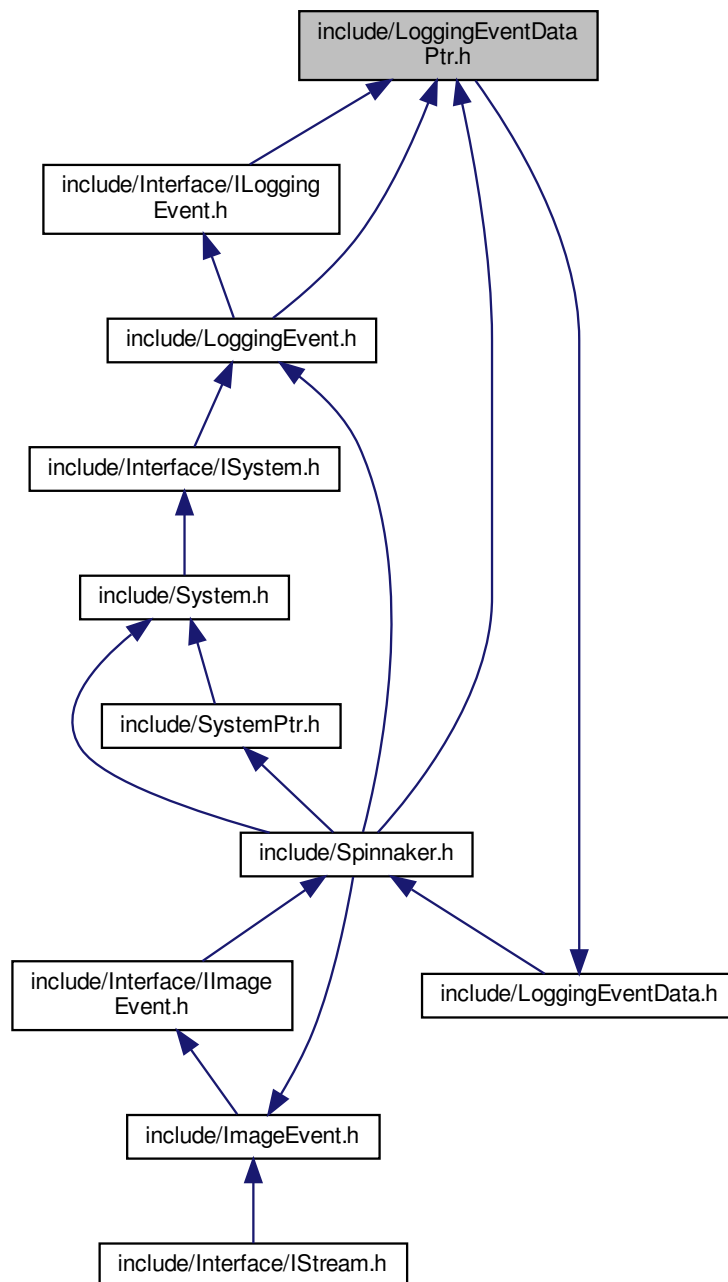
- [Spinnaker](#)

11.43 include/LoggingEventDataPtr.h File Reference

Include dependency graph for LoggingEventDataPtr.h:



This graph shows which files directly or indirectly include this file:



Classes

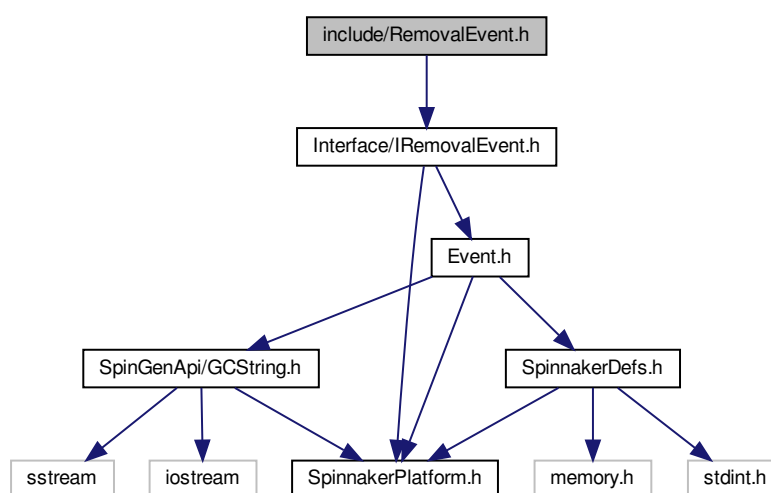
- class [LoggingEventDataPtr](#)
A reference tracked pointer to the [LoggingEvent](#) object.

Namespaces

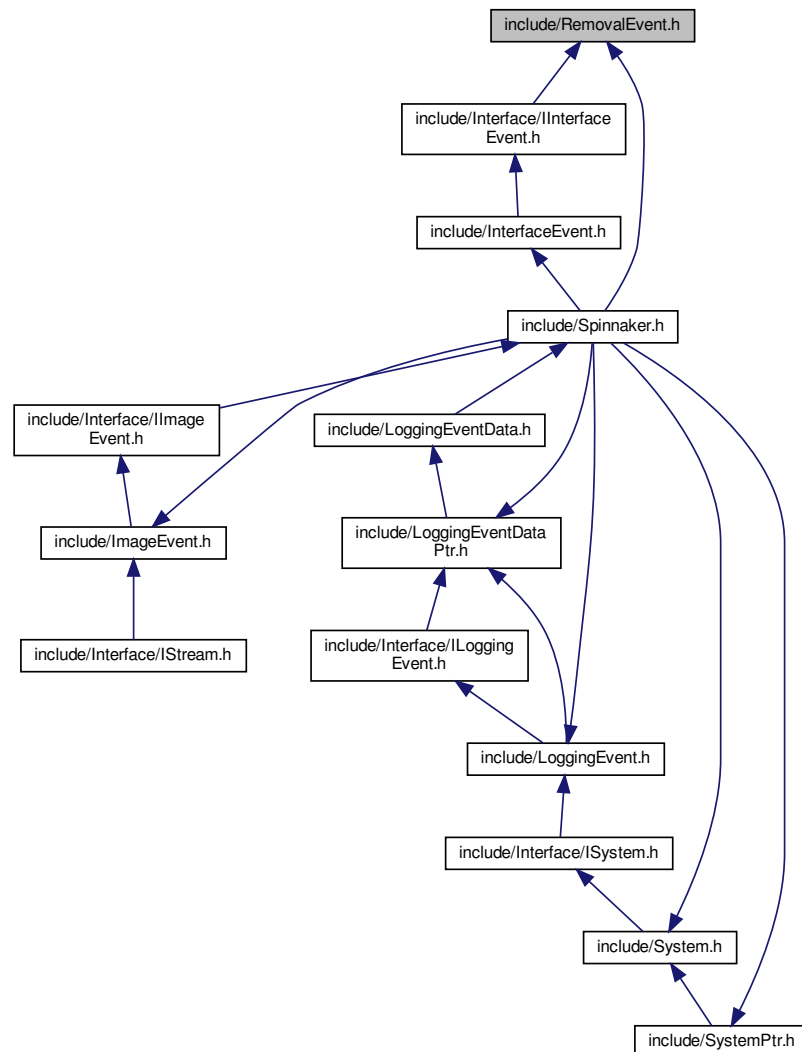
- [Spinnaker](#)

11.44 include/RemovalEvent.h File Reference

Include dependency graph for RemovalEvent.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [RemovalEvent](#)

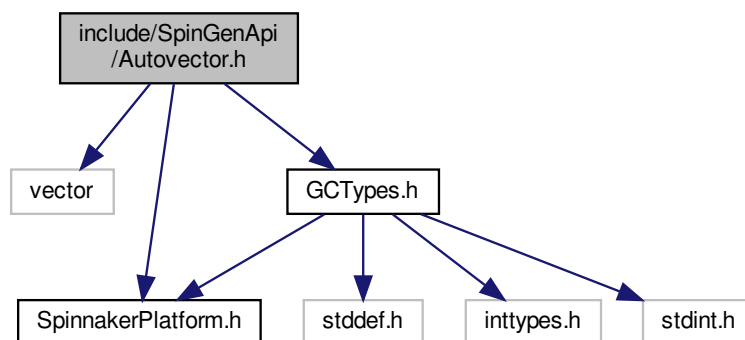
An event handler for capturing the device removal event.

Namespaces

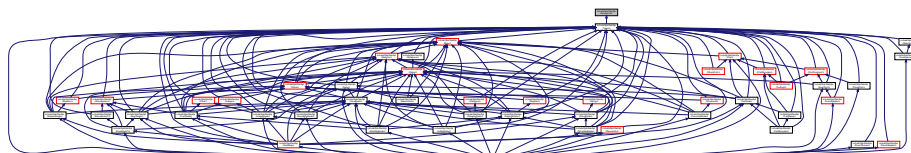
- [Spinnaker](#)

11.45 include/SpinGenApi/Autovector.h File Reference

Include dependency graph for Autovector.h:



This graph shows which files directly or indirectly include this file:



Classes

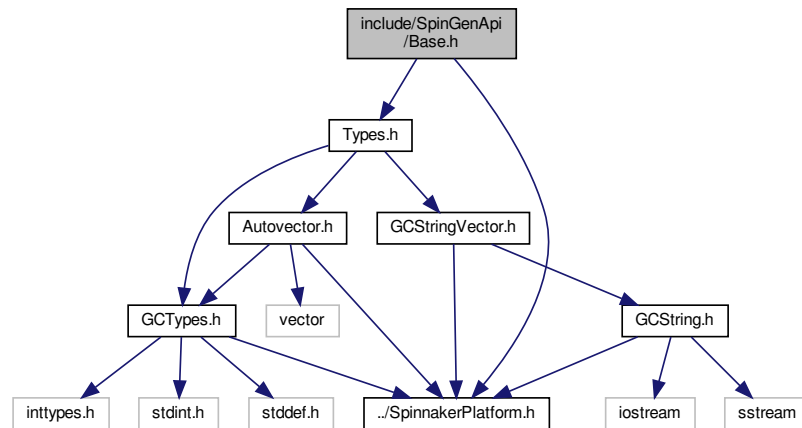
- class [int64_autovector_t](#)
Vector of integers with reference counting.
- class [double_autovector_t](#)
Vector of doubles with reference counting.

Namespaces

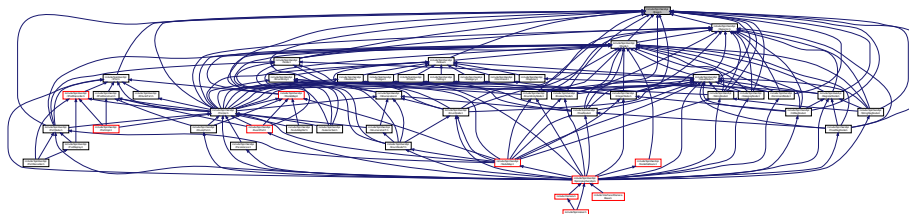
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.46 include/SpinGenApi/Base.h File Reference

Include dependency graph for Base.h:



This graph shows which files directly or indirectly include this file:



Namespaces

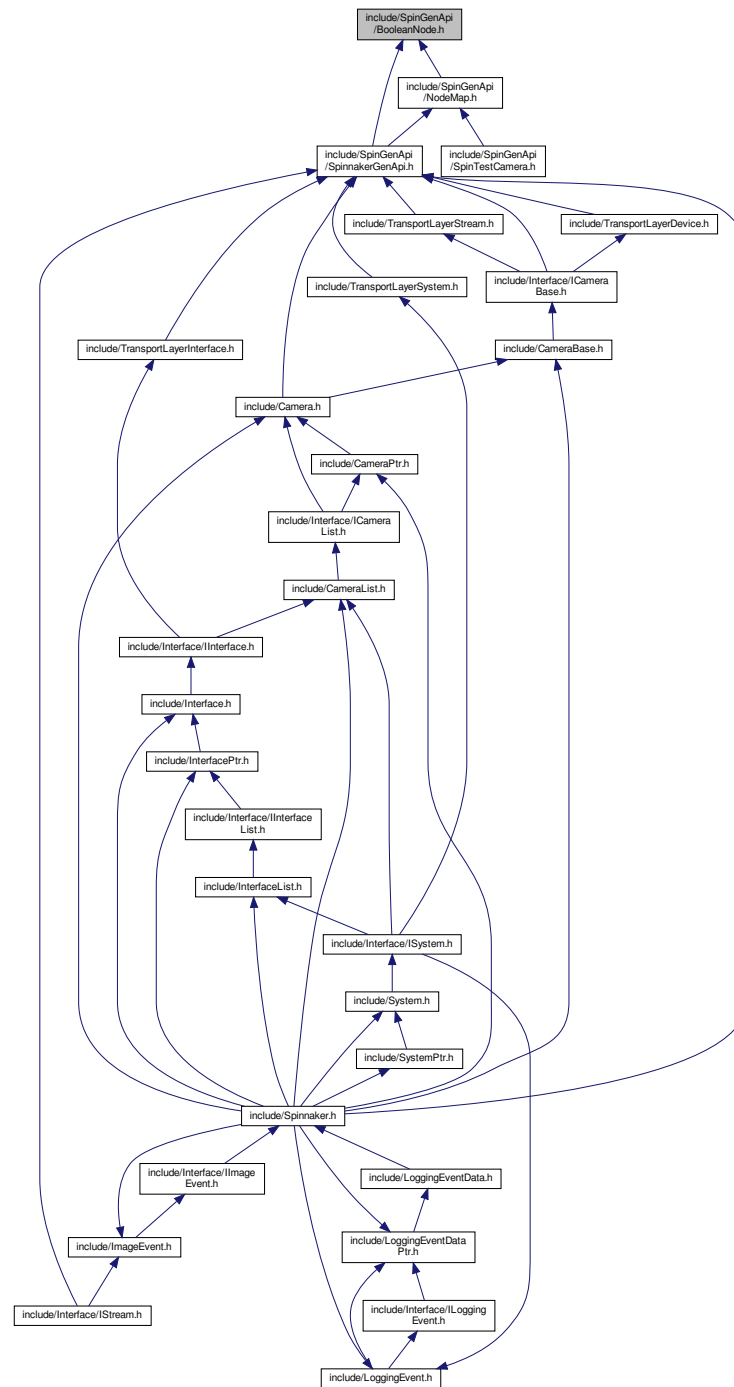
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Variables

- [interface SPINNAKER_API_ABSTRACT IBase](#)

Base interface common to all nodes.

This graph shows which files directly or indirectly include this file:



Classes

- class [BooleanNode](#)
Interface for string properties.

Namespaces

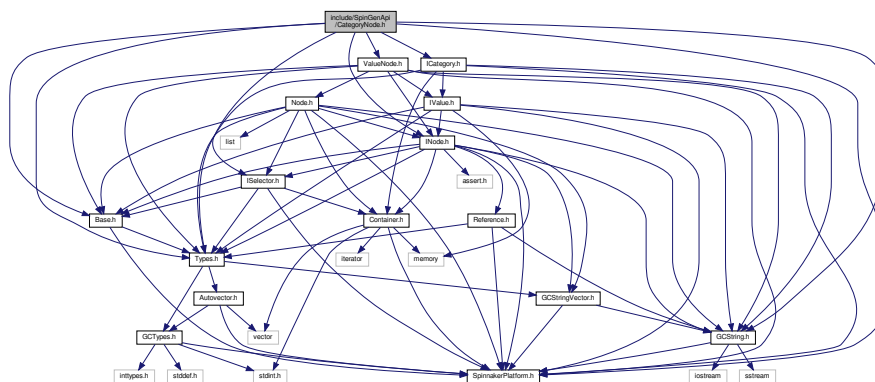
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

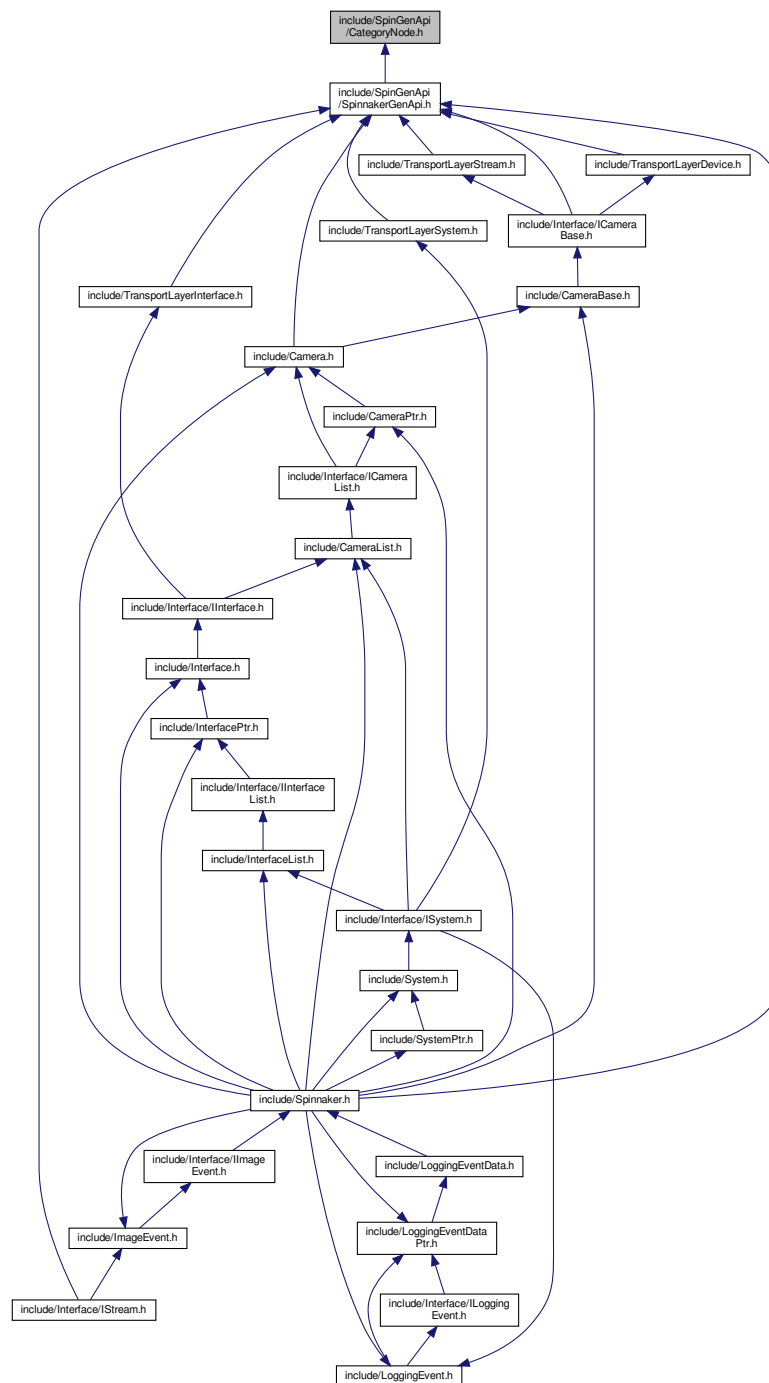
- typedef BooleanNode [CBooleanRef](#)

11.48 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

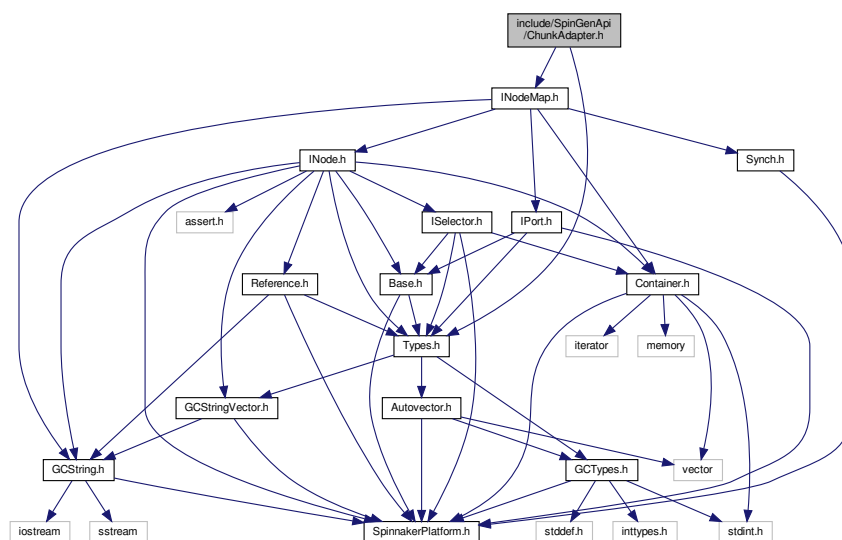
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

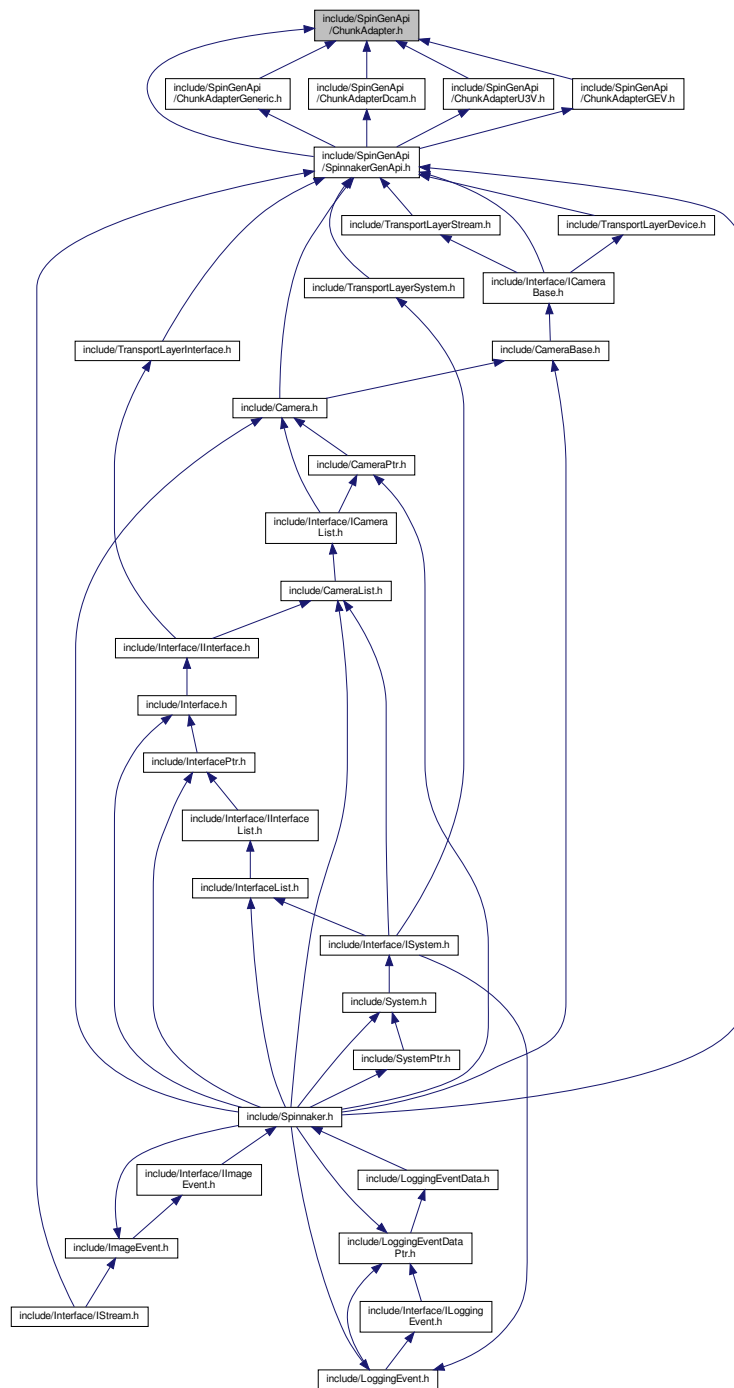
- typedef CategoryNode [CCategoryRef](#)

11.49 include/SpinGenApi/ChunkAdapter.h File Reference

Include dependency graph for ChunkAdapter.h:



This graph shows which files directly or indirectly include this file:



Classes

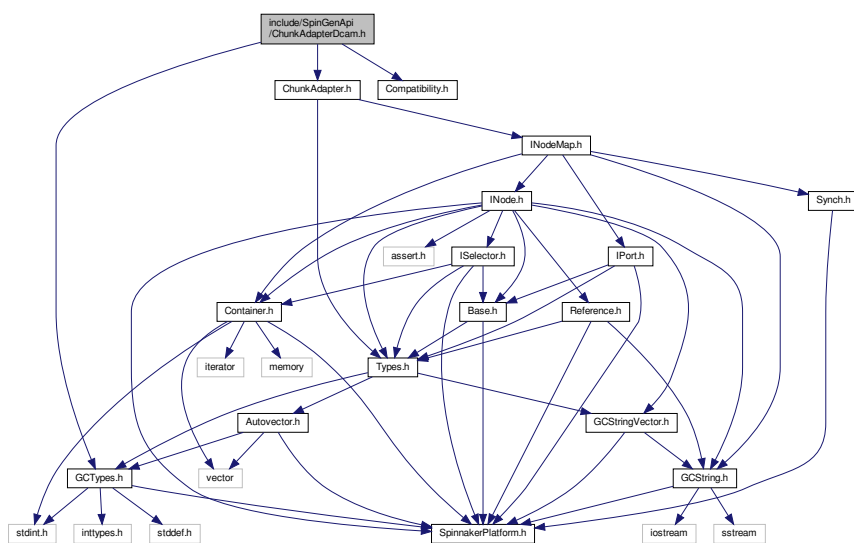
- struct [AttachStatistics_t](#)
Delivers information about the attached chunks and nodes.
- class [CChunkAdapter](#)
Connects a chunked buffer to a node map.

Namespaces

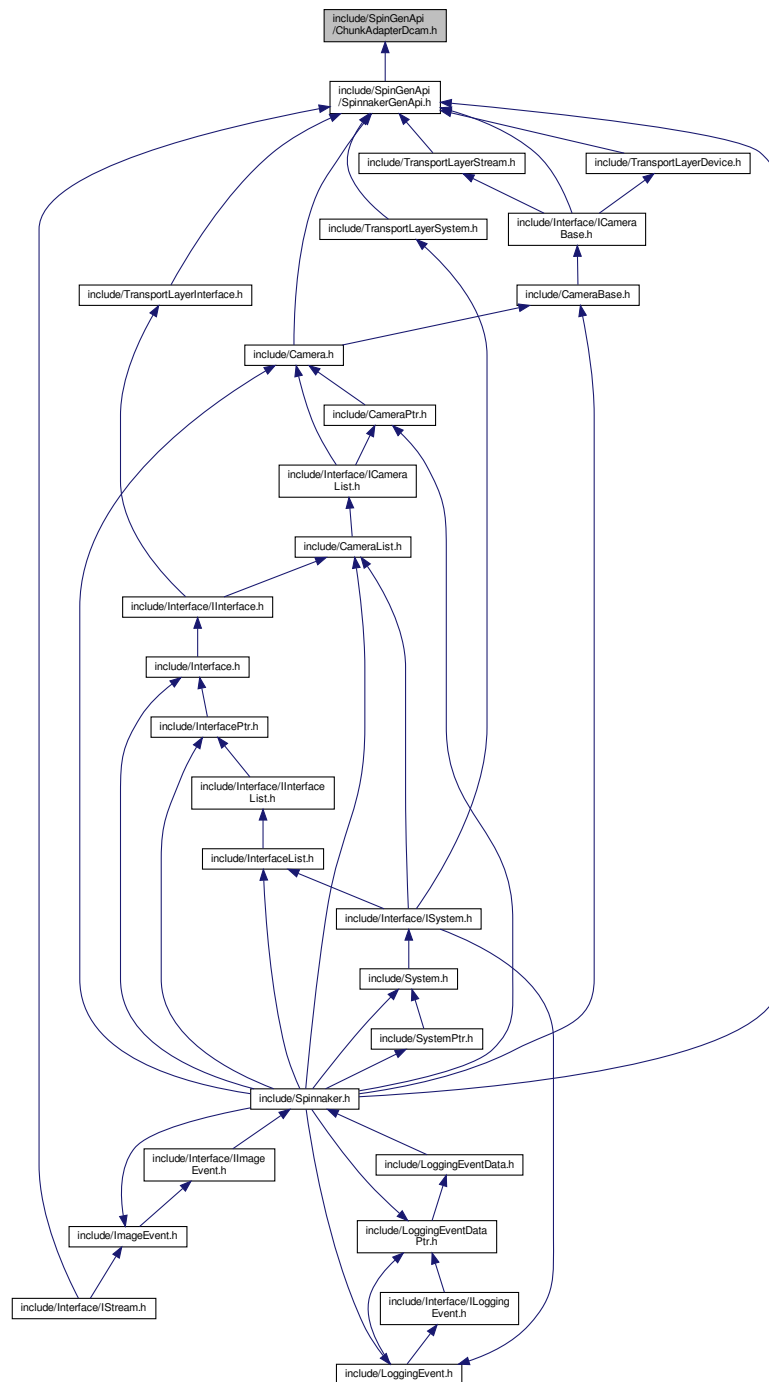
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.50 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

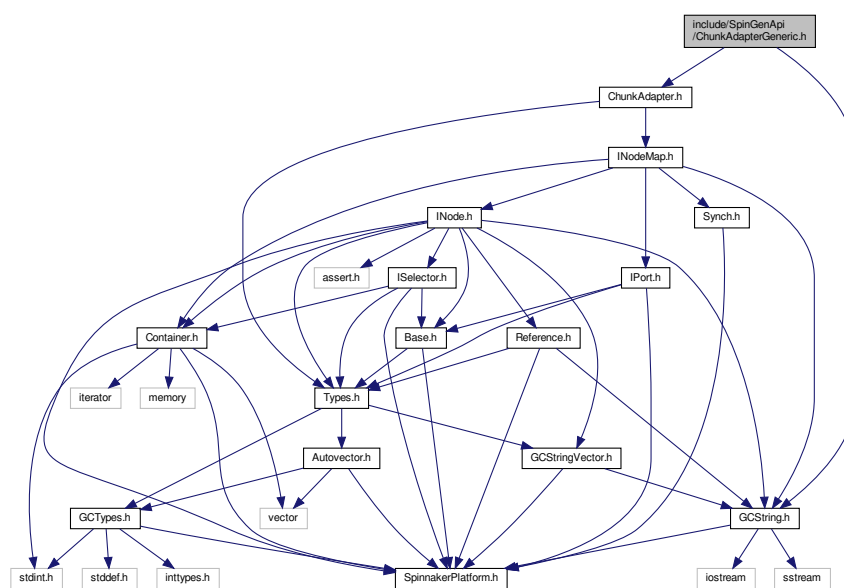
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

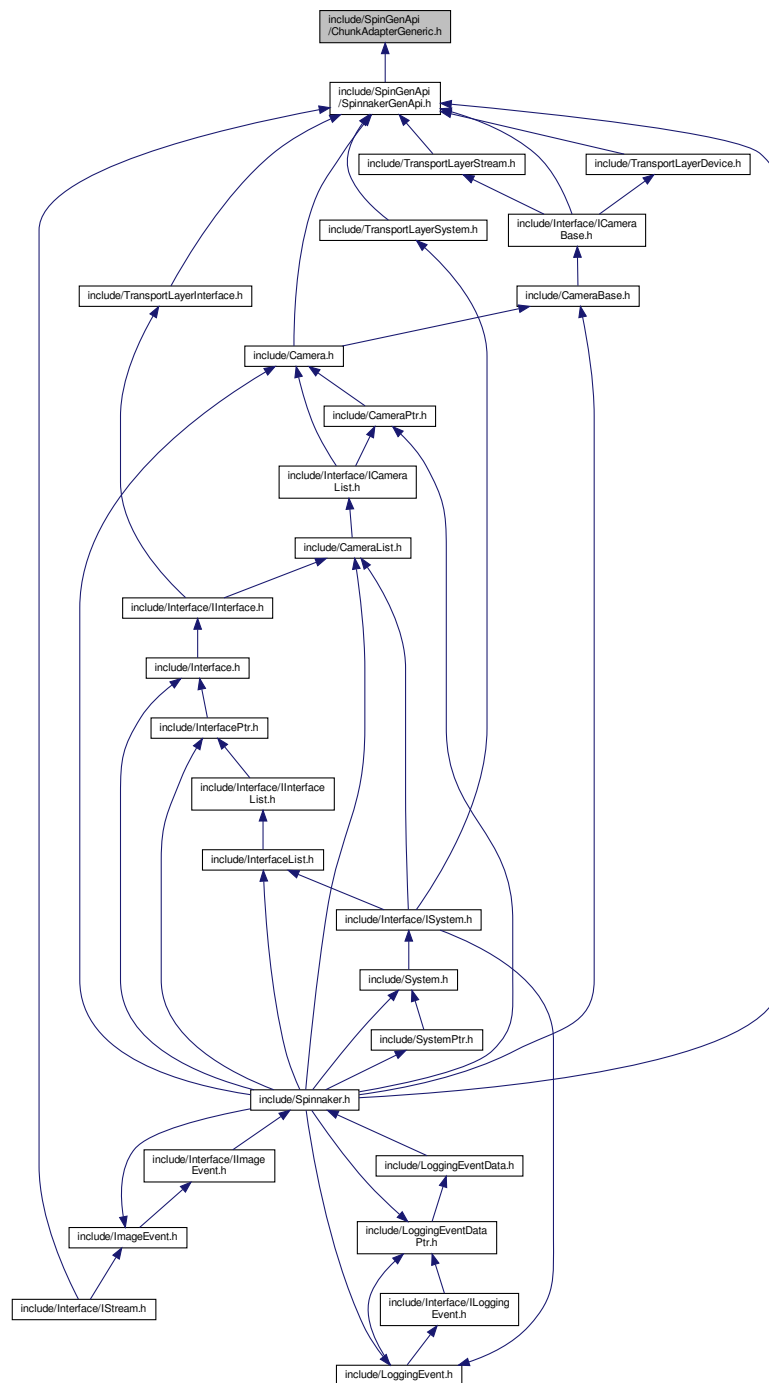
- void [SPINNAKER_API SET_GUID](#) (SPIN_GUID &name, uint32_t l, uint16_t w1, uint16_t w2, uint8_t b1, uint8_t b2, uint8_t b3, uint8_t b4, uint8_t b5, uint8_t b6, uint8_t b7, uint8_t b8)

11.51 include/SpinGenApi/ChunkAdapterGeneric.h File Reference

Include dependency graph for ChunkAdapterGeneric.h:



This graph shows which files directly or indirectly include this file:



Classes

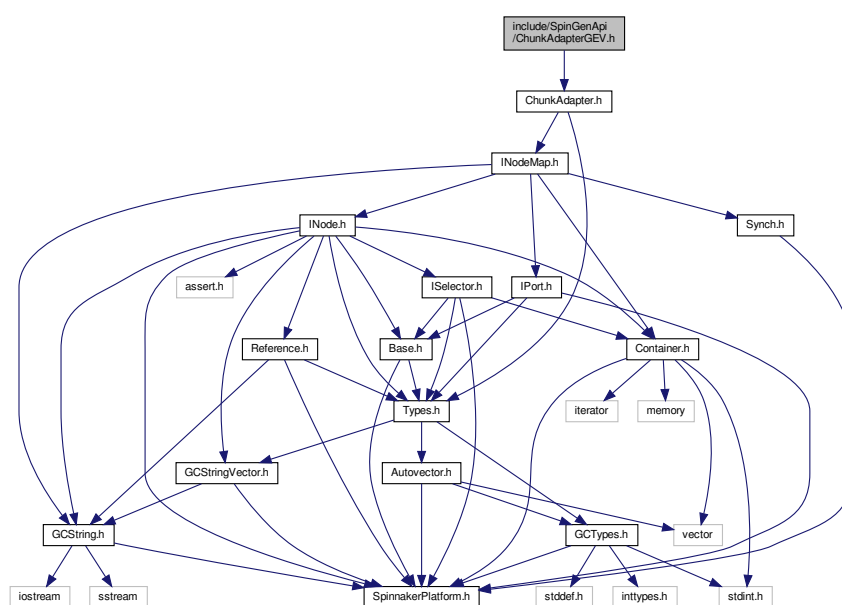
- struct [SingleChunkData_t](#)
- struct [SingleChunkDataStr_t](#)
- class [CChunkAdapterGeneric](#)

Namespaces

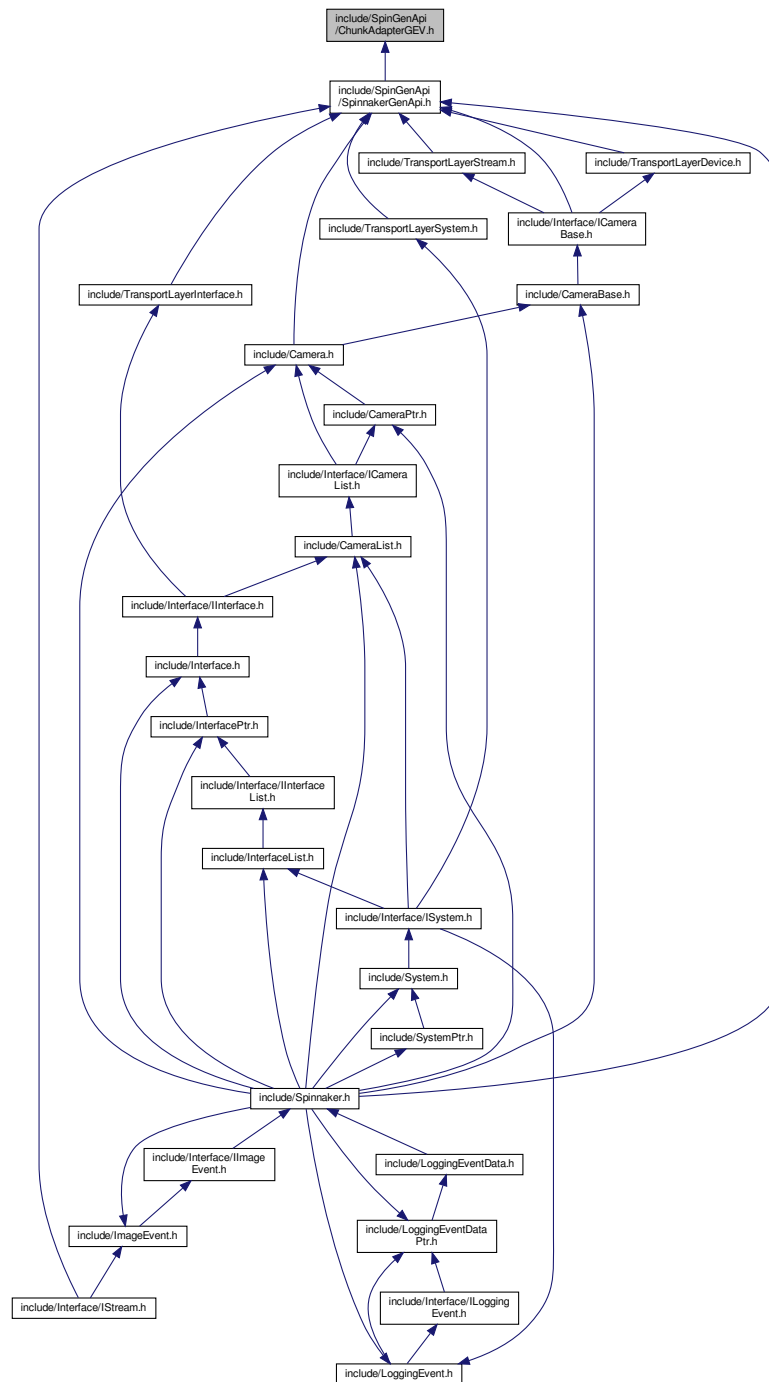
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.52 include/SpinGenApi/ChunkAdapterGEV.h File Reference

Include dependency graph for ChunkAdapterGEV.h:



This graph shows which files directly or indirectly include this file:

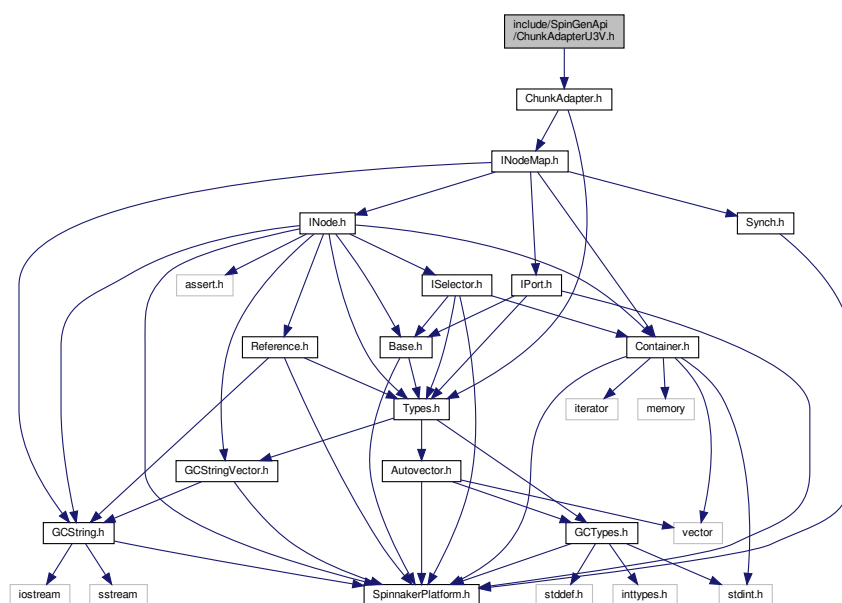


Classes

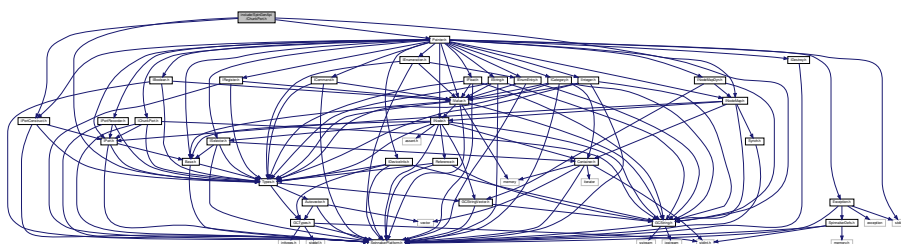
- struct [GVCP_CHUNK_TRAILER](#)
header of a GVCP request packet
- class [CChunkAdapterGEV](#)
Connects a chunked DCAM buffer to a node map.

- Spinnaker
- Spinnaker::GenApi

Include dependency graph for ChunkAdapterU3V.h:

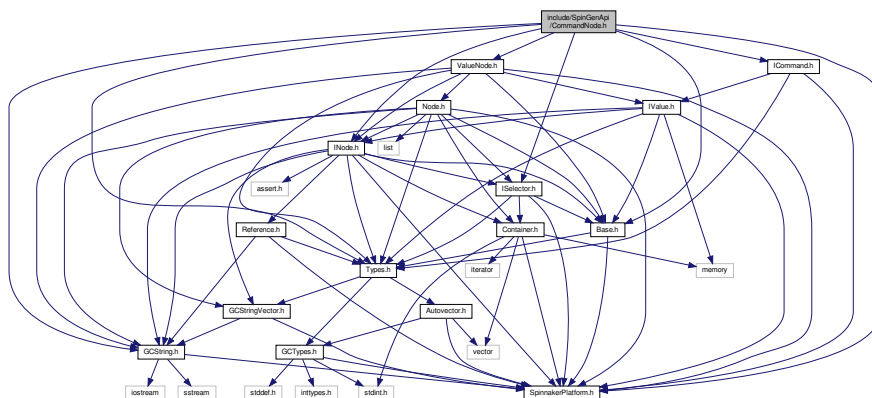


- Spinnaker
- Spinnaker::GenApi

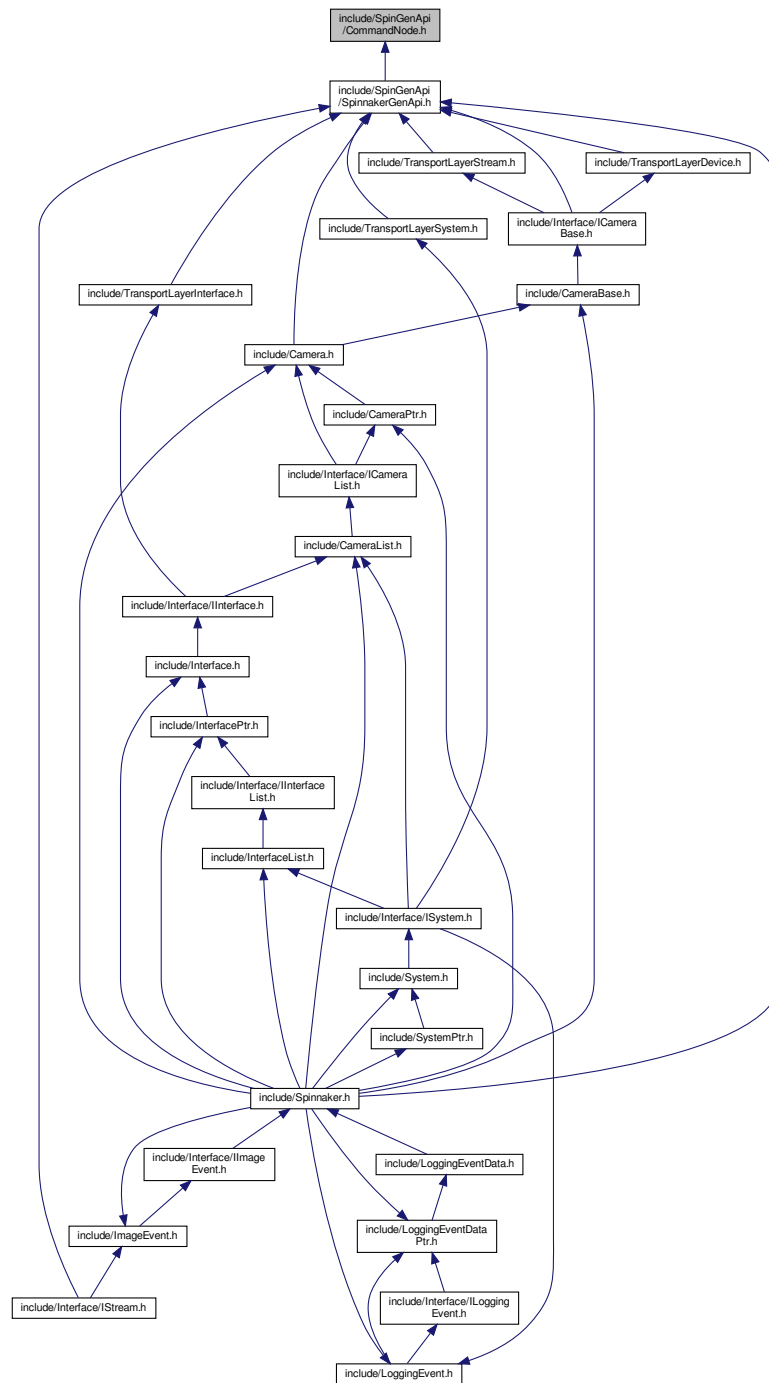


- Spinnaker
- Spinnaker::GenApi

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

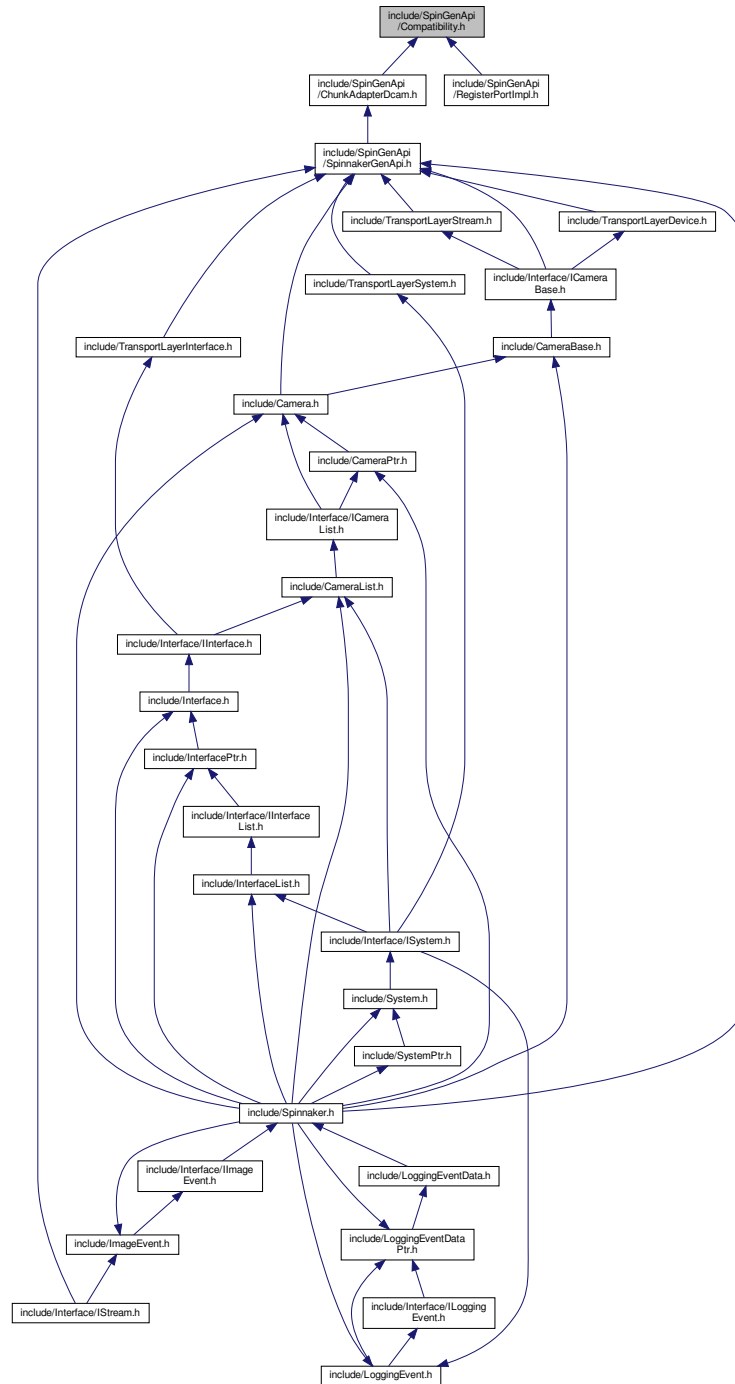
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef CommandNode [CCommandRef](#)

11.56 include/SpinGenApi/Compatibility.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define FMT_I64 "ll"`

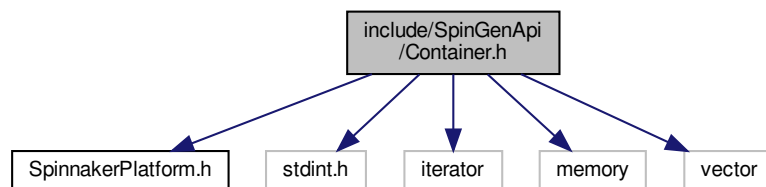
11.56.1 Macro Definition Documentation

11.56.1.1 FMT_I64

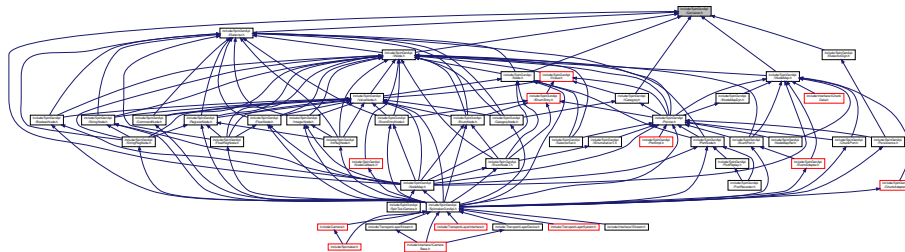
```
#define FMT_I64 "l1"
```

11.57 include/SpinGenApi/Container.h File Reference

Include dependency graph for Container.h:



This graph shows which files directly or indirectly include this file:



11.58 include/SpinGenApi/Counter.h File Reference

Classes

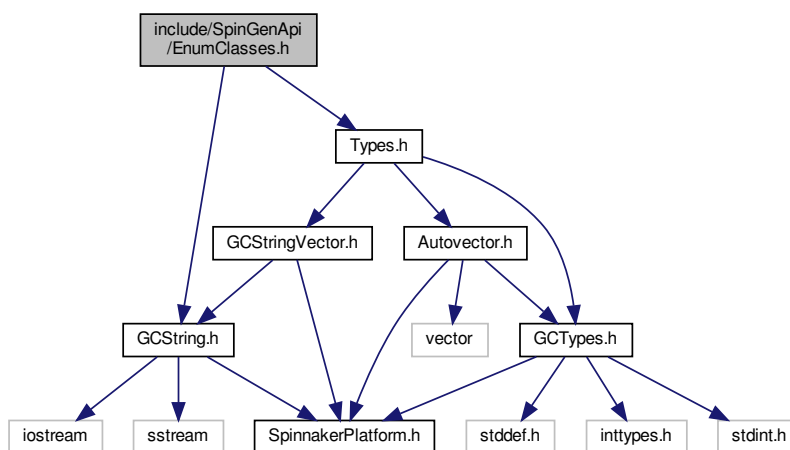
- class [Counter](#)
Definition of a simple [Counter](#) class.

Namespaces

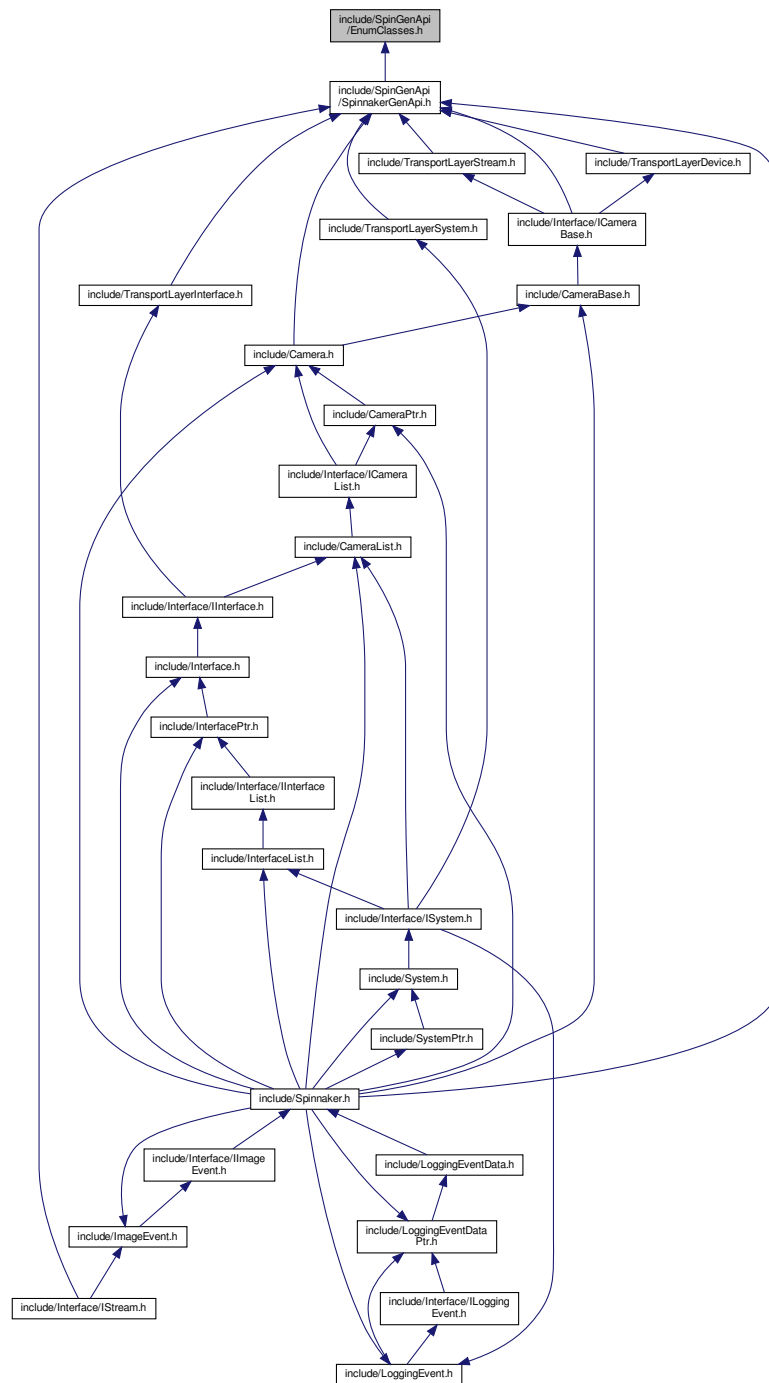
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.59 include/SpinGenApi/EnumClasses.h File Reference

Include dependency graph for EnumClasses.h:



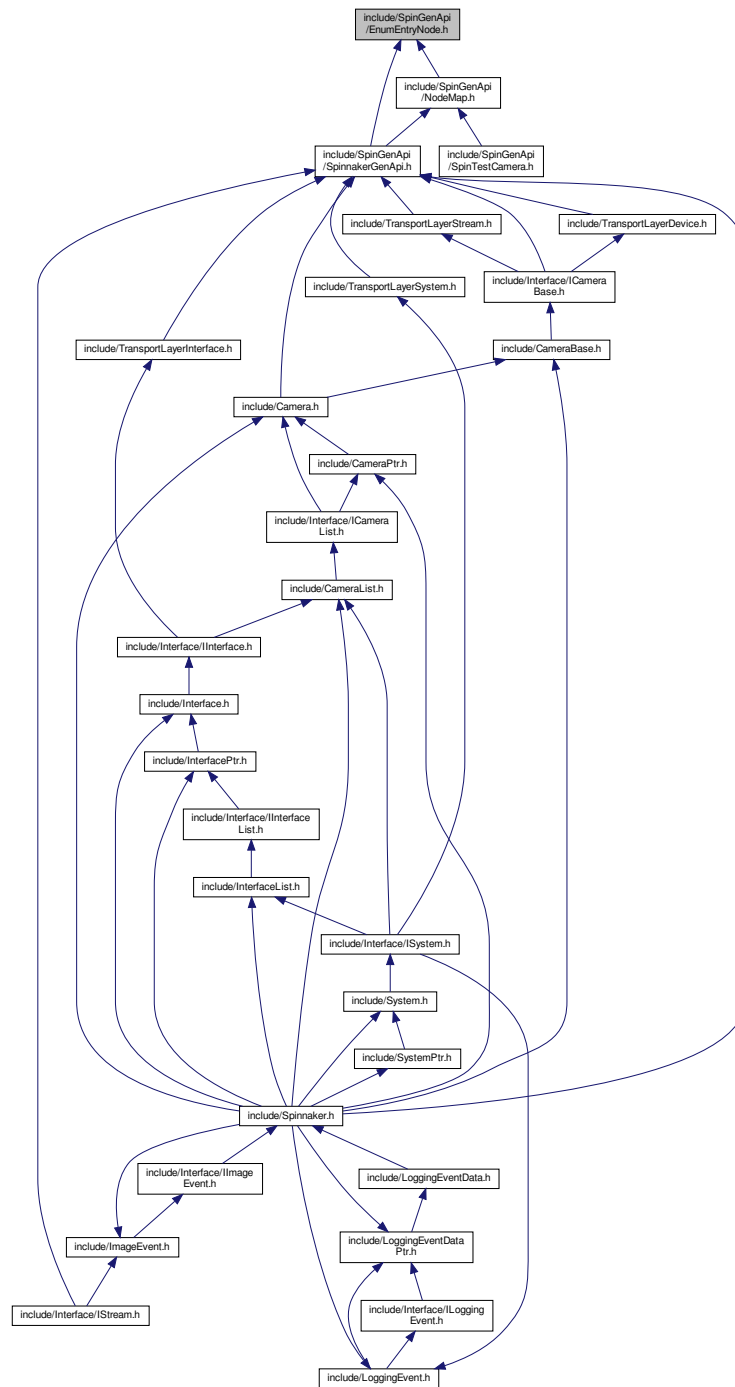
This graph shows which files directly or indirectly include this file:



Classes

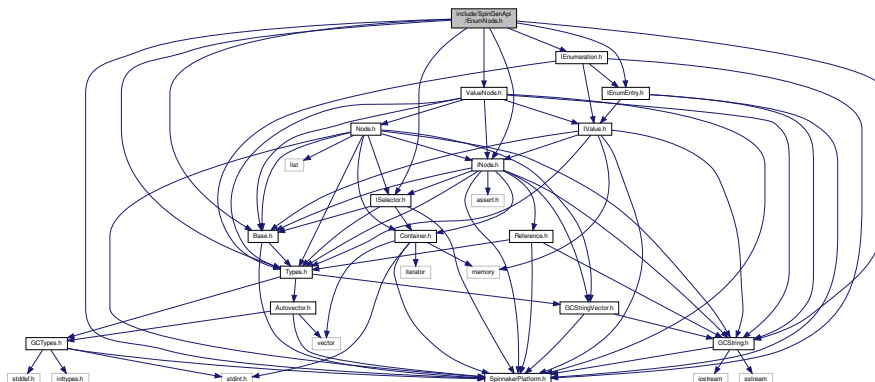
- class [ESignClass](#)
Holds conversion methods for the sign enumeration.
- class [EEndianessClass](#)
Holds conversion methods for the endianess enumeration.
- class [ERepresentationClass](#)

This graph shows which files directly or indirectly include this file:

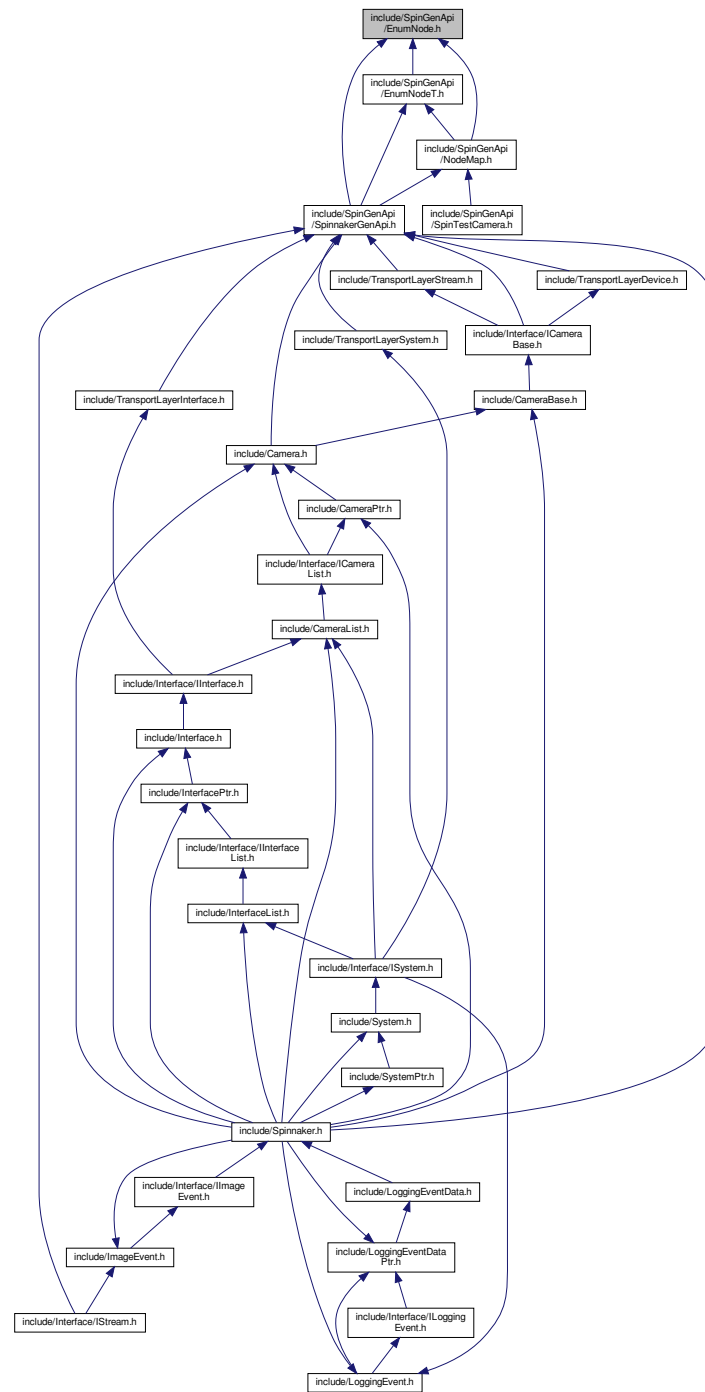


Classes

- class [EnumEntryNode](#)
Interface for string properties.



This graph shows which files directly or indirectly include this file:



Classes

- class [EnumNode](#)
Interface for string properties.

Namespaces

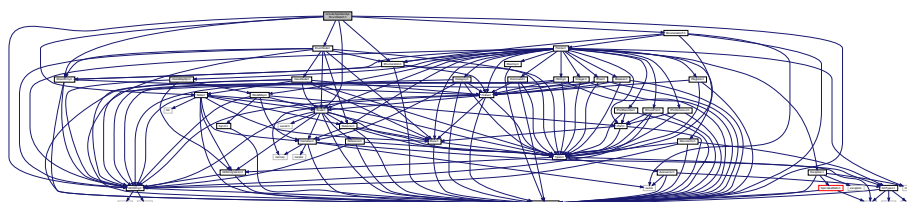
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

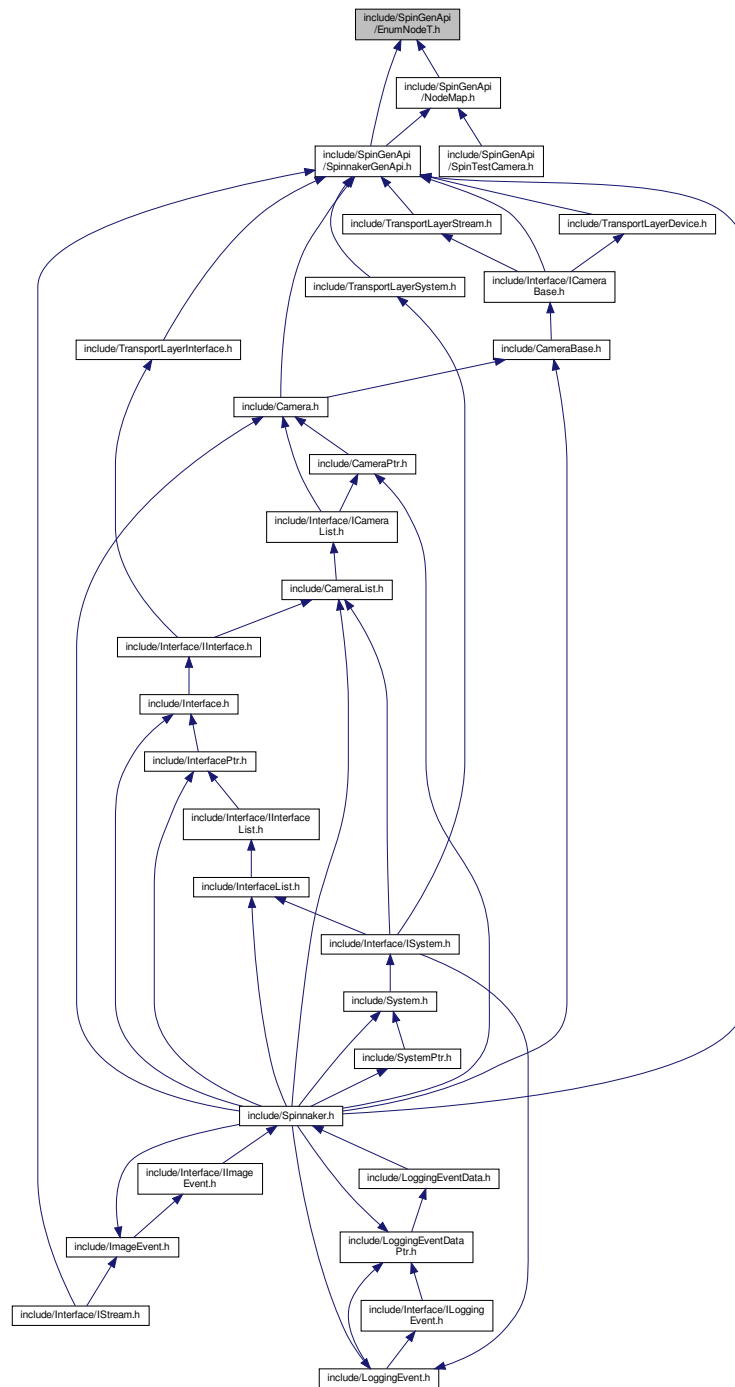
- typedef EnumNode [CEnumerationRef](#)

11.62 include/SpinGenApi/EnumNodeT.h File Reference

Include dependency graph for EnumNodeT.h:



This graph shows which files directly or indirectly include this file:



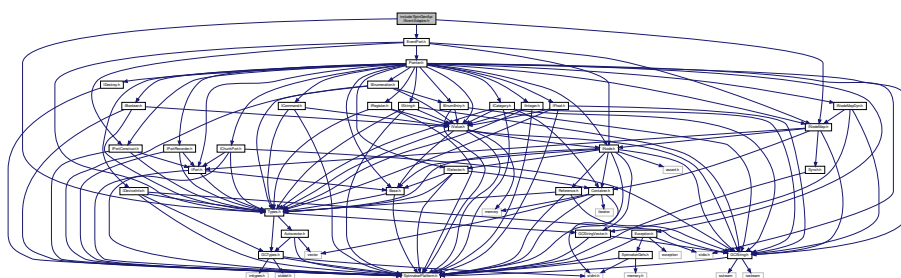
Classes

- class [CEnumerationTRef< EnumT >](#)
Interface for string properties.

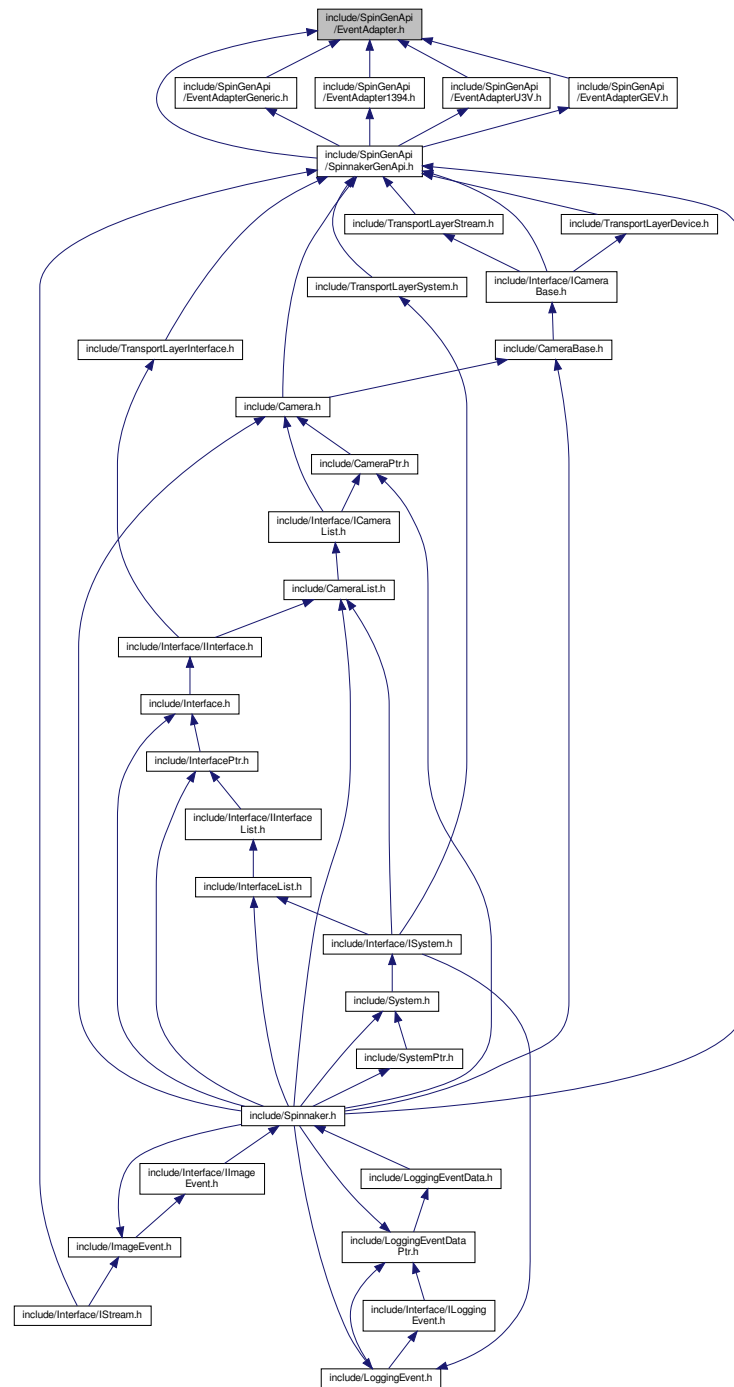
- Spinnaker
- Spinnaker::GenApi

11.63 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](#)

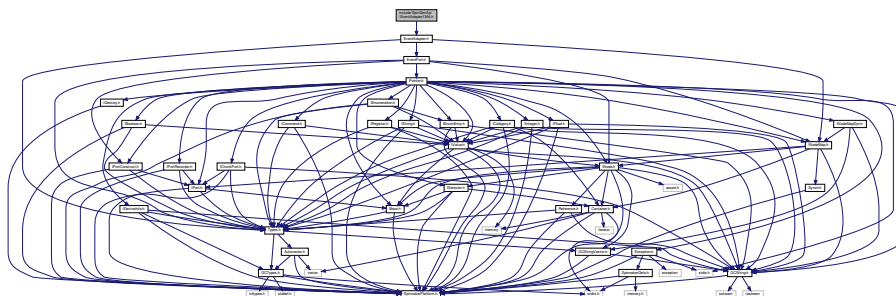
Delivers Events to ports.

Namespaces

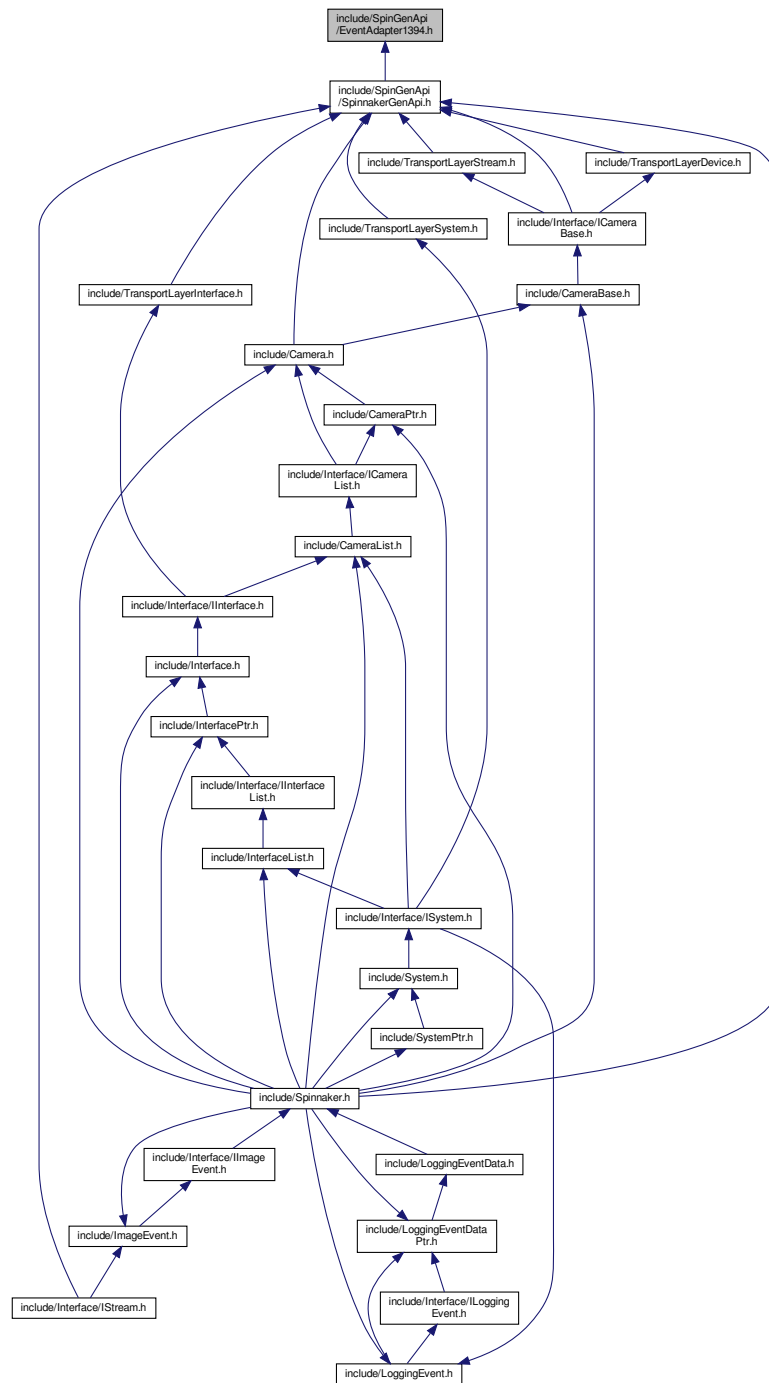
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.64 include/SpinGenApi/EventAdapter1394.h File Reference

Include dependency graph for EventAdapter1394.h:



This graph shows which files directly or indirectly include this file:

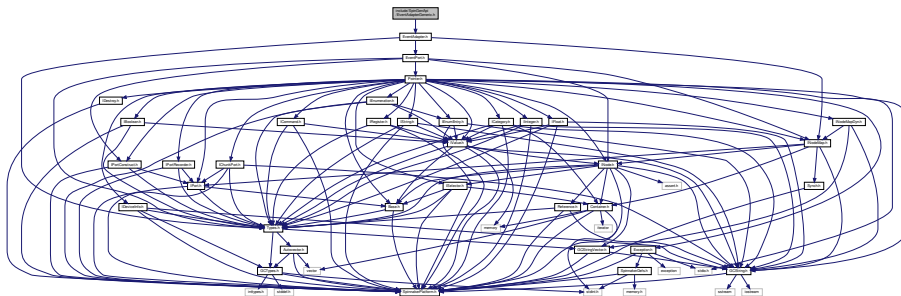


Classes

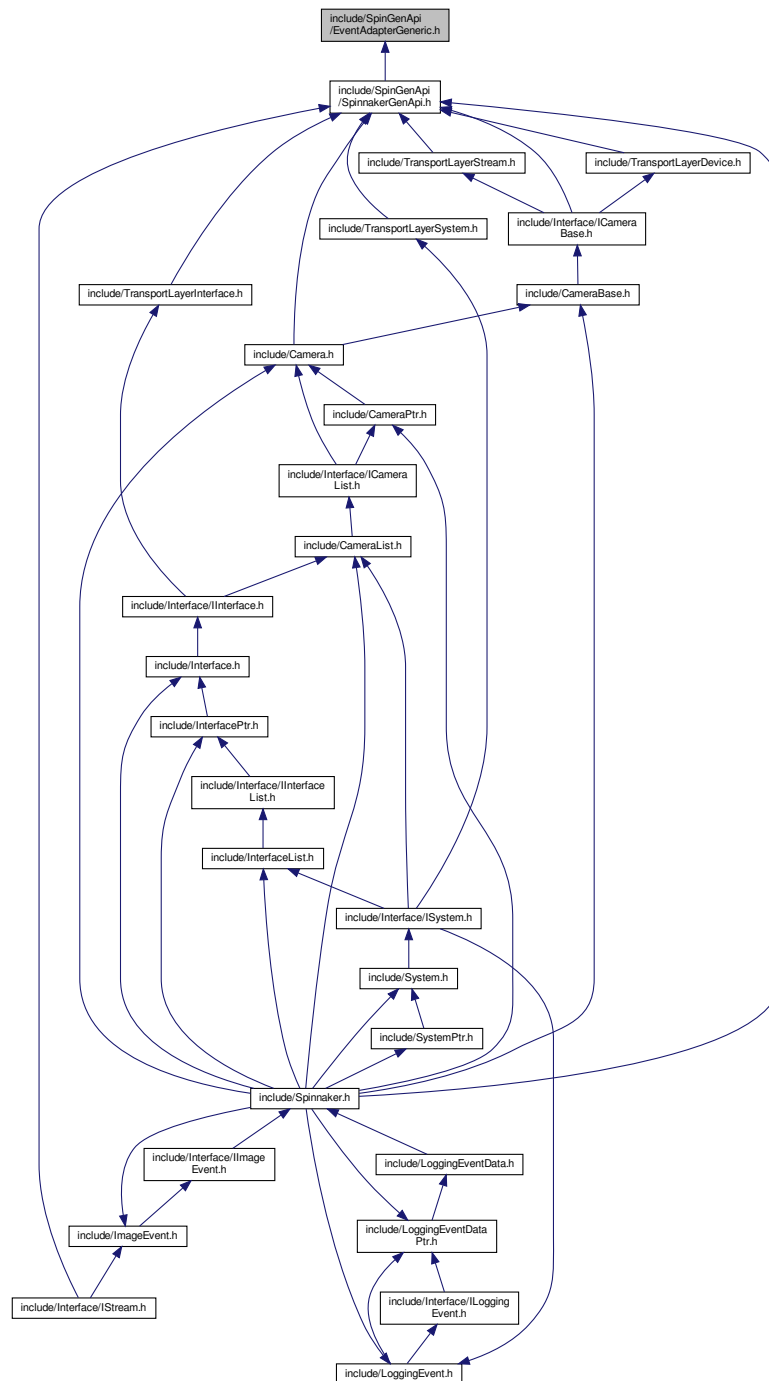
- class [CEventAdapter1394](#)

Distribute the events to the node map.

- Spinnaker
- Spinnaker::GenApi



This graph shows which files directly or indirectly include this file:



Classes

- class [CEventAdapterGeneric](#)

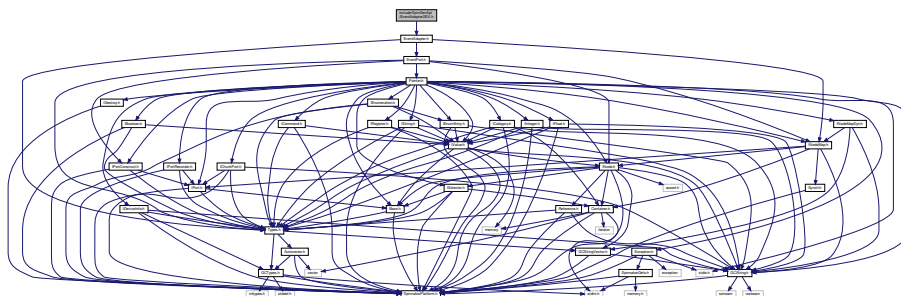
Connects a generic event to a node map.

Namespaces

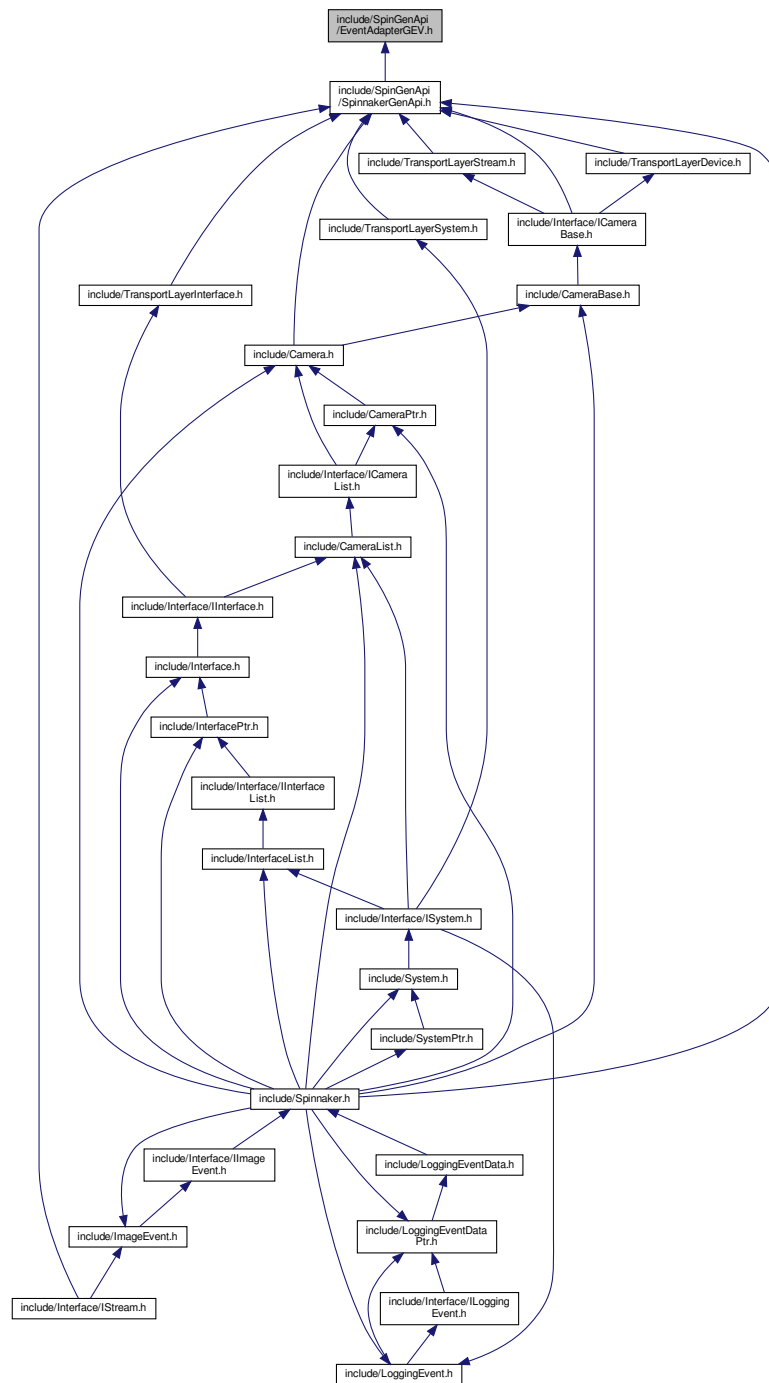
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.66 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

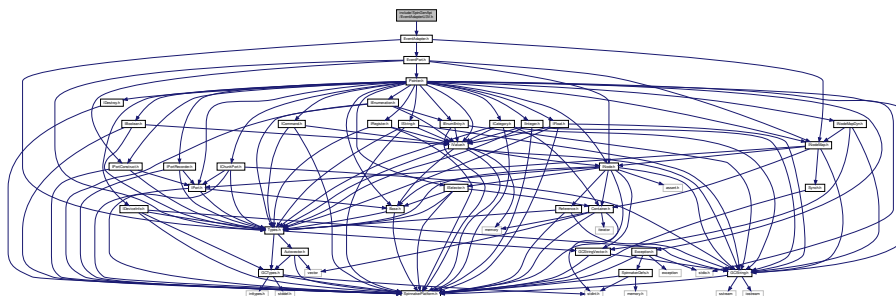
- enum [GVCP_MESSAGE_TAGS](#) {
[TAG_EVENT_CMD](#) = 0xc0,
[TAG_EVENTDATA_CMD](#) = 0xc2 }

Variables

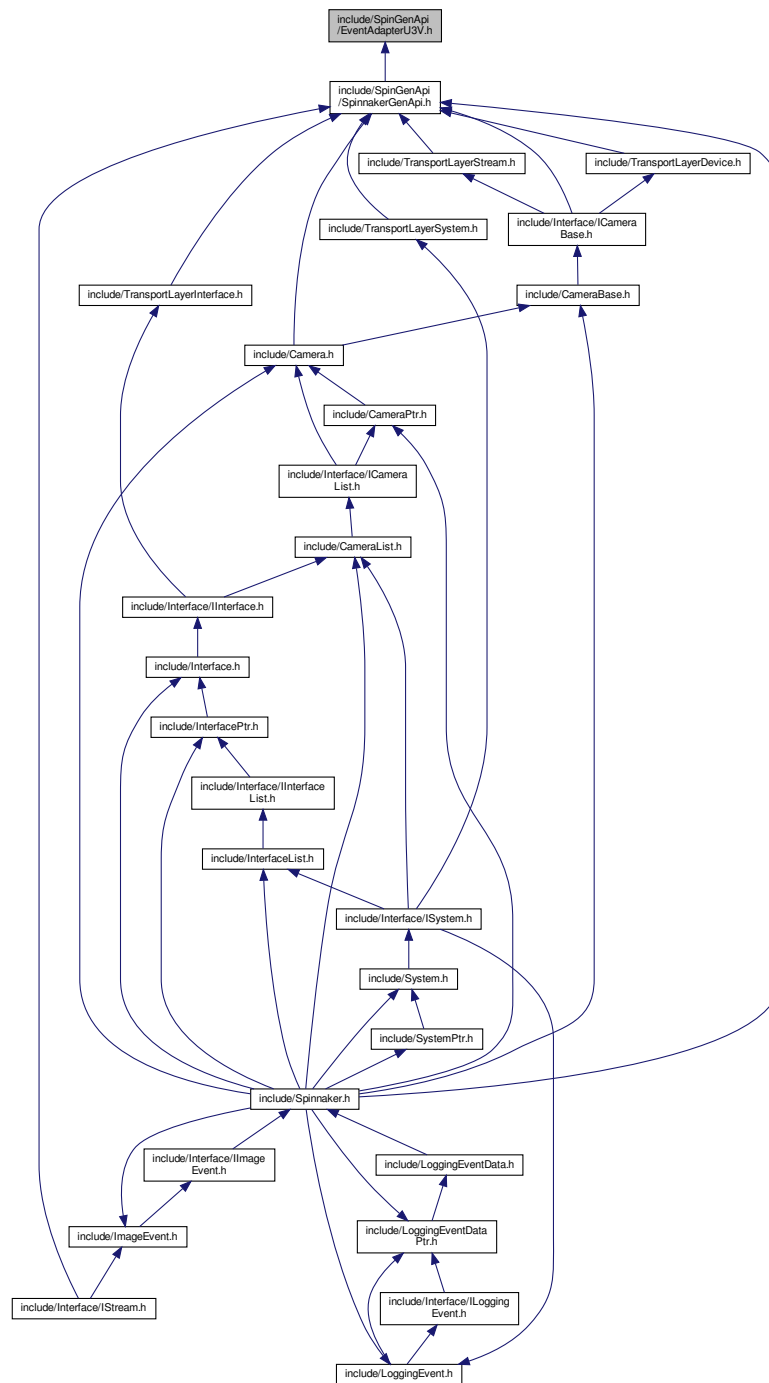
- const uint8_t [COMMAND_MAGIC](#) = 0x42

11.67 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

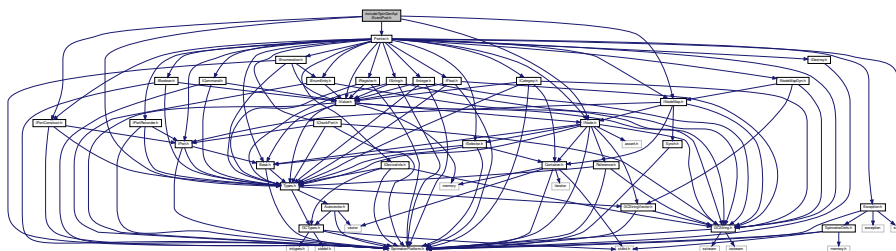
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Variables

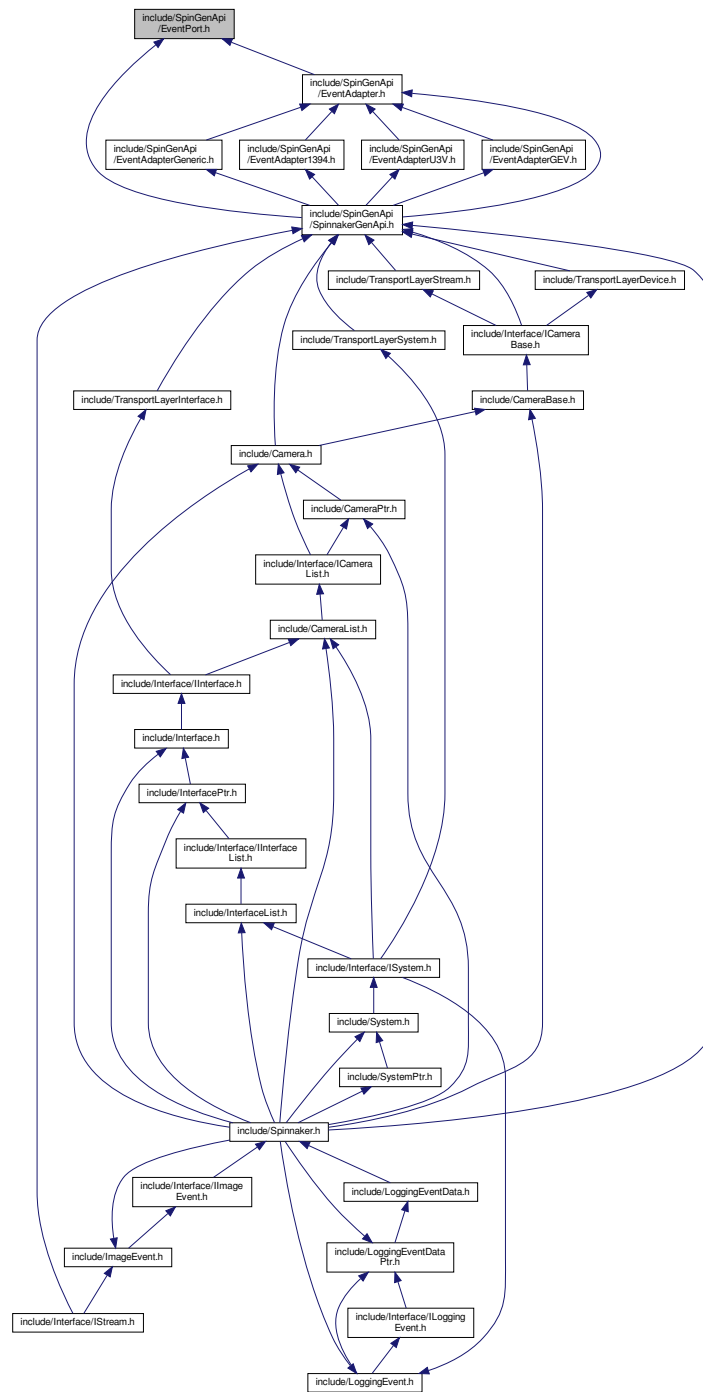
- const uint32_t [U3V_EVENT_PREFIX](#) = 0x45563355
- const uint16_t [GENCP_EVENT_CMD_ID](#) = 0x0C00
- const size_t [GENCP_COMMAND_HEADER_SIZE](#) = sizeof([U3V_COMMAND_HEADER](#))
- const size_t [GENCP_EVENT_BASIC_SIZE](#) = sizeof([U3V_EVENT_MESSAGE](#))

11.68 include/SpinGenApi/EventPort.h File Reference

Include dependency graph for EventPort.h:



This graph shows which files directly or indirectly include this file:



Classes

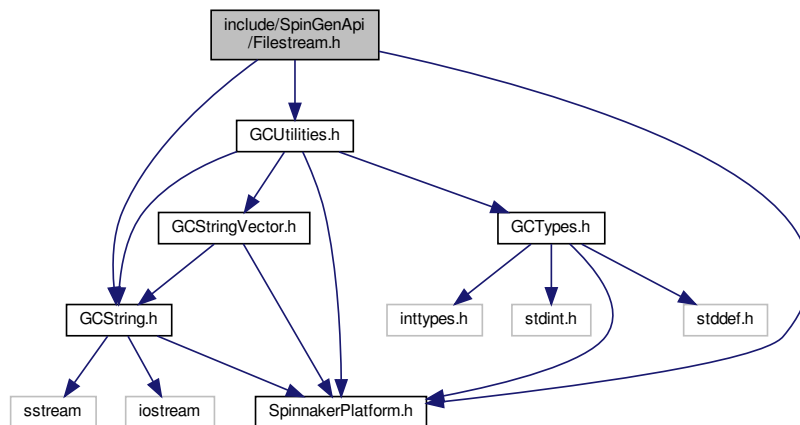
- class [CEventPort](#)

Port attachable to an event.

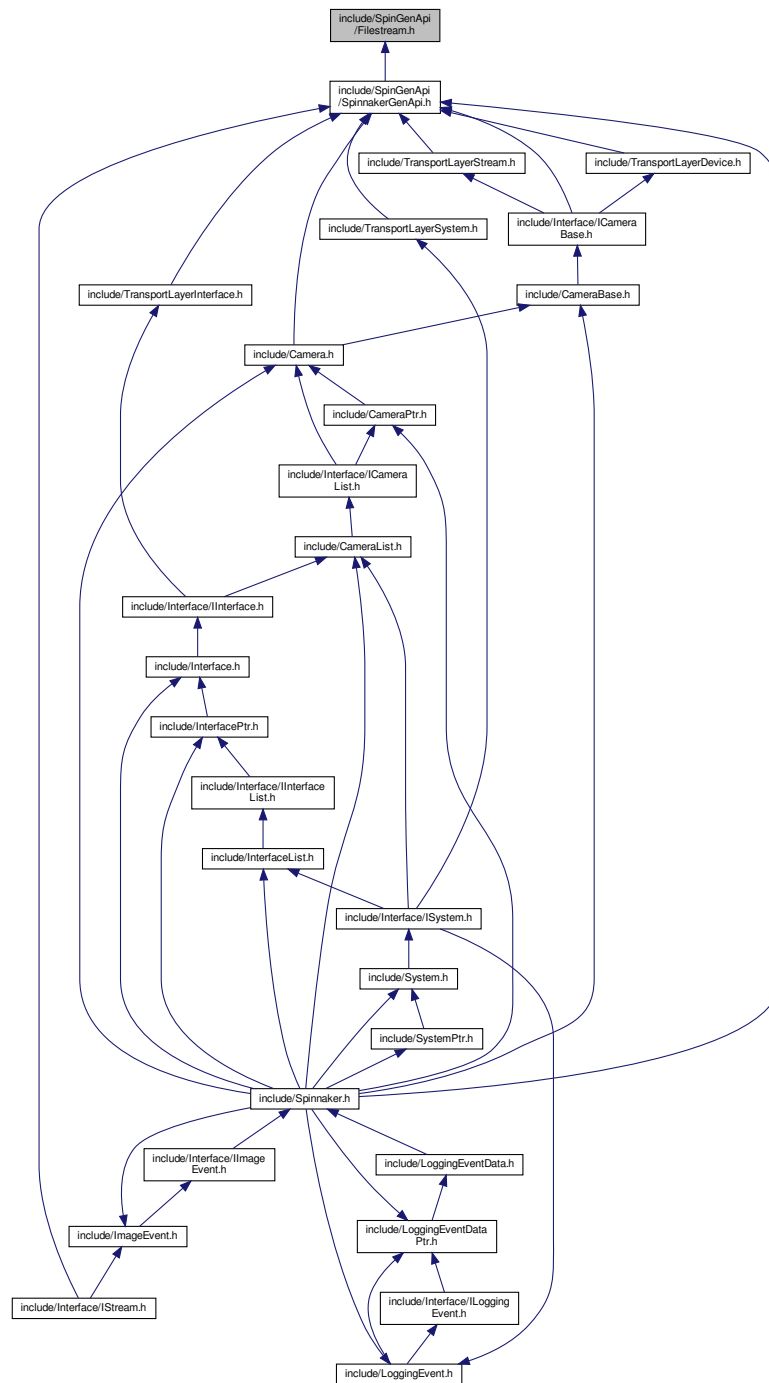
- Spinnaker
- Spinnaker::GenApi

11.69 include/SpinGenApi/Filestream.h File Reference

Include dependency graph for Filestream.h:



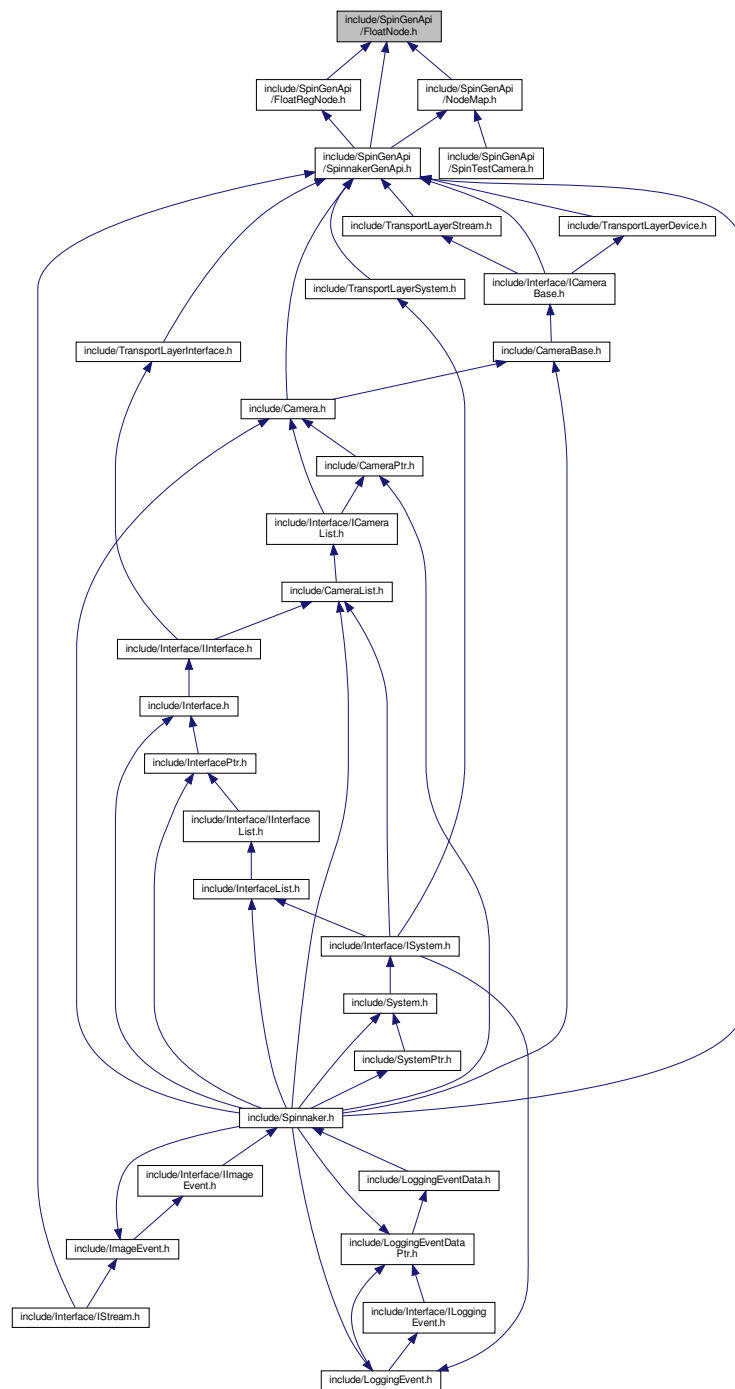
This graph shows which files directly or indirectly include this file:



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 >](#)

This graph shows which files directly or indirectly include this file:



Classes

- class [FloatNode](#)
Interface for string properties.

Namespaces

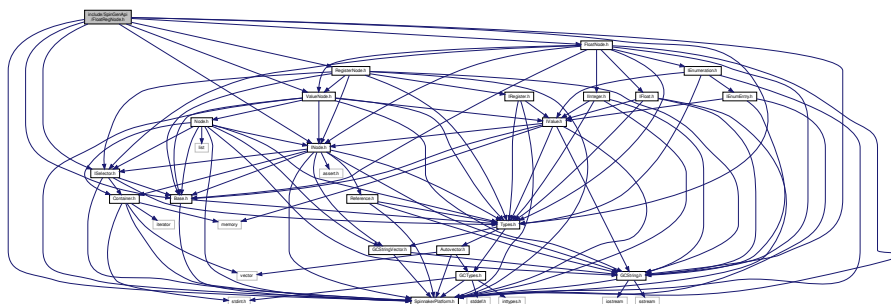
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

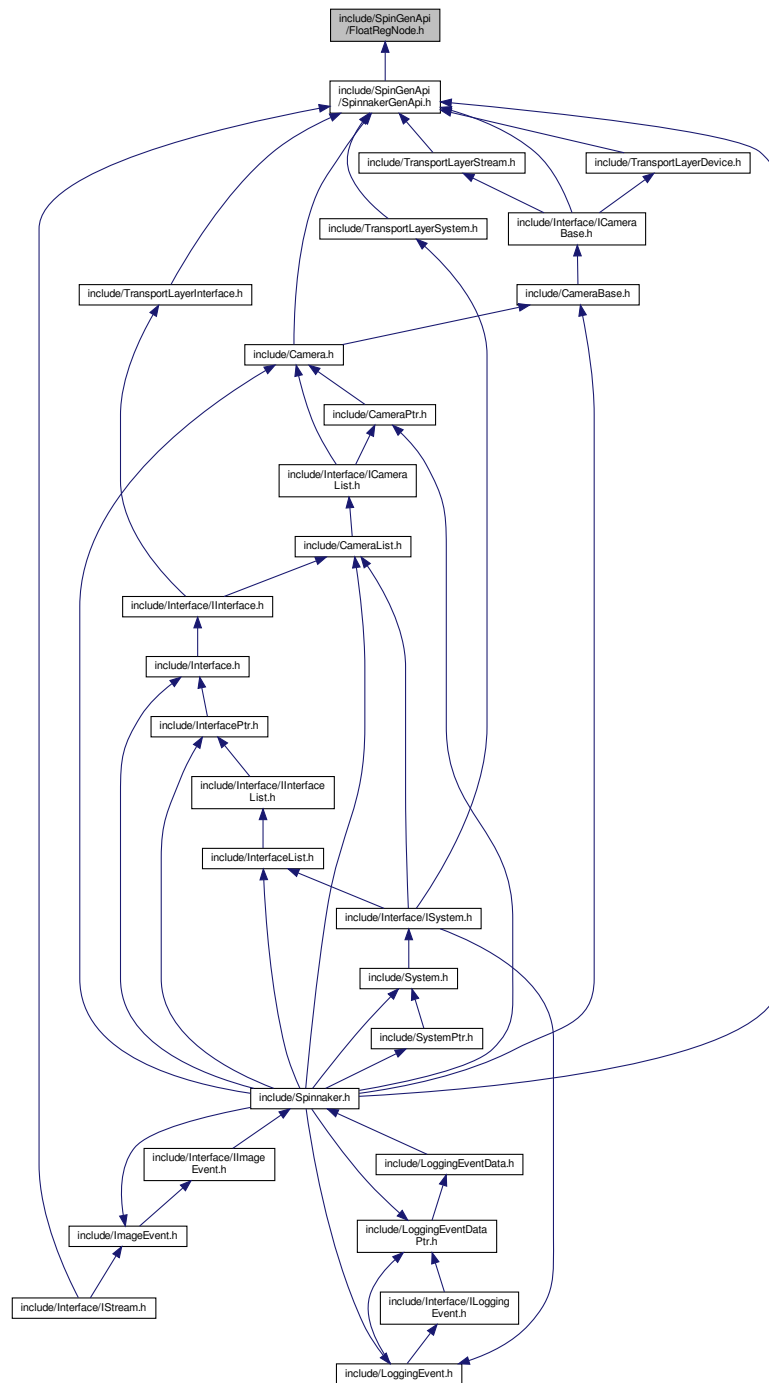
- typedef FloatNode [CFloatRef](#)

11.71 include/SpinGenApi/FloatRegNode.h File Reference

Include dependency graph for FloatRegNode.h:



This graph shows which files directly or indirectly include this file:



Classes

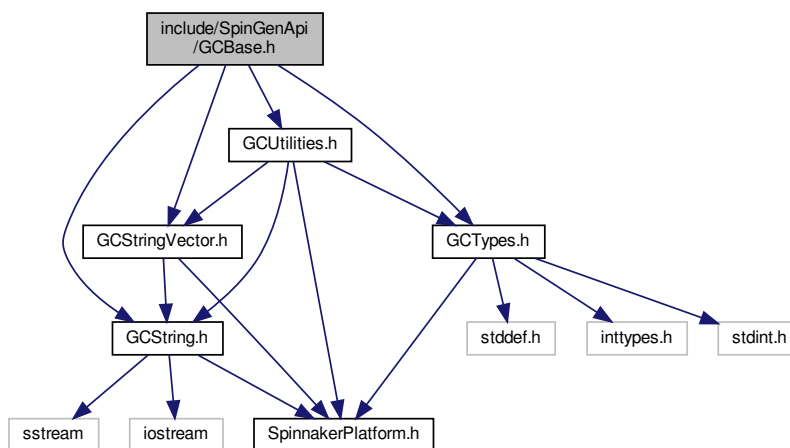
- class [FloatRegNode](#)
Interface for string properties.

Namespaces

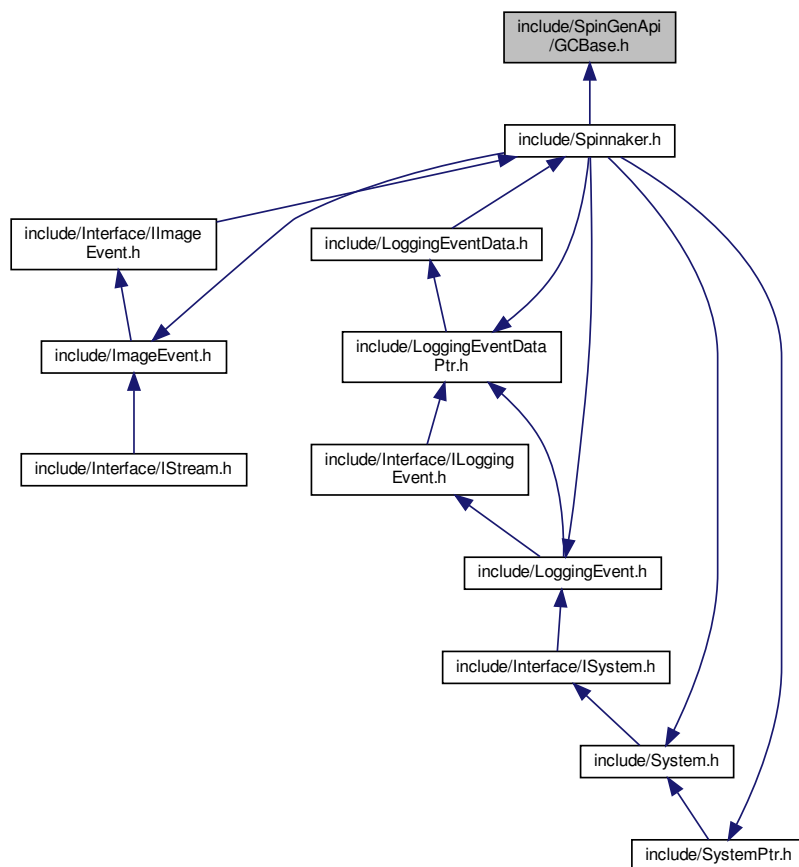
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.72 include/SpinGenApi/GCBase.h File Reference

Include dependency graph for GCBase.h:

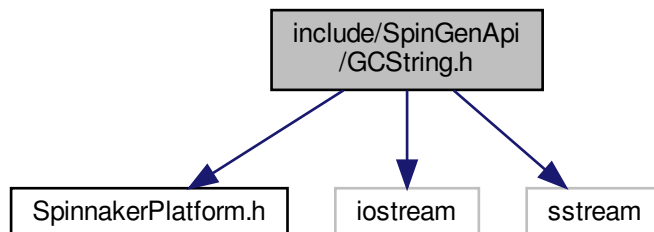


This graph shows which files directly or indirectly include this file:

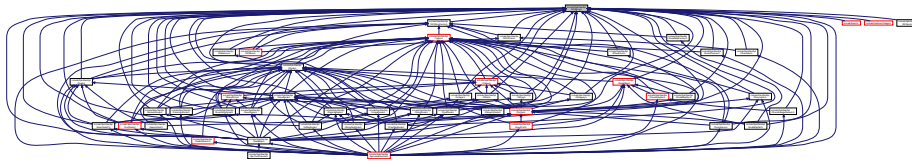


11.73 include/SpinGenApi/GCString.h File Reference

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.

11.73.1 Macro Definition Documentation

11.73.1.1 GCSTRING_NPOS

```
#define GCSTRING_NPOS size_t(-1)
```

11.73.2 Function Documentation

11.73.2.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & ostr,
    const Spinnaker::GenICam::gcstring & str ) [inline]
```

STL operator out.

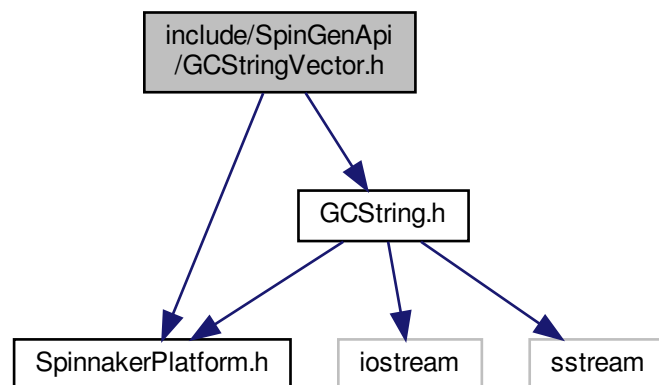
11.73.2.2 operator>>()

```
std::istream& operator>> (
    std::istream & istr,
    Spinnaker::GenICam::gcstring & str ) [inline]
```

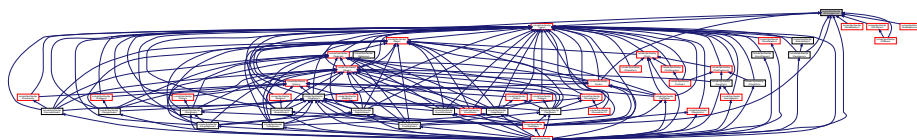
STL operator in.

11.74 include/SpinGenApi/GCStringVector.h File Reference

Include dependency graph for GCStringVector.h:

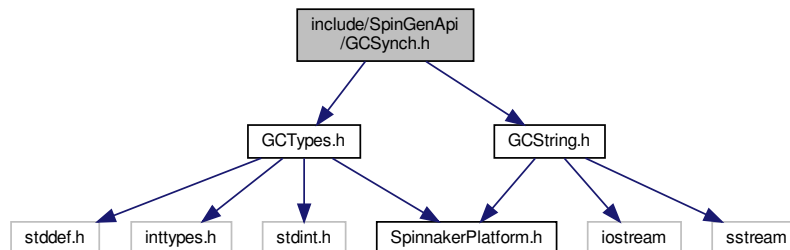


This graph shows which files directly or indirectly include this file:



11.75 include/SpinGenApi/GCSynch.h File Reference

Include dependency graph for GCSynch.h:



Classes

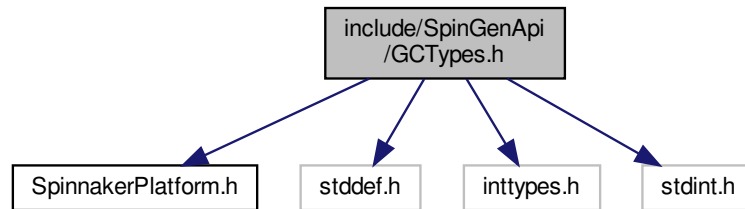
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)
- class [LockableObject< Object >](#)
Instance-Lock for an object.
- class [LockableObject< Object >::Lock](#)
A scopelevel [Lock](#) class.
- class [CGlobalLock](#)
Named global lock which can be used over process boundaries.
- class [CGlobalLockUnlocker](#)
Unlocks the global lock object on destruction.

Namespaces

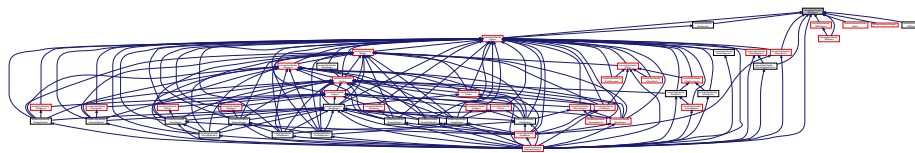
- [Spinnaker](#)
- [Spinnaker::GenICam](#)

11.76 include/SpinGenApi/GCTypes.h File Reference

Include dependency graph for GCTypes.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Version_t](#)
Version.

Namespaces

- [Spinnaker](#)
- [Spinnaker::GenICam](#)

Macros

- `#define __STDC_LIMIT_MACROS`
- `#define __STDC_CONSTANT_MACROS`
- `#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */`
- `#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */`
- `#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */`
- `#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */`
- `#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */`
- `#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */`
- `#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */`
- `#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffff80LL) /* minimum signed int8 value */`
- `#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */`

Typedefs

- typedef float [float32_t](#)
32 bit floating point
- typedef double [float64_t](#)
64 bit floating point

11.76.1 Macro Definition Documentation

11.76.1.1 __STDC_CONSTANT_MACROS

```
#define __STDC_CONSTANT_MACROS
```

11.76.1.2 __STDC_LIMIT_MACROS

```
#define __STDC_LIMIT_MACROS
```

11.76.1.3 GC_INT32_MAX

```
#define GC_INT32_MAX static_cast<int64_t>(0x000000007fffffffLL) /* maximum signed int32 value */
```

11.76.1.4 GC_INT32_MIN

```
#define GC_INT32_MIN static_cast<int64_t>(0xffffffff80000000LL) /* minimum signed int32 value */
```

11.76.1.5 GC_INT64_MAX

```
#define GC_INT64_MAX static_cast<int64_t>(0x7fffffffffffffffLL) /* maximum signed int64 value */
```

11.76.1.6 GC_INT64_MIN

```
#define GC_INT64_MIN static_cast<int64_t>(0x8000000000000000LL) /* minimum signed int64 value */
```

11.76.1.7 GC_INT8_MAX

```
#define GC_INT8_MAX static_cast<int64_t>(0x000000000000007fLL) /* maximum signed int8 value */
```

11.76.1.8 GC_INT8_MIN

```
#define GC_INT8_MIN static_cast<int64_t>(0xffffffffffffff80LL) /* minimum signed int8 value */
```

11.76.1.9 GC_UINT32_MAX

```
#define GC_UINT32_MAX static_cast<uint64_t>(0x00000000ffffffffULL) /* maximum unsigned int32 value */
```

11.76.1.10 GC_UINT64_MAX

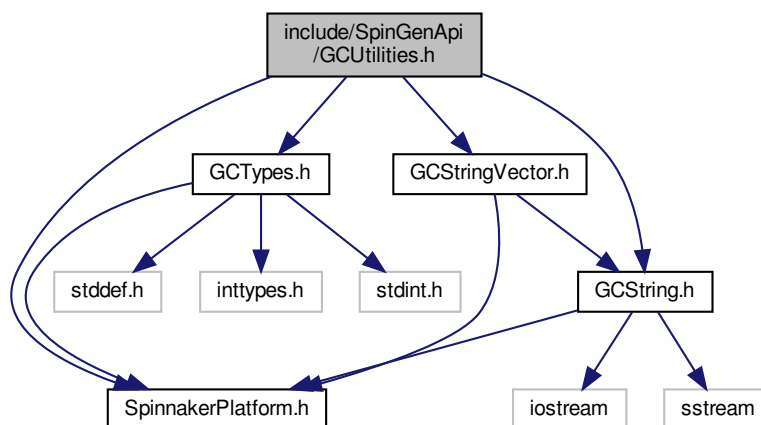
```
#define GC_UINT64_MAX static_cast<uint64_t>(0xffffffffffffffffULL) /* maximum unsigned int64 value */
```

11.76.1.11 GC_UINT8_MAX

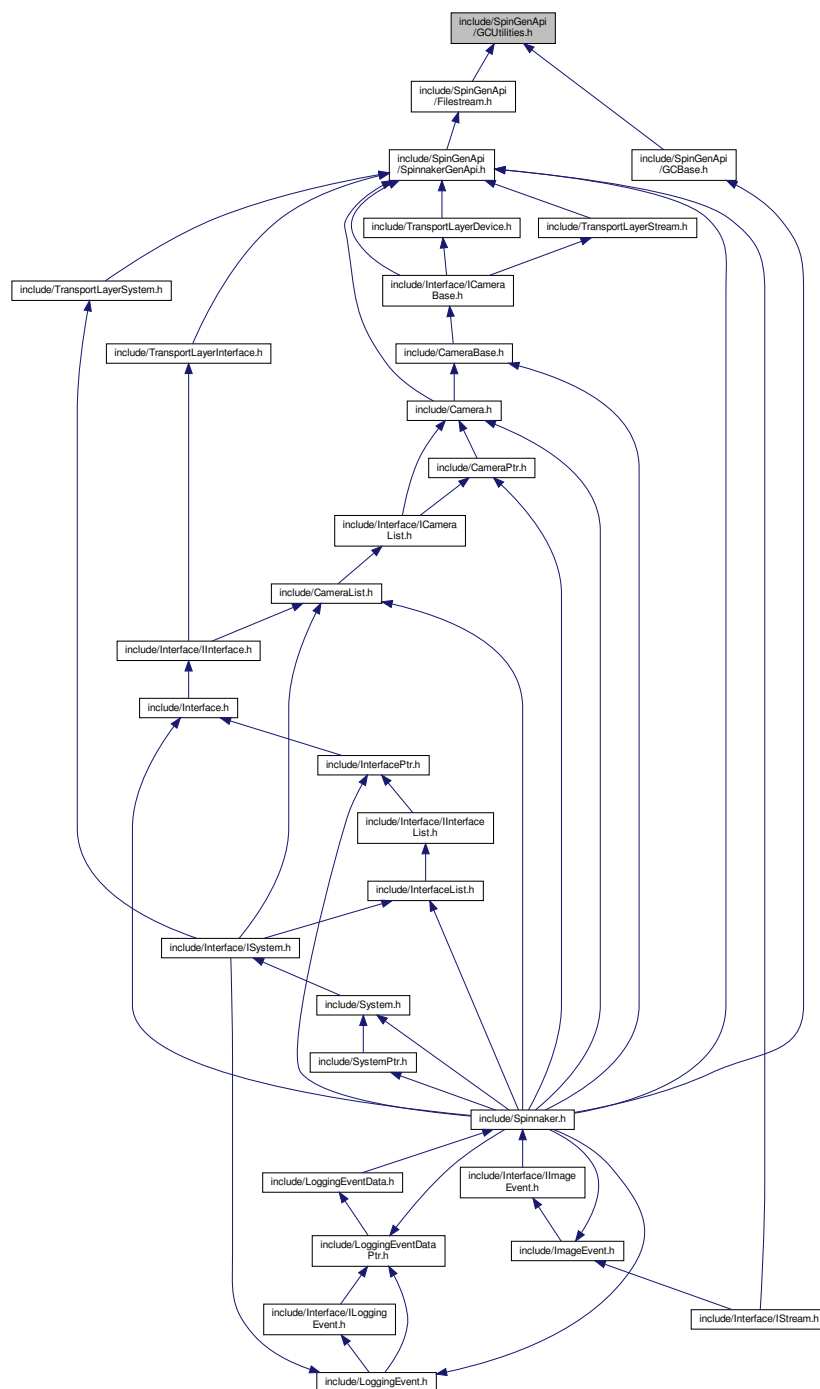
```
#define GC_UINT8_MAX static_cast<uint64_t>(0x00000000000000ffULL) /* maximum unsigned int8 value */
```

11.77 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.

11.77.1 Macro Definition Documentation

11.77.1.1 __ERR__

```
#define __ERR__ \_\_OUTPUT\_FORMATER\_\_("ERROR")
```

11.77.1.2 __LINE_STR__

```
#define __LINE_STR__ EXPAND\_TO\_STRINGISE(__LINE__)
```

11.77.1.3 __LOCATION__

```
#define __LOCATION__ __FILE__ "(" \_\_LINE\_STR\_\_ ")"
```

11.77.1.4 __OUTPUT_FORMATER__

```
#define __OUTPUT_FORMATER__(  
    _type ) \_\_LOCATION\_\_ " : " _type " : "
```

11.77.1.5 __TODO__

```
#define __TODO__ \_\_OUTPUT\_FORMATER\_\_("TBD")
```

11.77.1.6 __WARN__

```
#define __WARN__ \_\_OUTPUT\_FORMATER\_\_("WARNING")
```

11.77.1.7 `_TO_STRING`

```
#define _TO_STRING(  
    __stN ) #__stN
```

11.77.1.8 `EXPAND_TO_STRINGISE`

```
#define EXPAND_TO_STRINGISE(  
    __stN ) _TO_STRING( __stN )
```

11.77.1.9 `GC_COUNTOF`

```
#define GC_COUNTOF(  
    arr ) (sizeof (arr) / sizeof (arr)[0] )
```

11.77.1.10 `GENICAM_DEPRECATED`

```
#define GENICAM_DEPRECATED(  
    FUNCTION ) FUNCTION
```

11.77.1.11 `GENICAM_UNUSED`

```
#define GENICAM_UNUSED(  
    unused_var ) ((void)(unused_var))
```

11.77.1.12 `USE_TEMP_CACHE_FILE` [1/2]

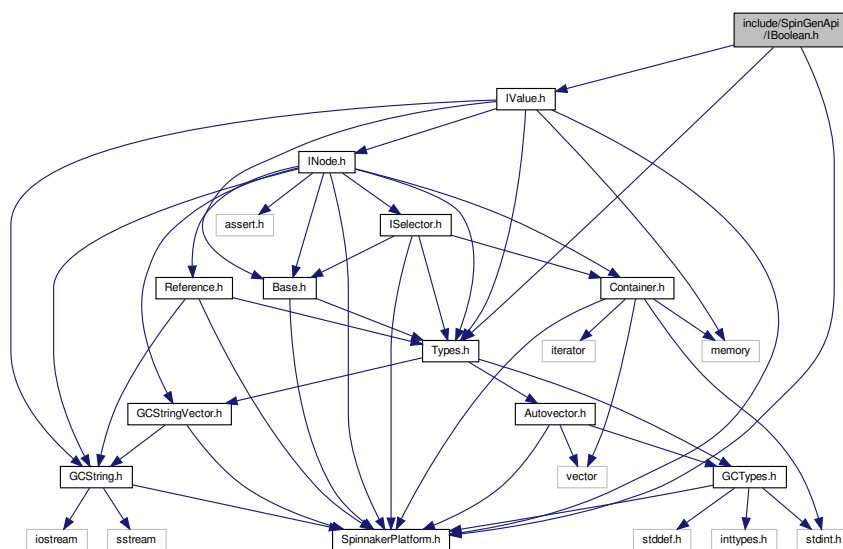
```
#define USE_TEMP_CACHE_FILE 1
```

11.77.1.13 `USE_TEMP_CACHE_FILE` [2/2]

```
#define USE_TEMP_CACHE_FILE 1
```

11.78 include/SpinGenApi/IBoolean.h File Reference

Include dependency graph for IBoolean.h:



Variables

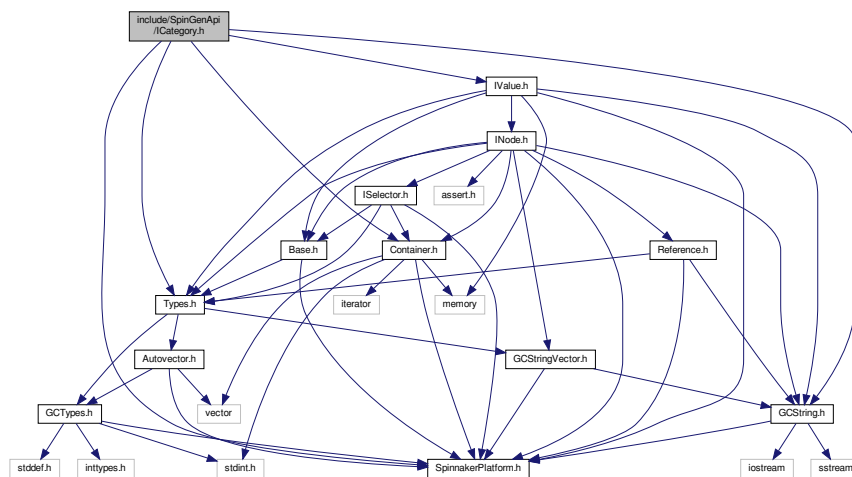
- [interface SPINNAKER_API_ABSTRACT IBoolean](#)

Interface for Boolean properties.

- [interface SPINNAKER_API_ABSTRACT](#) bool [Verify](#) = true) = 0

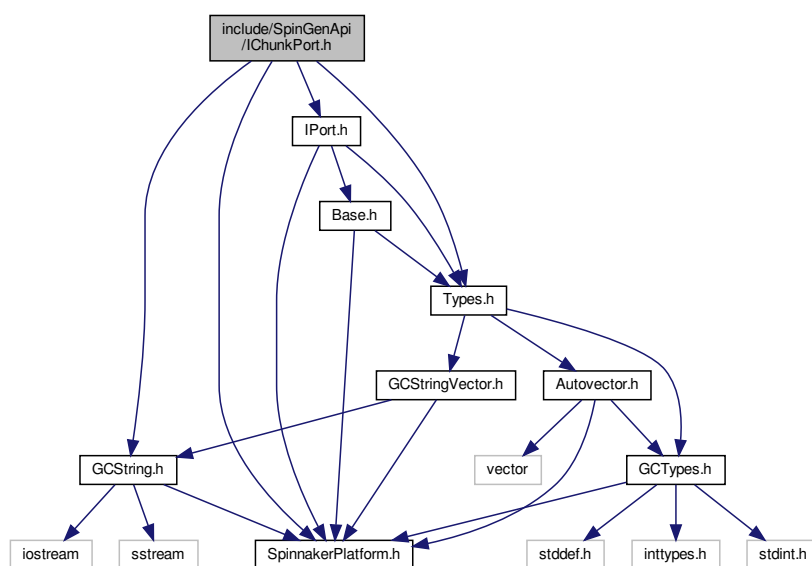
11.79 include/SpinGenApi/ICategory.h File Reference

Include dependency graph for ICategory.h:



11.80 include/SpinGenApi/IChunkPort.h File Reference

Include dependency graph for IChunkPort.h:



Functions

- virtual EYesNo [CacheChunkData](#) () const =0

Indicates if the chunk a adapter must hold a cached version of the chunk data.

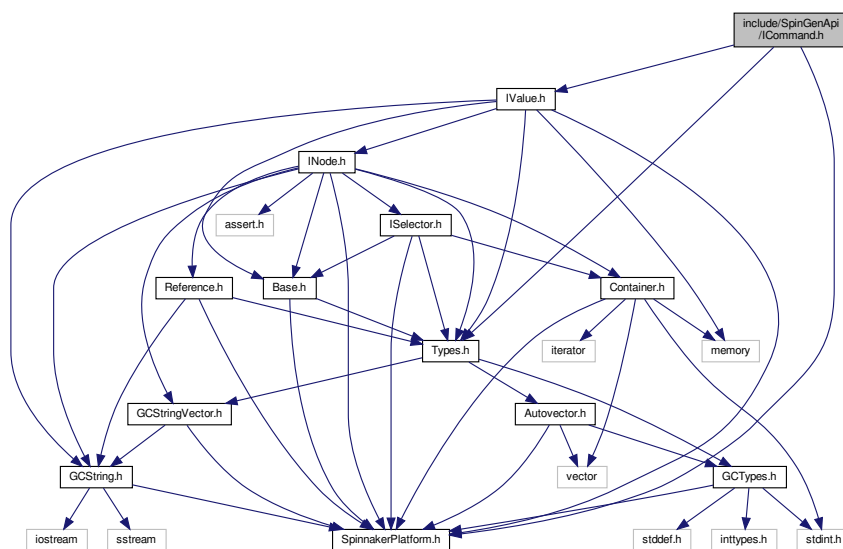
Variables

- [interface SPINNAKER_API_ABSTRACT IChunkPort](#)

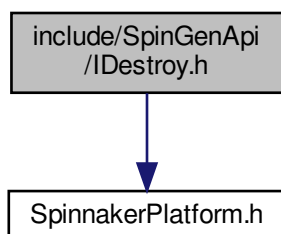
Interface for ports attached to a chunk.

11.81 include/SpinGenApi/ICommand.h File Reference

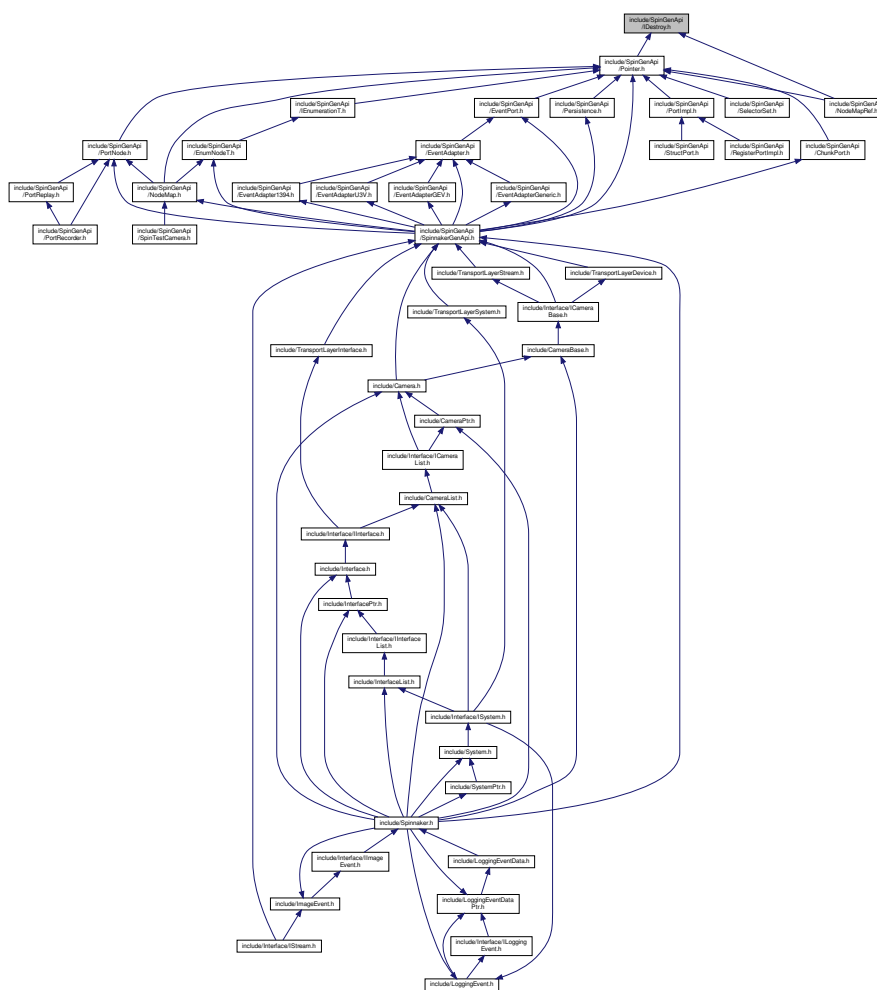
Include dependency graph for ICommand.h:



Include dependency graph for IDestroy.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

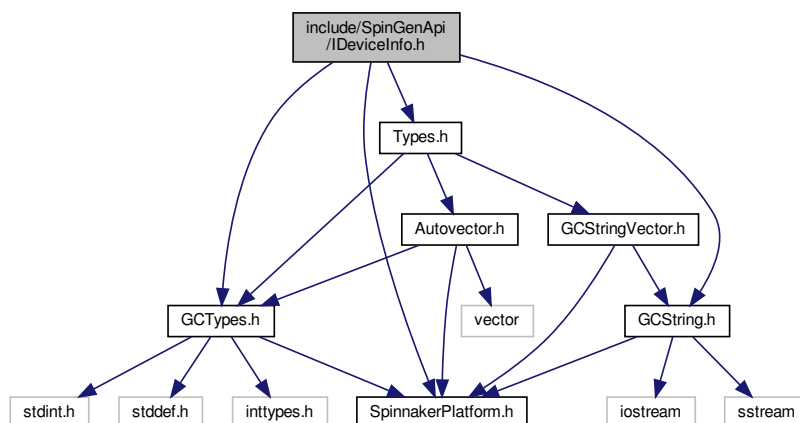
Variables

- [interface SPINNAKER_API_ABSTRACT IDestroy](#)

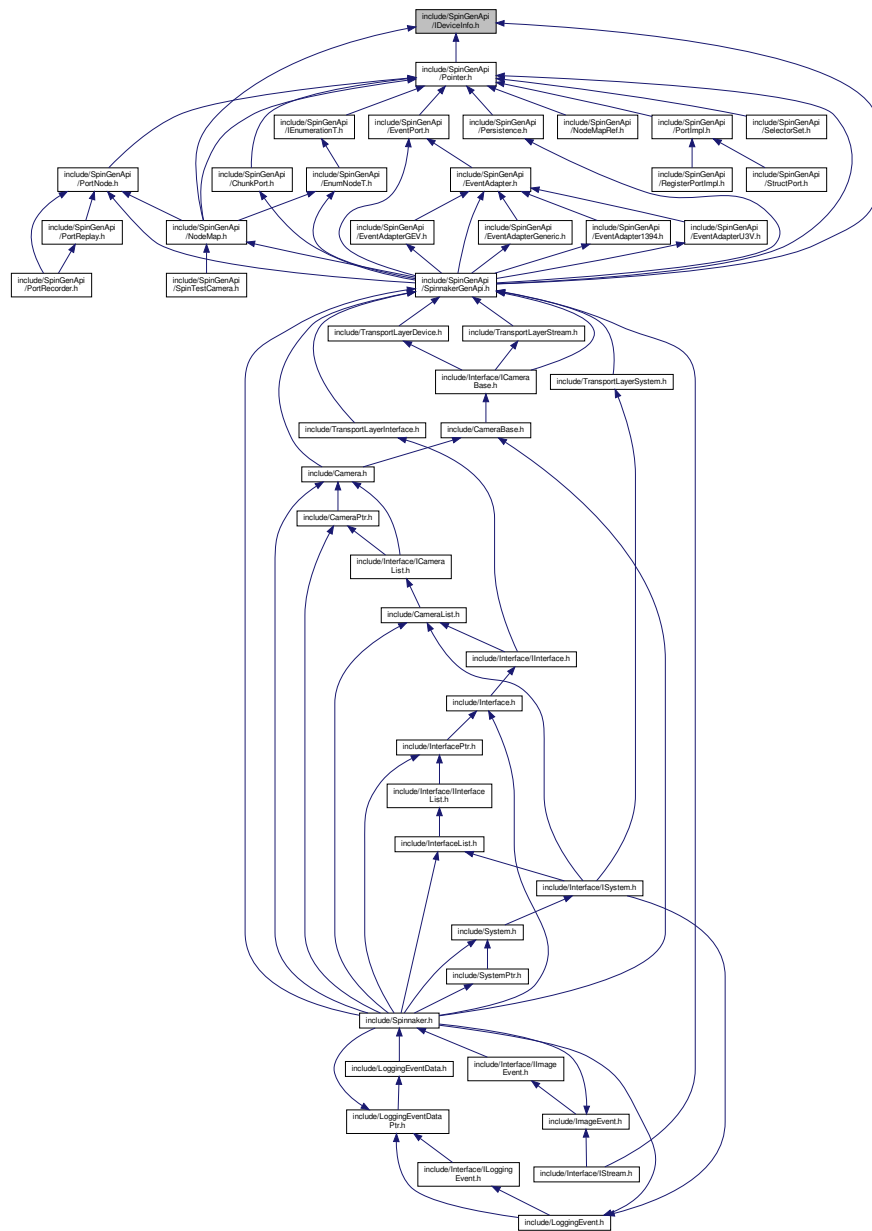
Interface to destroy an object.

11.83 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.

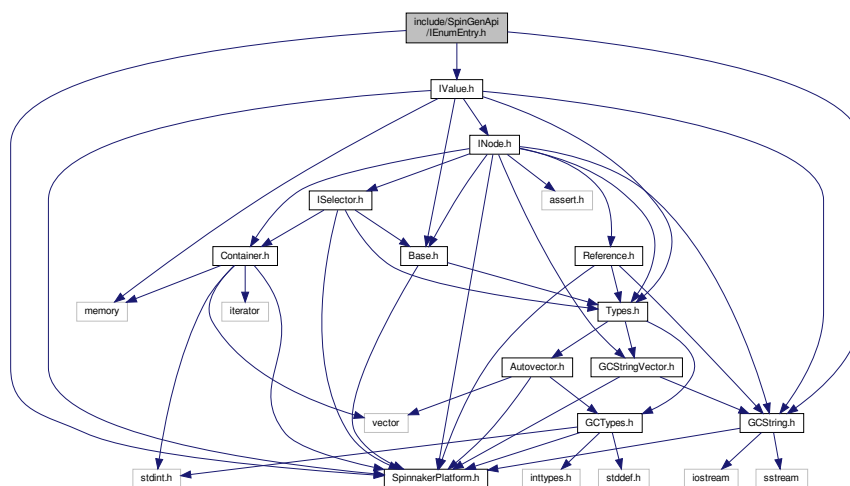
- virtual GenICam::gcstring [GetStandardNameSpace](#) ()=0
Get the standard name space.
- virtual void [GetGenApiVersion](#) (GenICam::Version_t &Version, uint16_t &Build)=0
Get the version of the DLL's [GenApi](#) implementation.
- virtual void [GetSchemaVersion](#) (GenICam::Version_t &Version)=0
Get 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)

11.84 include/SpinGenApi/IEnumEntry.h File Reference

Include dependency graph for IEnumEntry.h:



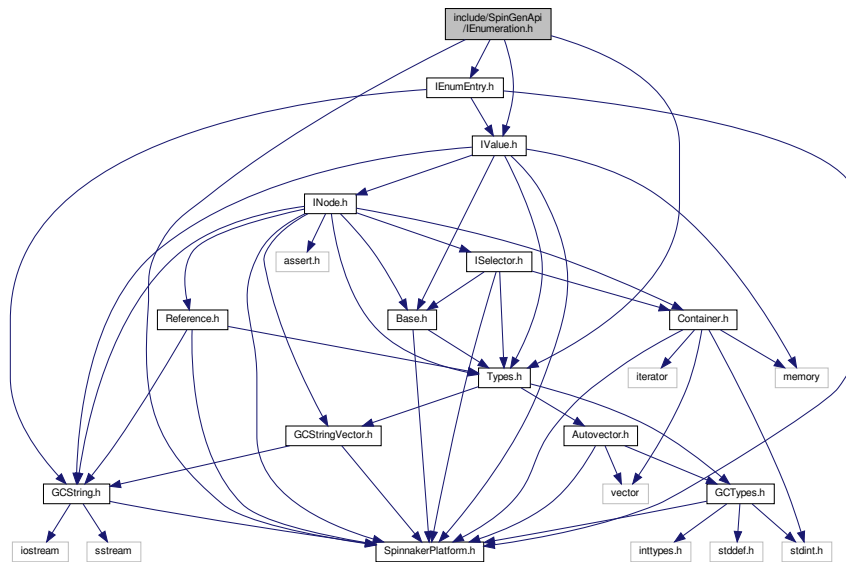
- Spinnaker
- Spinnaker::GenApi

- virtual GenlCam::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.

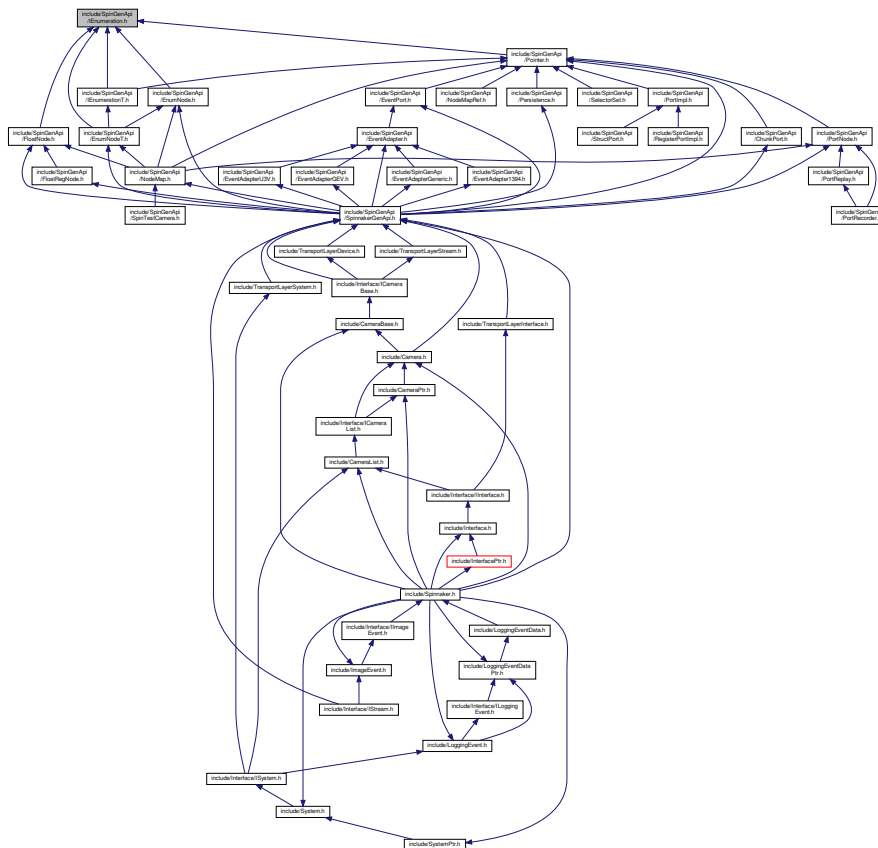
- `interface SPINNAKER_API_ABSTRACT IEnumEntry`
Interface of single enum value.

11.85 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

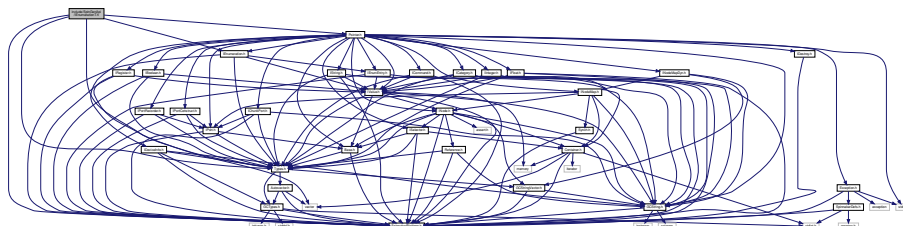
- virtual void [GetEntries](#) (NodeList_t &Entries)=0
Get list of entry nodes.
- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0
Set string node value.
- virtual void [SetIntValue](#) (int64_t Value, bool Verify=true)=0
Set integer node value.
- virtual GenICam::gcstring [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.

11.86 include/SpinGenApi/IEnumerationT.h File Reference

Include dependency graph for IEnumerationT.h:



Set node value.

- virtual bool [GetValue](#) (bool Verify=false, bool IgnoreCache=false) const =0

Get node value.

- virtual bool [operator\(\)](#) () const

Get node value.

- virtual IEnumeration & [operator=](#) (const GenICam::gcstring &ValueStr)=0

Set string node value.

- virtual IEnumEntry * [GetEntry](#) (const EnumT Value)=0

returns the EnumEntry object belonging to the Value

- virtual IEnumEntry * [GetCurrentEntry](#) (bool Verify=false, bool IgnoreCache=false)=0

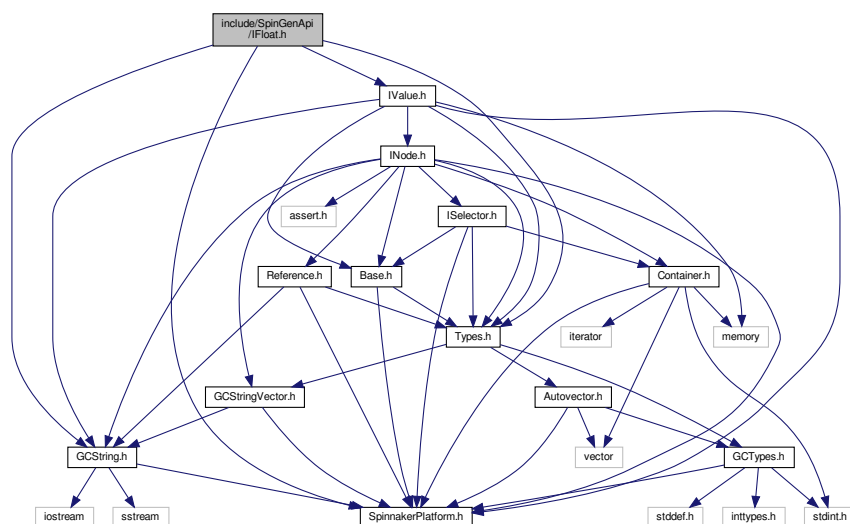
Get the current entry.

Variables

- template<typename EnumT >
[interface SPINNAKER_API_ABSTRACT IEnumerationT](#)
Interface for enumeration properties.
- template<typename EnumT >
[interface SPINNAKER_API_ABSTRACT](#) virtual public [IEnumReference](#)
Interface to construct an enum reference.

11.87 include/SpinGenApi/IFloat.h File Reference

Include dependency graph for IFloat.h:



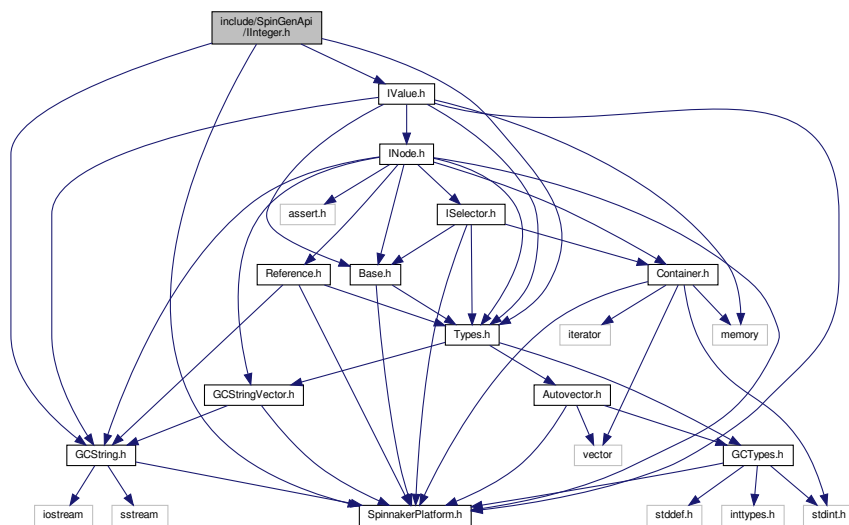
- virtual double [GetInc](#) ()=0
Get the constant increment if there is any.
- virtual double_autovector_t [GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual ERepresentation [GetRepresentation](#) ()=0
Get recommended representation.
- virtual GenICam::gcstring [GetUnit](#) () 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.

11.88 include/SpinGenApi/IInteger.h File Reference

Include dependency graph for IInteger.h:



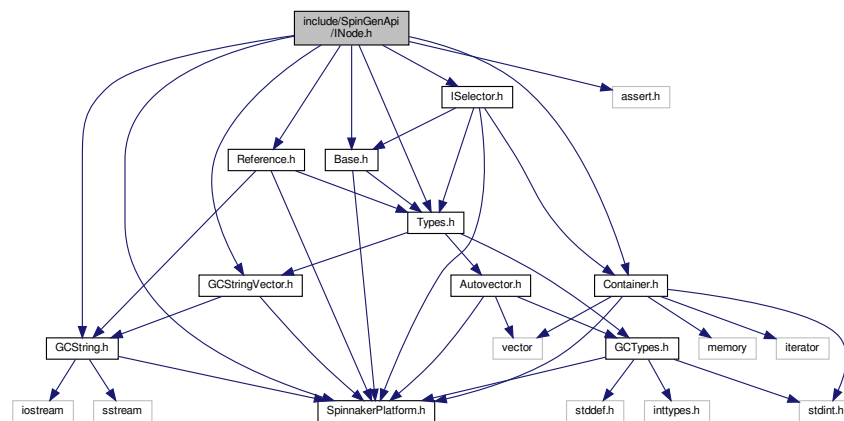
- virtual double_autovector_t [GetListOfValidValues](#) (bool bounded=true)=0
Get list of valid value.
- virtual ERepresentation [GetRepresentation](#) ()=0
Get recommended representation.
- virtual GenICam::gcstring [GetUnit](#) () const =0
Get the physical unit name.
- virtual void [ImposeMin](#) (int64_t Value)=0
Restrict minimum value.
- virtual void [ImposeMax](#) (int64_t Value)=0
Restrict maximum value.

Variables

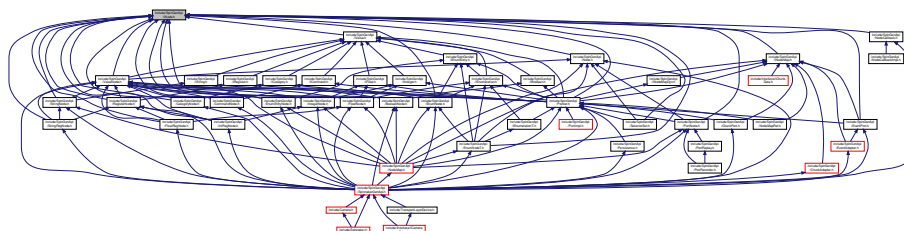
- [interface SPINNAKER_API_ABSTRACT Integer](#)
Interface for integer properties.

11.89 include/SpinGenApi/INode.h File Reference

Include dependency graph for INode.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef node_vector [NodeList_t](#)
a list of node references
- typedef intptr_t [CallbackHandleType](#)
the callback handle for nodes

Functions

- virtual GenApi::ENamespace [GetNameSpace](#) () const =0
Get name space.
- virtual EVisibility [GetVisibility](#) () const =0
Get the recommended visibility of the node.
- virtual void [InvalidateNode](#) ()=0
Indicates that the node's value may have changed.
- virtual bool [IsCacheable](#) () const =0
Is the node value cacheable.
- virtual EYesNo [IsAccessModeCacheable](#) () const =0
True if the AccessMode can be cached.
- virtual ECachingMode [GetCachingMode](#) () const =0
Get Caching Mode.
- virtual int64_t [GetPollingTime](#) () const =0
recommended polling time (for non-cacheable nodes)
- virtual GenICam::gcstring [GetToolTip](#) ()=0
Get tool tip.
- virtual GenICam::gcstring [GetDescription](#) () const =0
Get a long description of the node.
- virtual GenICam::gcstring [GetDisplayName](#) () const =0
Get a name string for display.
- virtual GenICam::gcstring [GetDeviceName](#) () const =0
Get a name of the device.
- virtual void [GetChildren](#) (GenApi::NodeList_t &Children, ELinkType LinkType=ctReadingChildren) const =0
Get all nodes this node directly depends on.
- virtual void [GetParents](#) (GenApi::NodeList_t &Parents) const =0
Gets all nodes this node is directly depending on.
- virtual CallbackHandleType [RegisterCallback](#) (CNodeCallback *pCallback)=0
Register change callback Takes ownership of the CNodeCallback object.
- virtual bool [DeregisterCallback](#) (CallbackHandleType hCallback)=0
De register change callback Destroys CNodeCallback object.
- virtual INodeMap * [GetNodeMap](#) () const =0
Retrieves the central node map.
- virtual GenICam::gcstring [GetEventID](#) () const =0
Get the EventId of the node.
- virtual bool [IsStreamable](#) () const =0
True if the node is streamable.

- virtual void [GetPropertyNames](#) (GenICam::gcstring_vector &PropertyNames) const =0
Returns a list of the names all properties set during initialization.
- virtual bool [GetProperty](#) (const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)=0
Retrieves a property plus an additional attribute by name. If a property has multiple values/attribute they come with Tabs as delimiters.
- virtual void [ImposeAccessMode](#) (EAccessMode ImposedAccessMode)=0
Imposes an access mode to the natural access mode of the node.
- virtual void [ImposeVisibility](#) (EVisibility ImposedVisibility)=0
Imposes a visibility to the natural visibility of the node.
- virtual INode * [GetAlias](#) () const =0
Retrieves the a node which describes the same feature in a different way.
- virtual INode * [GetCastAlias](#) () const =0
Retrieves the a node which describes the same feature so that it can be casted.
- virtual GenICam::gcstring [GetDocuURL](#) () const =0
Gets a URL pointing to the documentation of that feature.
- virtual bool [IsDeprecated](#) () const =0
True if the node should not be used any more.
- virtual EInterfaceType [GetPrincipalInterfaceType](#) () const =0
Get the type of the main interface of a node.
- virtual bool [IsFeature](#) () const =0
True if the node can be reached via category nodes from a category node named "Root".
- virtual bool [operator==](#) (int nullPtr) const =0
- virtual bool [operator!=](#) (int nullPtr) const =0
- bool [IsReadable](#) (EAccessMode AccessMode)
Tests if readable.
- bool [IsReadable](#) (const IBase *p)
Checks if a node is readable.
- bool [IsReadable](#) (const IBase &r)
Checks if a node is readable.
- bool [IsWritable](#) (EAccessMode AccessMode)
Tests if writable.
- bool [IsWritable](#) (const IBase *p)
Checks if a node is writable.
- bool [IsWritable](#) (const IBase &r)
Checks if a node is writable.
- bool [IsImplemented](#) (EAccessMode AccessMode)
Tests if implemented.
- bool [IsImplemented](#) (const IBase *p)
Checks if a node is implemented.
- bool [IsImplemented](#) (const IBase &r)
Checks if a node is implemented.
- bool [IsAvailable](#) (EAccessMode AccessMode)
Tests if available.
- bool [IsAvailable](#) (const IBase *p)
Checks if a node is available.
- bool [IsAvailable](#) (const IBase &r)
Checks if a node is available.
- EAccessMode [Combine](#) (EAccessMode Peter, EAccessMode Paul)
Computes which access mode the two guards allow together.
- bool [IsVisible](#) (EVisibility Visibility, EVisibility MaxVisibility)

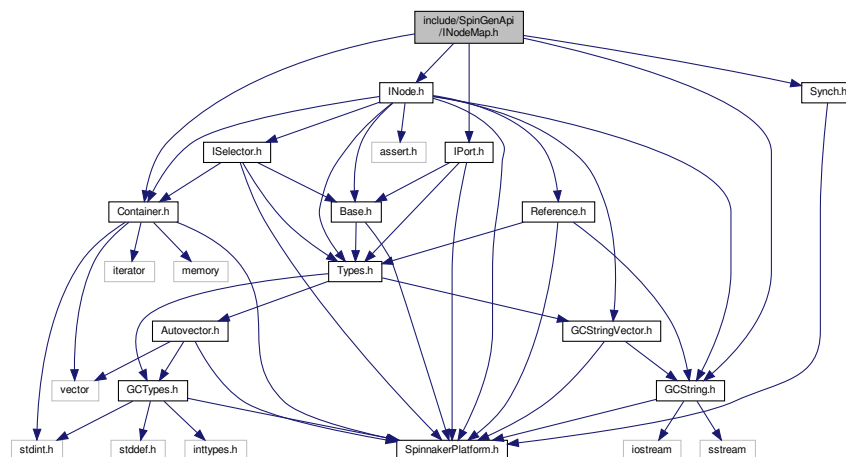
- Tests Visibility CAVE : this relies on the EVisibility enum's coding.
- EVisibility [Combine](#) (EVisibility Peter, EVisibility Paul)
Computes which visibility the two guards allow together.
- bool [IsCacheable](#) (ECachingMode CachingMode)
Tests Cacheability.
- ECachingMode [Combine](#) (ECachingMode Peter, ECachingMode Paul)
Computes which CachingMode results from a combination.

Variables

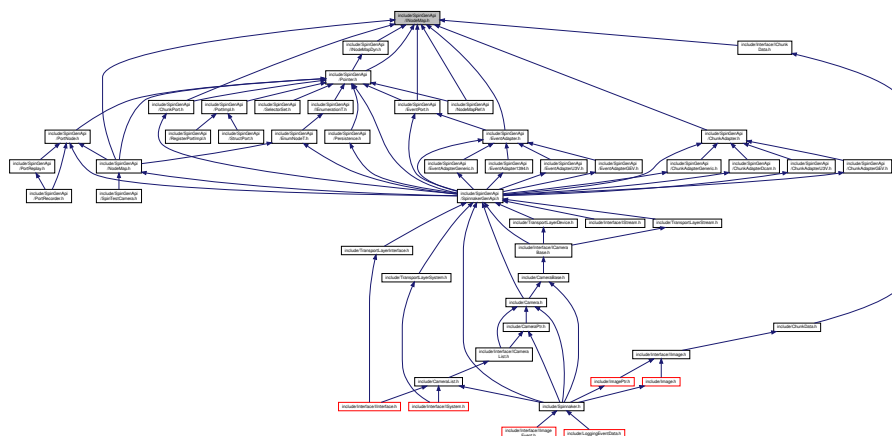
- [interface SPINNAKER_API_ABSTRACT INode](#)
Interface common to all nodes.
- [interface SPINNAKER_API_ABSTRACT virtual public IReference](#)
Interface to construct a reference.

11.90 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

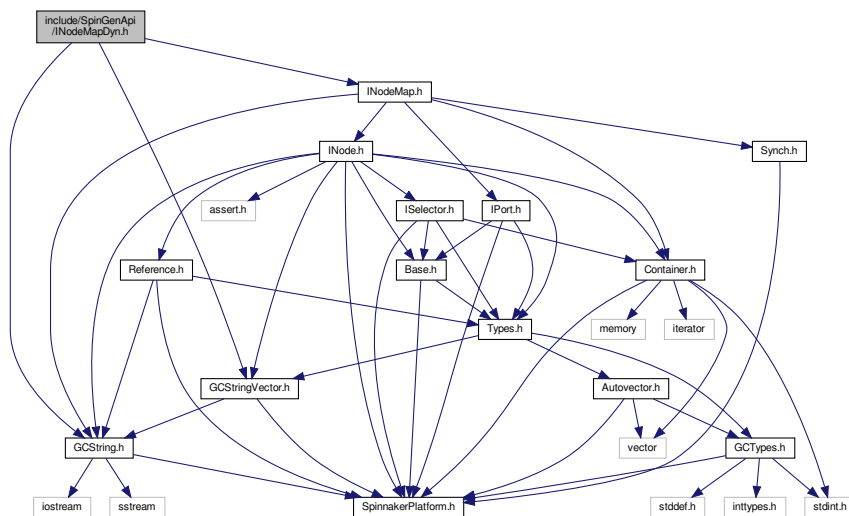
- virtual `INode *` [GetNode](#) (`const GenICam::gcstring &Name`) `const =0`
Retrieves the node from the central map by Name.
- virtual void [InvalidateNodes](#) () `const =0`
Invalidates all nodes.
- virtual bool [Connect](#) (`IPort *pPort`, `const GenICam::gcstring &PortName`) `const =0`
Connects 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.

11.91 include/SpinGenApi/INodeMapDyn.h File Reference

Include dependency graph for INodeMapDyn.h:



- ## Variables

- ## 11.92 include/SpinGenApi/IntegerNode.h File Reference

[illegible]

Namespaces

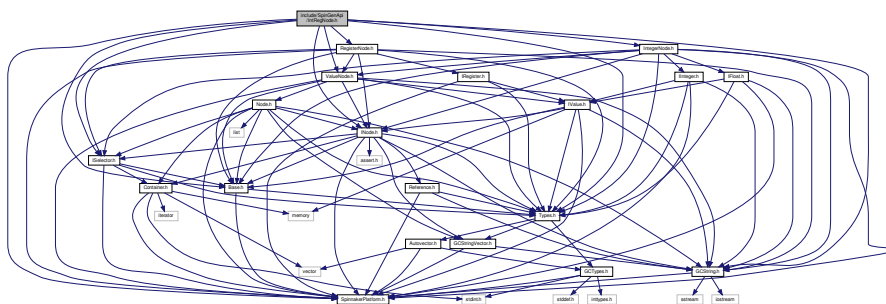
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

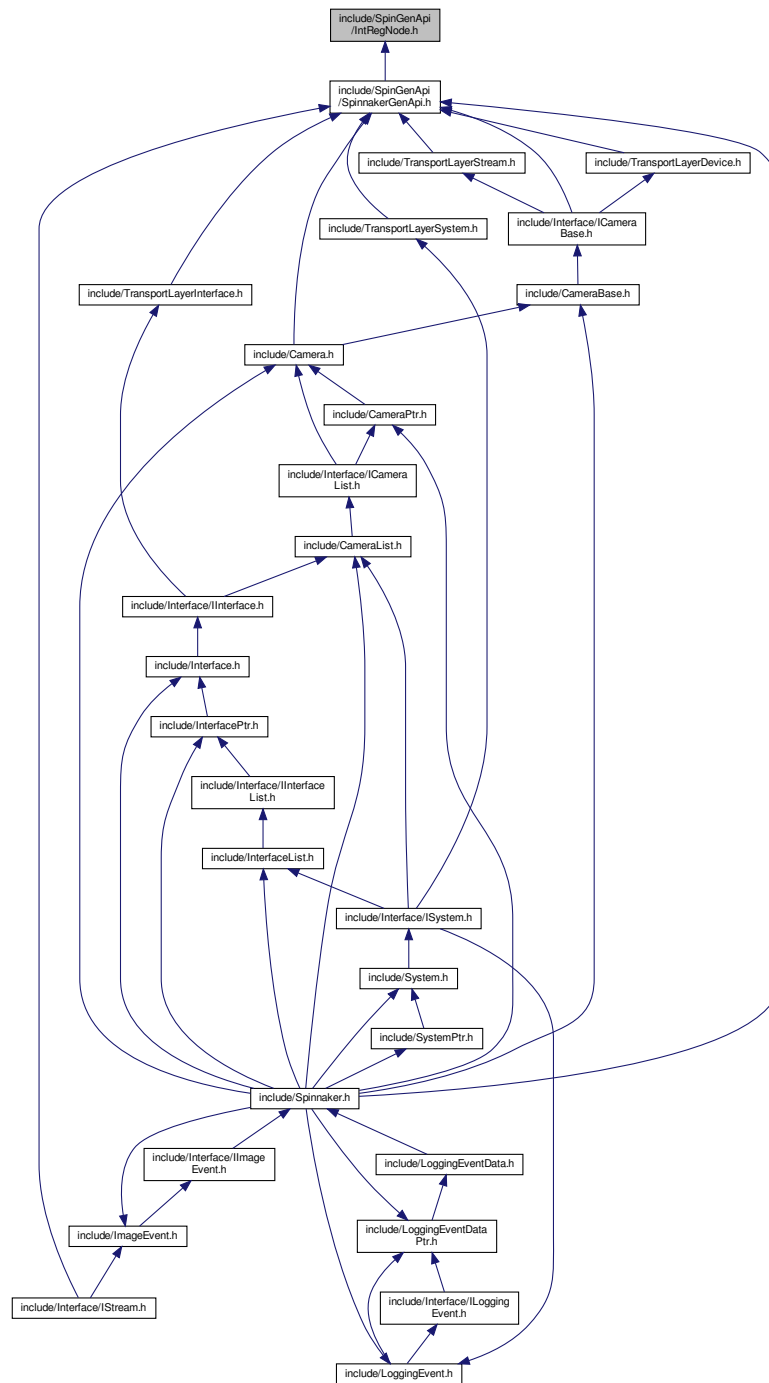
- typedef IntegerNode [CIntegerRef](#)

11.93 include/SpinGenApi/IntRegNode.h File Reference

Include dependency graph for IntRegNode.h:



This graph shows which files directly or indirectly include this file:



Classes

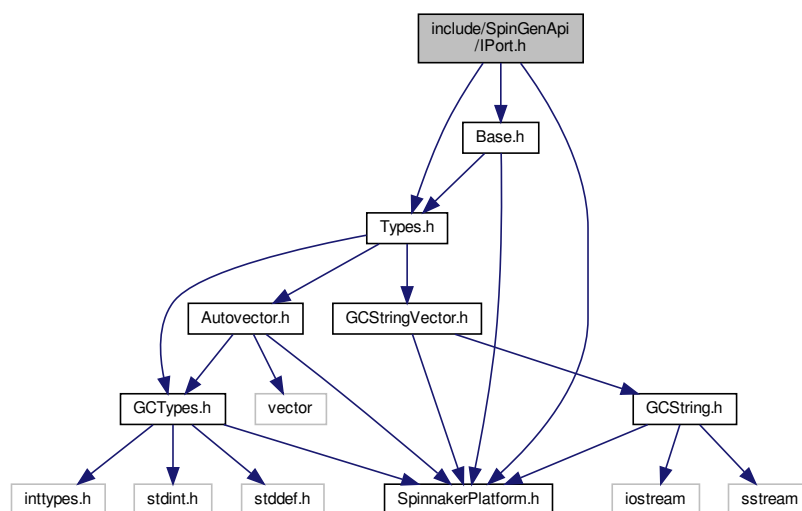
- class [IntRegNode](#)
Interface for string properties.

Namespaces

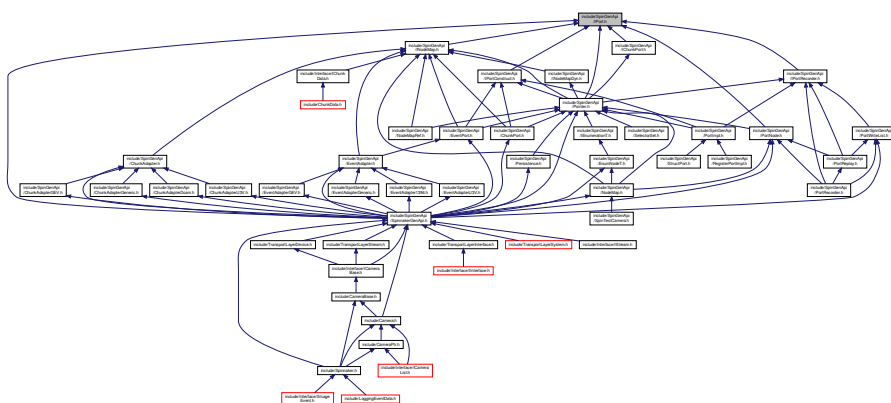
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.94 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

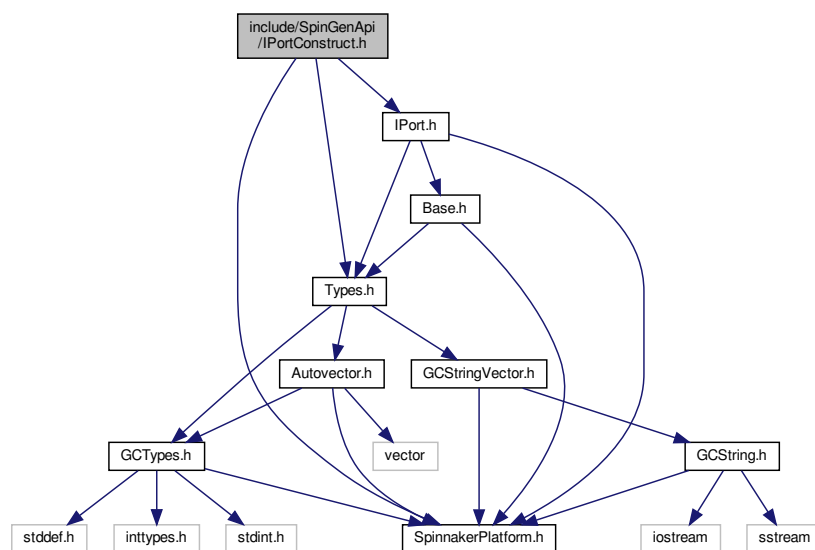
- [interface SPINNAKER_API_ABSTRACT IPort](#)

Interface for ports.

- [interface SPINNAKER_API_ABSTRACT int64_t Address](#)
- [interface SPINNAKER_API_ABSTRACT int64_t int64_t Length = 0](#)

11.95 include/SpinGenApi/IPortConstruct.h File Reference

Include dependency graph for IPortConstruct.h:



[illegible]

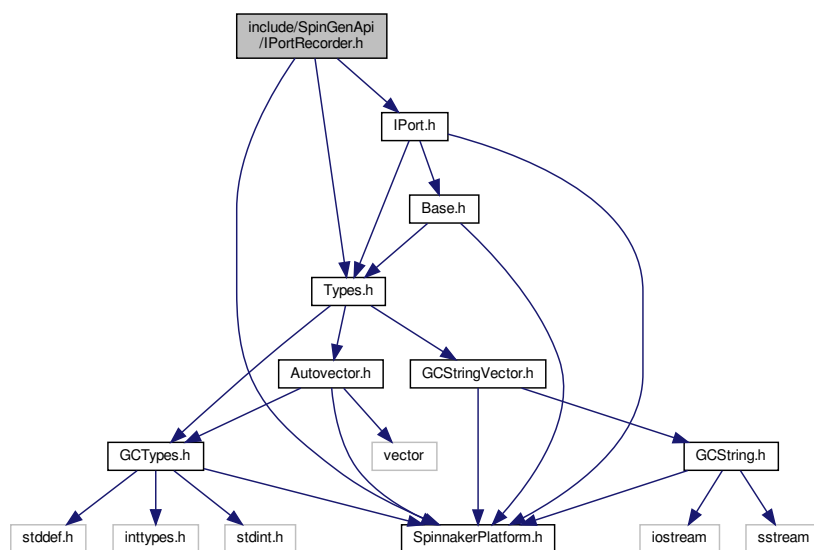
- Spinnaker
- Spinnaker::GenApi

- virtual EYesNo `GetSwapEndianness` ()=0
Determines if the port adapter must perform an endianness swap.

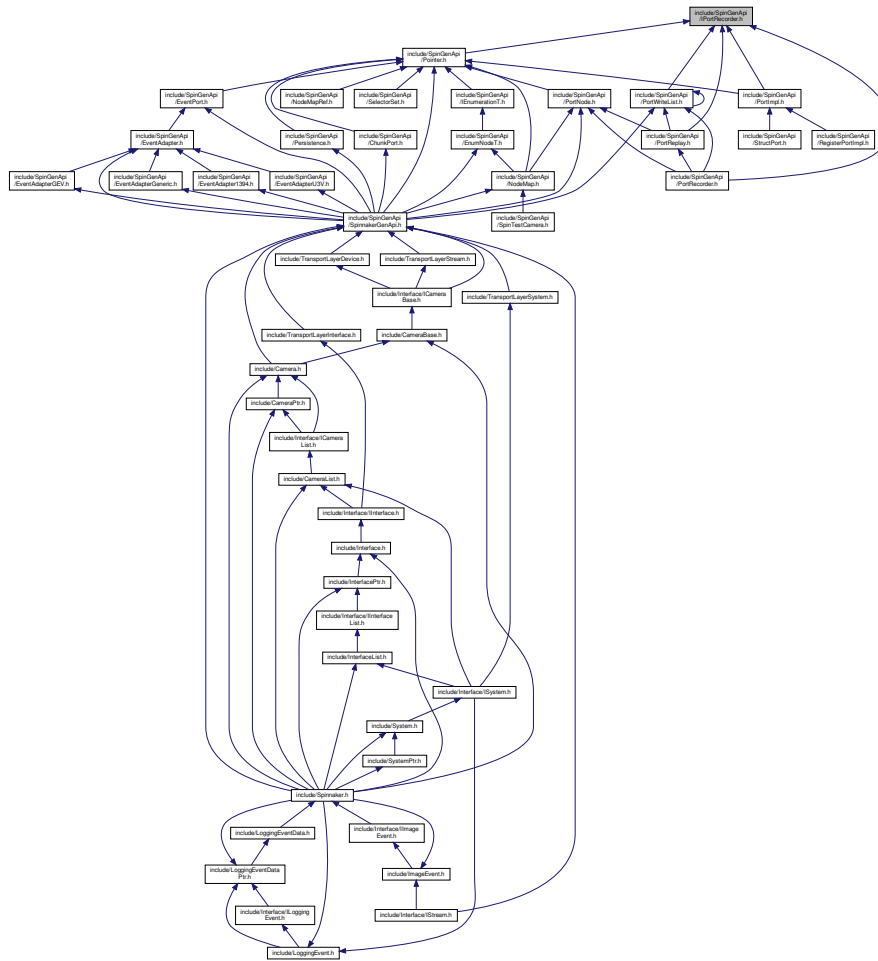
- interface SPINNAKER_API IPortConstruct
Interface for ports.

11.96 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](#)

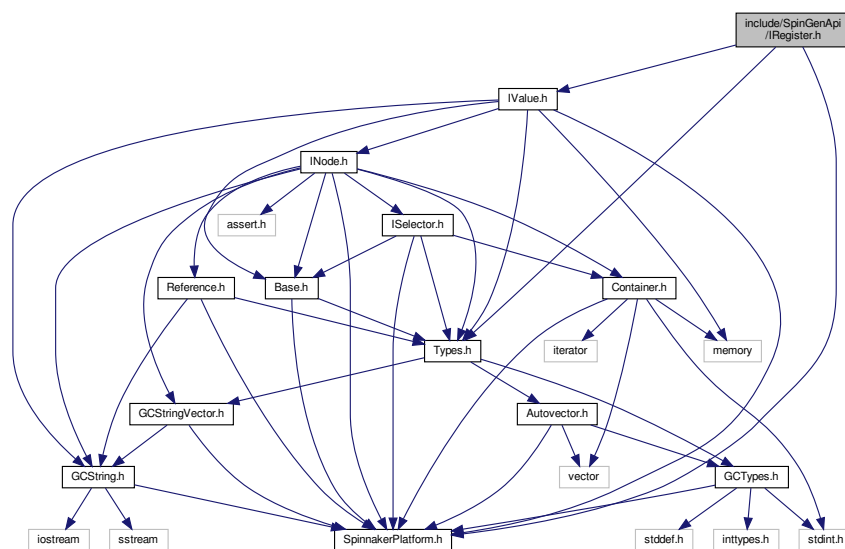
Interface for replaying write commands on a port.

- [interface SPINNAKER_API_ABSTRACT bool Invalidate = true\) = 0](#)
- [interface SPINNAKER_API_ABSTRACT IPortRecorder](#)

Interface for recording write commands on a port.

11.97 include/SpinGenApi/IRegister.h File Reference

Include dependency graph for IRegister.h:



[illegible]

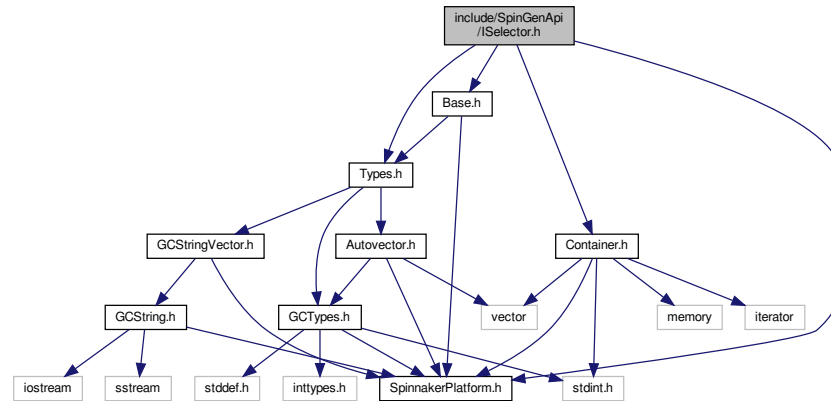
- Spinnaker
- Spinnaker::GenApi

- 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.

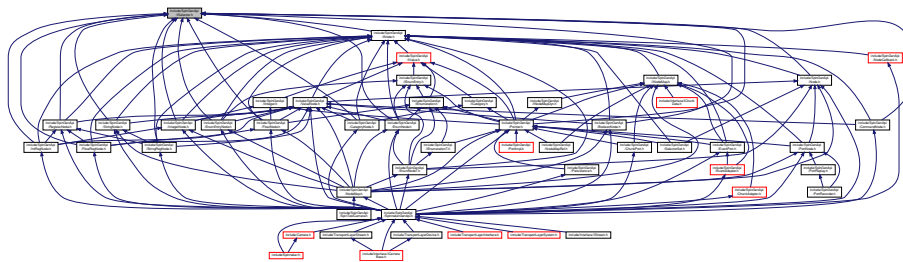
- `interface SPINNAKER_API_ABSTRACT IRegister`
Interface for registers.

11.98 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

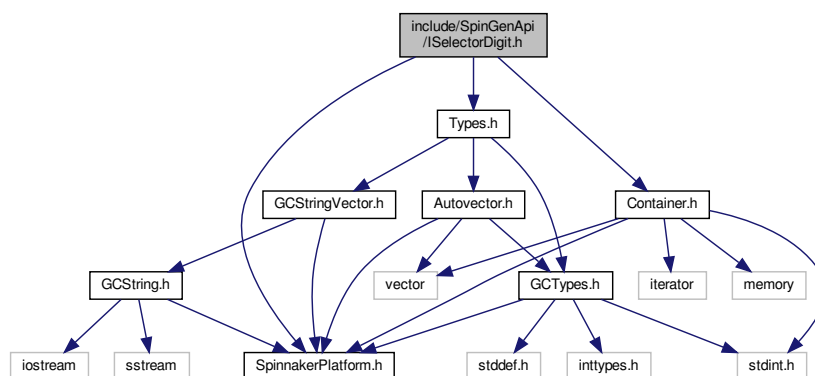
- virtual void [GetSelectedFeatures](#) (FeatureList_t &) const =0
retrieve the group of selected features
- virtual void [GetSelectingFeatures](#) (FeatureList_t &) const =0
retrieve the group of features selecting this node

Variables

- [interface SPINNAKER_API_ABSTRACT ISelector](#)
Interface for groups of features selected by a single one.

11.99 include/SpinGenApi/ISelectorDigit.h File Reference

Include dependency graph for ISelectorDigit.h:



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.

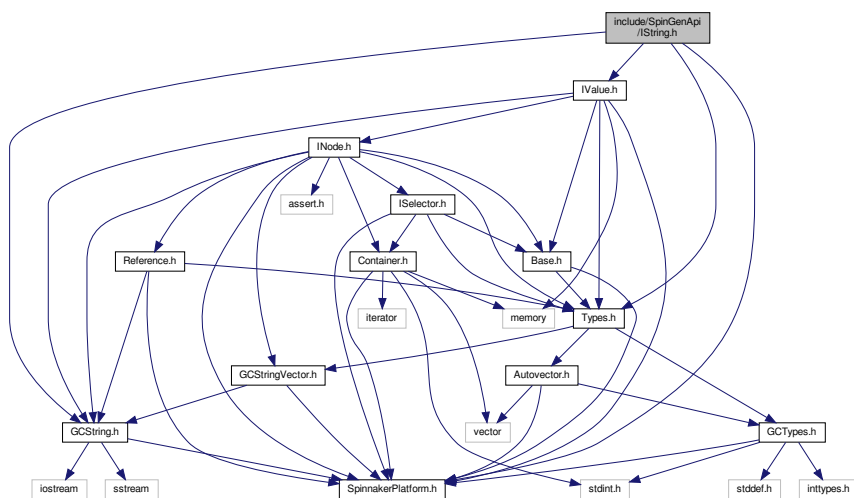
Variables

- [interface SPINNAKER_API_ABSTRACT ISelectorDigit](#)

Interface of a "digit" of the "counter" formed by the selector set.

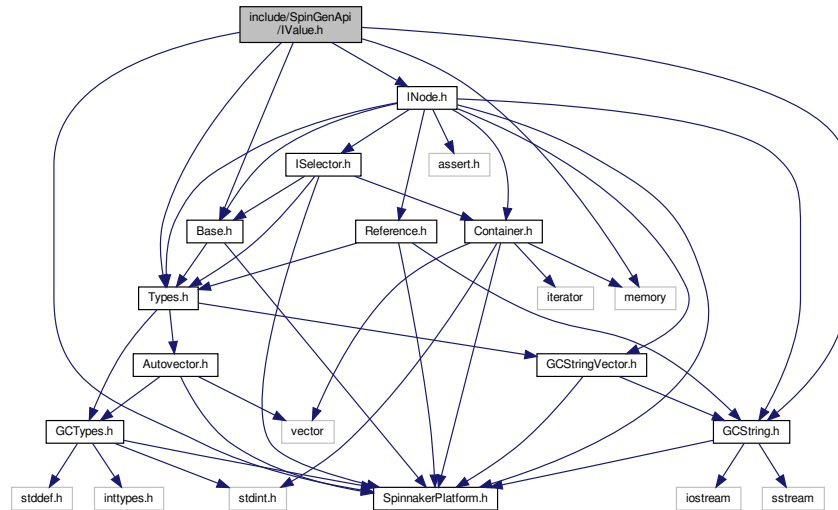
11.100 include/SpinGenApi/IString.h File Reference

Include dependency graph for IString.h:

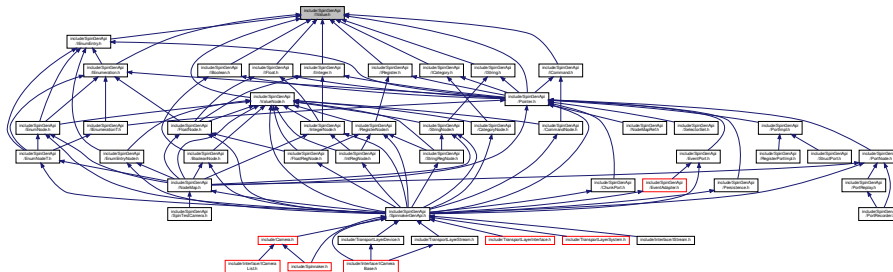


11.101 include/SpinGenApi/IValue.h File Reference

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.

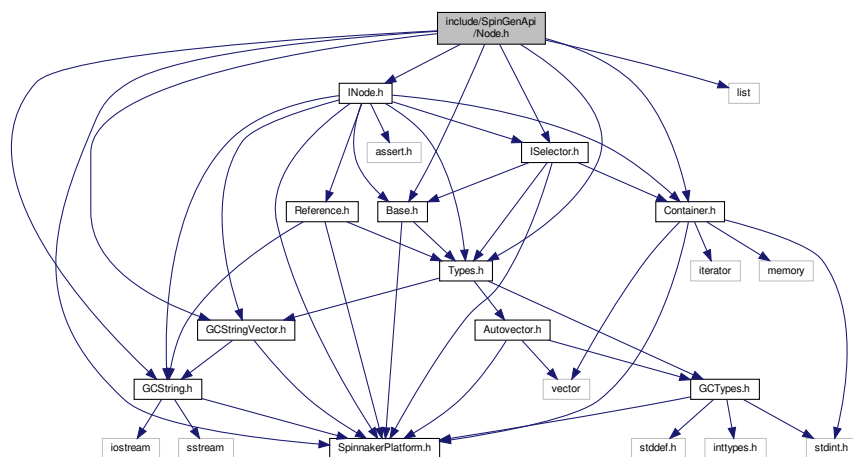
Variables

- [interface SPINNAKER_API_ABSTRACT IValue](#)

Interface for value properties.

11.102 include/SpinGenApi/Node.h File Reference

Include dependency graph for Node.h:



The diagram illustrates a complex dependency graph for a system, likely a sensor network or data processing application. The graph is organized into layers, with the most fundamental interfaces at the bottom and more complex, application-specific modules at the top.

Key Components and Dependencies:

- Base Interfaces (Bottom Layer):**
 - `include/LoggingEvent.h` is the base for `include/LoggingEventData.h` and `include/LoggingEventData.h`.
 - `include/LoggingEventData.h` is the base for `include/LoggingEventData.h` and `include/LoggingEventData.h`.
 - `include/LoggingEventData.h` is the base for `include/LoggingEventData.h` and `include/LoggingEventData.h`.
 - `include/LoggingEventData.h` is the base for `include/LoggingEventData.h` and `include/LoggingEventData.h`.
- System and Interface Layers:**
 - `include/System.h` and `include/SystemPr.h` are foundational for many other modules.
 - `include/Interface.h` and `include/InterfacePr.h` define the communication protocols.
 - `include/InterfaceCamera.h` and `include/InterfaceCameraPr.h` are specific to camera-related functionality.
 - `include/InterfaceImage.h` and `include/InterfaceImagePr.h` are specific to image-related functionality.
- Transport and Sensor Layers:**
 - `include/Transport.h` and `include/TransportPr.h` handle data transport.
 - `include/Sensor.h` and `include/SensorPr.h` define the sensor interfaces.
 - `include/SensorApi.h` is a central hub for sensor-related APIs.
- Application-Specific Modules (Top Layer):**
 - `include/SensorApi.h` is the central hub for sensor-related APIs.
 - `include/SensorApi.h` is the central hub for sensor-related APIs.
 - `include/SensorApi.h` is the central hub for sensor-related APIs.
 - `include/SensorApi.h` is the central hub for sensor-related APIs.

The graph shows a highly interconnected system where many modules depend on a small number of core interfaces and system components. This suggests a modular architecture where functionality is built up from a common base.

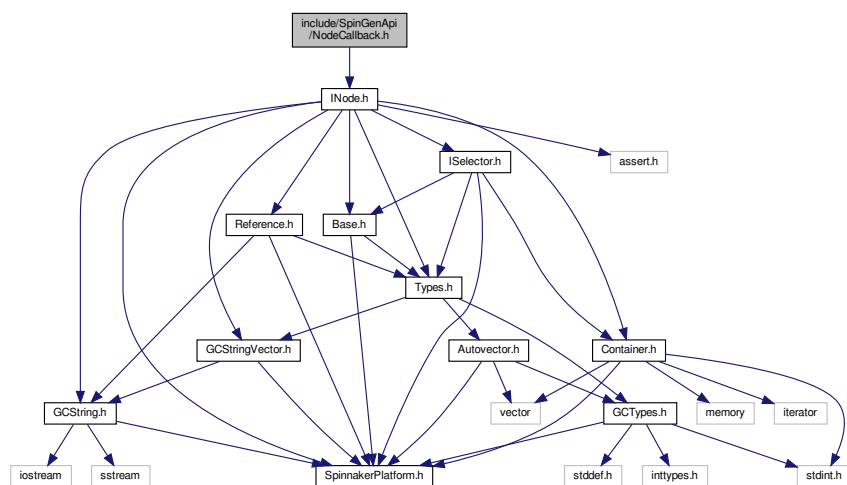
- class **Node**
class common to all nodes

- Spinnaker
- Spinnaker::GenApi

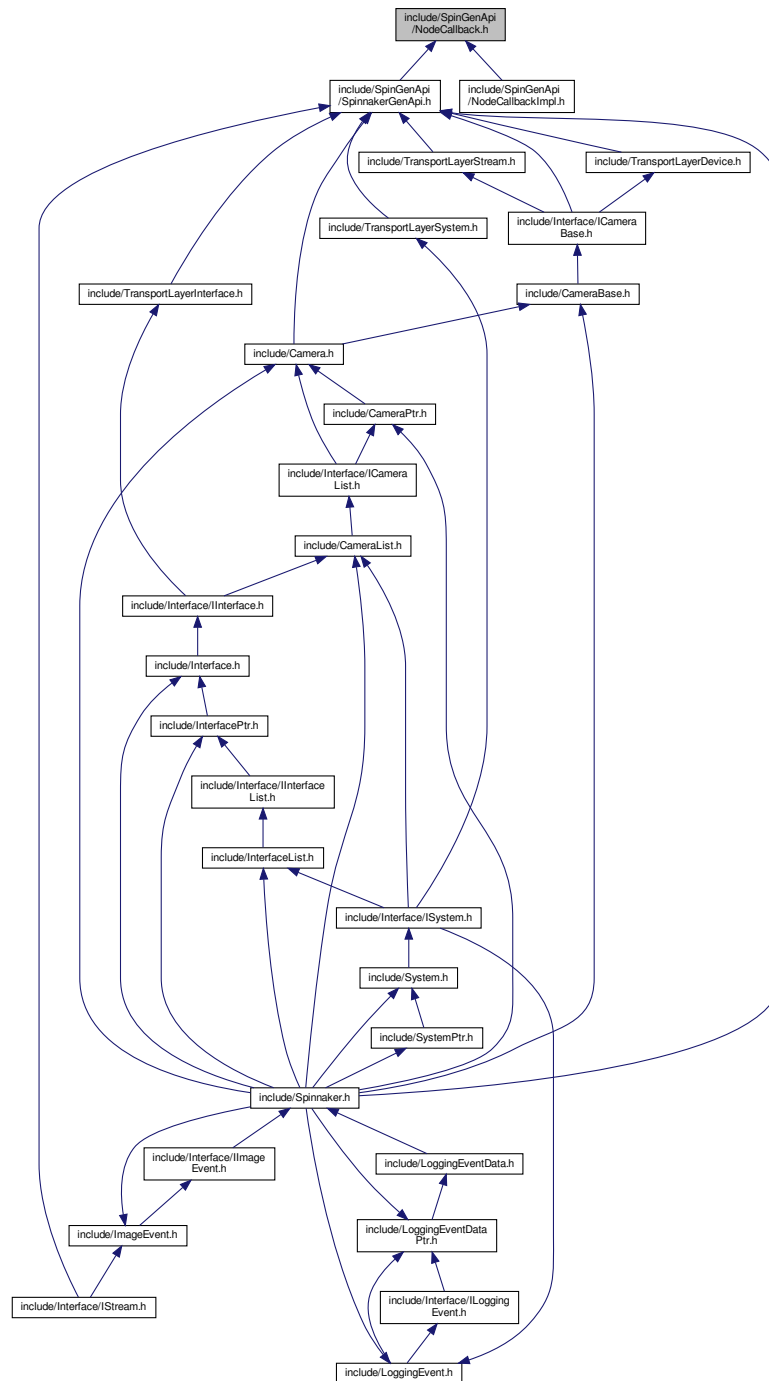
- typedef Node CNodeRef
- typedef Node CSelectorRef

11.103 include/SpinGenApi/NodeCallback.h File Reference

Include dependency graph for NodeCallback.h:



This graph shows which files directly or indirectly include this file:

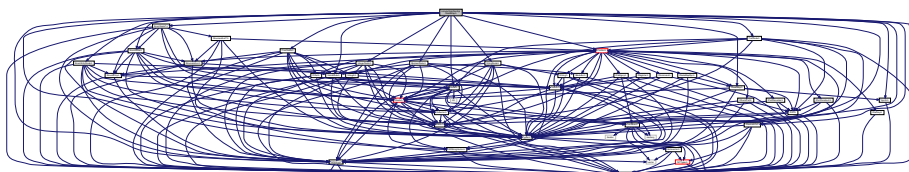


Classes

- 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.

11.105 include/SpinGenApi/NodeMap.h File Reference

Include dependency graph for NodeMap.h:



Namespaces

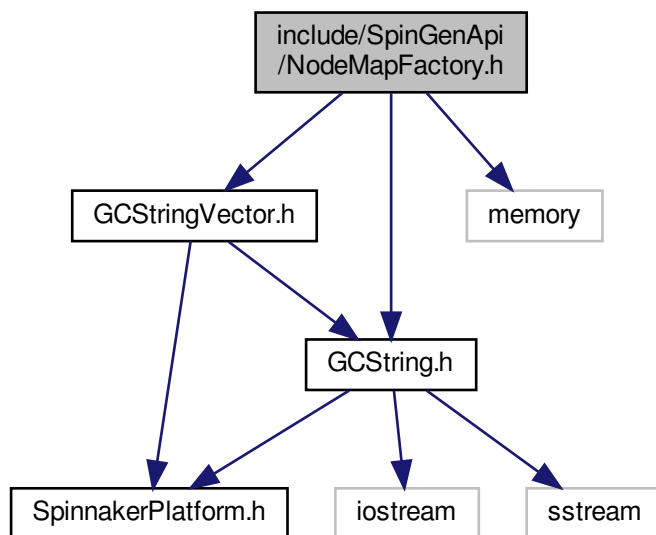
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

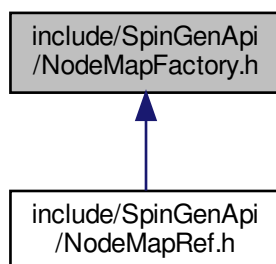
- typedef NodeMap [CNodeMapRef](#)

11.106 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

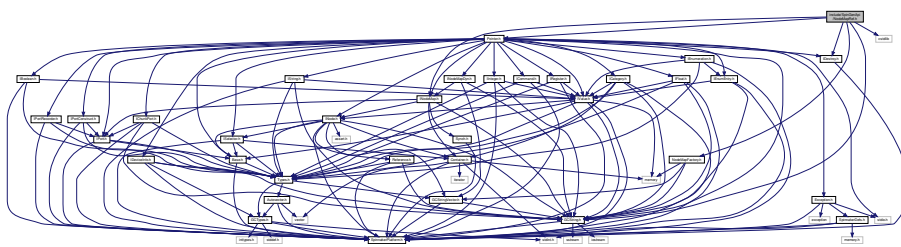
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

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.

11.107 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

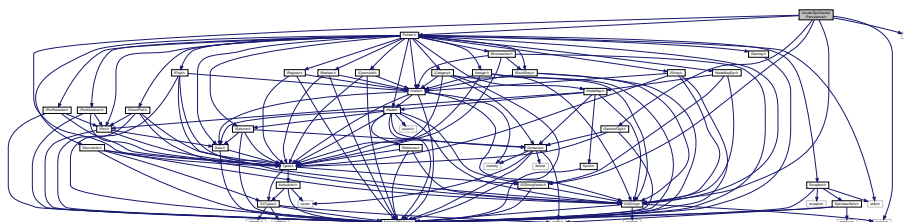
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

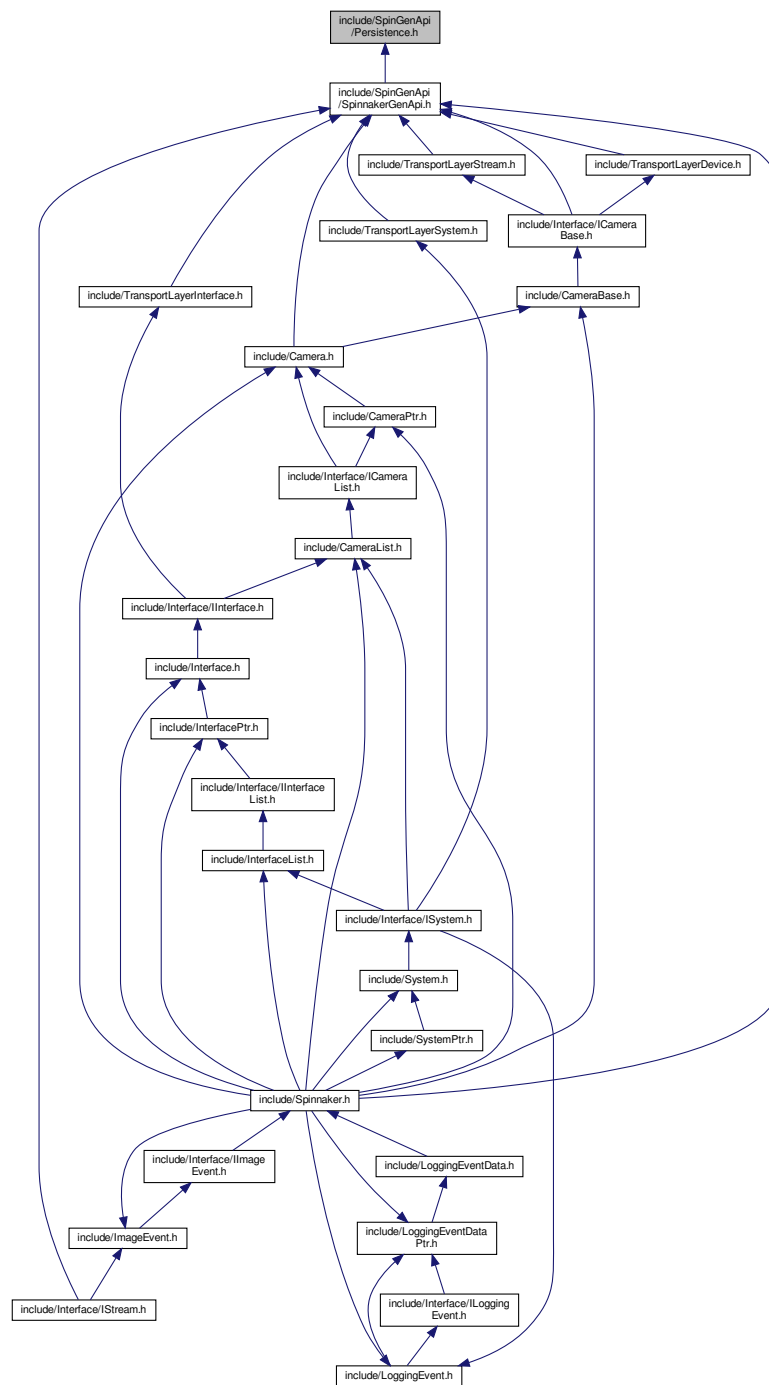
- [SPINNAKER_API](#) IDestroy * [CastToIDestroy](#) (INodeMap *pNodeMap)
makes sure the dynamic_cast operator is implemented in the DLL (due to a Linux bug)
- `template<class TCameraParams >`
`void _LoadXMLFromFile (const GenICam::gcstring &FileName)`
- `template<class TCameraParams >`
`void _LoadXMLFromZIPFile (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 ()`

11.108 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

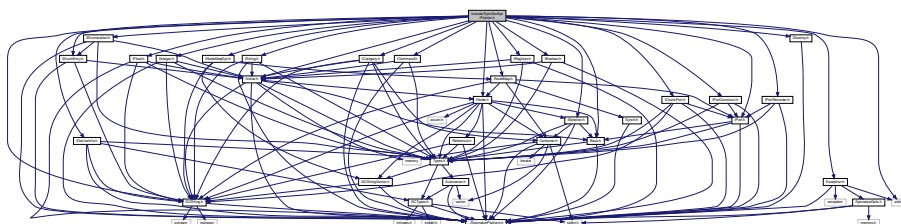
- virtual void [PersistFeature](#) (IValue &item)=0
Stores a feature.
- [SPINNAKER_API](#) std::istream & [EatComments](#) (std::istream &is)
*Helper function ignoring lines starting with comment character '#'.
Reads in persistent data from a stream.*
- [SPINNAKER_API](#) std::istream & [operator>>](#) (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.

11.109 include/SpinGenApi/Pointer.h File Reference

Include dependency graph for Pointer.h:



- typedef CPointer< IValue > [CValuePtr](#)
SmartPointer for IValue interface pointer.
- typedef CPointer< ICategory > [CCategoryPtr](#)
SmartPointer for ICategory interface pointer.
- typedef CPointer< IBoolean > [CBooleanPtr](#)
SmartPointer for IBoolean interface pointer.
- typedef CPointer< IInteger > [CIntegerPtr](#)
SmartPointer for IInteger interface pointer.
- typedef CPointer< IString > [CStringPtr](#)
SmartPointer for IString interface pointer.
- typedef CPointer< IRegister > [CRegisterPtr](#)
SmartPointer for IRegister interface pointer.
- typedef CPointer< IEnumeration > [CEnumerationPtr](#)
SmartPointer for IEnumeration interface pointer.
- typedef CPointer< IEnumEntry > [CEnumEntryPtr](#)
SmartPointer for IEnumEntry interface pointer.
- typedef CPointer< IPort > [CPortPtr](#)
SmartPointer for IPort interface pointer.
- typedef CPointer< IPortReplay > [CPortReplayPtr](#)
SmartPointer for IPortReplay interface pointer.
- typedef CPointer< IPortRecorder > [CPortRecorderPtr](#)
SmartPointer for IPortRecorder interface pointer.
- typedef CPointer< IPortWriteList, IPortWriteList > [CPortWriteListPtr](#)
SmartPointer for IPortWriteList interface pointer.
- typedef CPointer< IChunkPort > [CChunkPortPtr](#)
SmartPointer for IChunkPort interface pointer.
- typedef CPointer< INodeMap, INodeMap > [CNodeMapPtr](#)
SmartPointer for INodeMap interface pointer.
- typedef CPointer< INodeMapDyn, INodeMap > [CNodeMapDynPtr](#)
SmartPointer for INodeMapDyn interface pointer.
- typedef CPointer< IDeviceInfo, INodeMap > [CDeviceInfoPtr](#)
SmartPointer for IDeviceInfo interface pointer.
- typedef CPointer< ISelector > [CSelectorPtr](#)
SmartPointer for ISelector interface pointer.
- typedef CPointer< ICommand > [CCommandPtr](#)
SmartPointer for ICommand interface pointer.
- typedef CPointer< IPortConstruct > [CPortConstructPtr](#)
SmartPointer for IPortConstruct interface pointer.

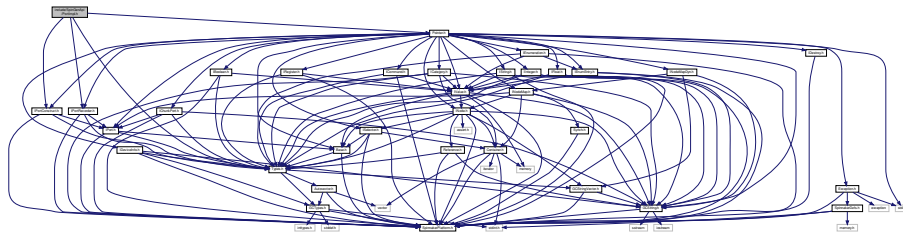
Functions

- template<class T , class B >
bool [IsReadable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is readable.
- template<class T , class B >
bool [IsWritable](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Writable.
- template<class T , class B >
bool [IsImplemented](#) (const [Spinnaker::GenApi::CPointer](#)< T, B > &ptr)
Checks if a node is Implemented.

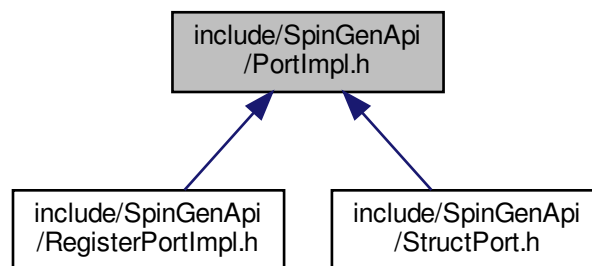
- `template<class T , class B >`
`bool IsAvailable (const Spinnaker::GenApi::CPointer< T, B > &ptr)`
Checks if a node is Available.
- `GenICam::gcstring GetInterfaceName (IBase *pBase)`
Returns the name of the main interface as string DEPRICATED, use `IBase::GetPrincipalInterfaceType()` instead.

11.110 include/SpinGenApi/PortImpl.h File Reference

Include dependency graph for PortImpl.h:



This graph shows which files directly or indirectly include this file:



Classes

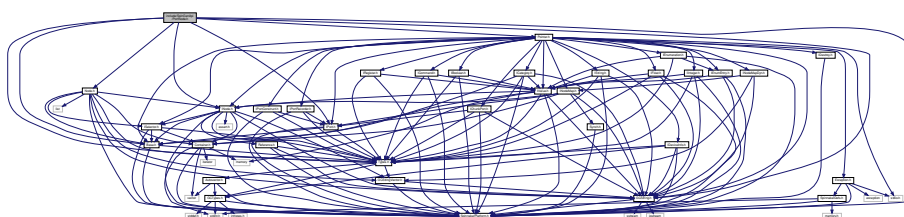
- class `CPortImpl`
Standard implementation for a port.

Namespaces

- `Spinnaker`
- `Spinnaker::GenApi`

11.111 include/SpinGenApi/PortNode.h File Reference

Include dependency graph for PortNode.h:



Namespaces

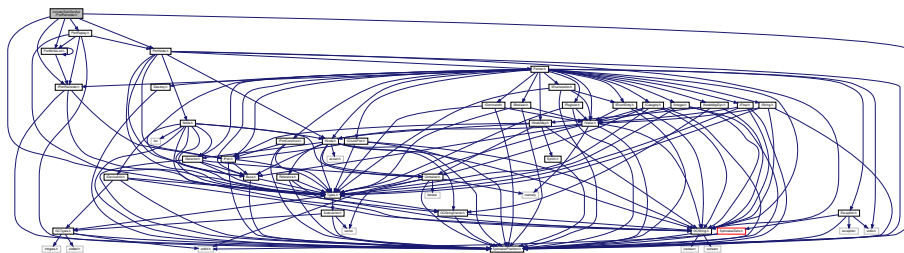
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

- typedef PortNode [CPortRef](#)

11.112 include/SpinGenApi/PortRecorder.h File Reference

Include dependency graph for PortRecorder.h:



Classes

- class [PortRecorder](#)
Interface for recording write commands on a port.

Namespaces

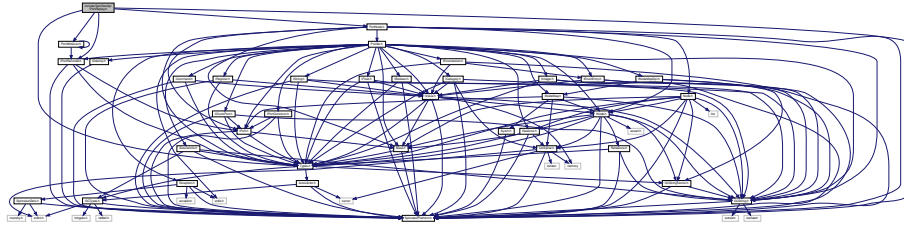
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Typedefs

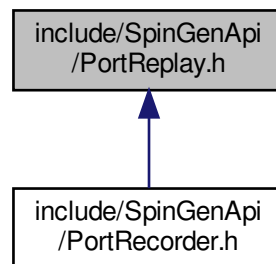
- typedef PortRecorder [CPortRecorderRef](#)
Reference to an IPortRecorder pointer.

11.113 include/SpinGenApi/PortReplay.h File Reference

Include dependency graph for PortReplay.h:



This graph shows which files directly or indirectly include this file:



Classes

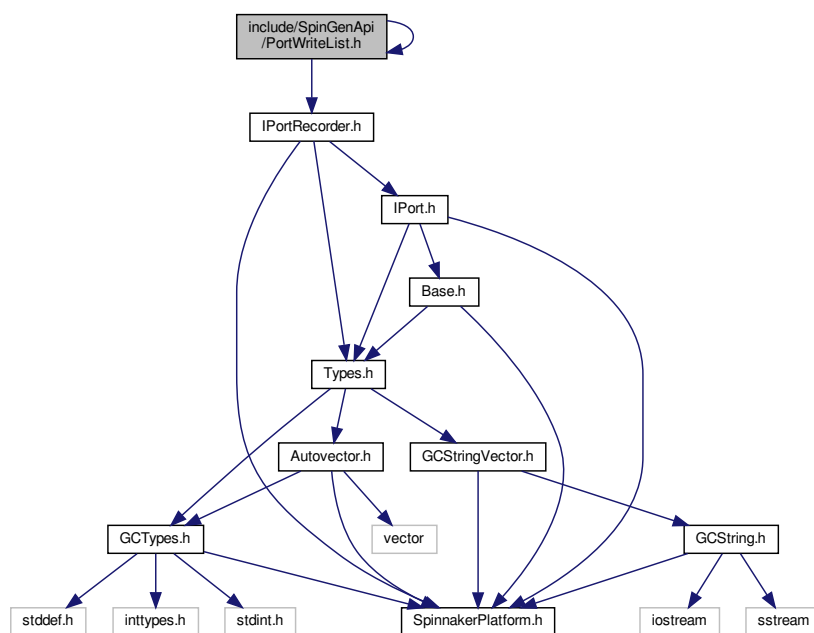
- class [PortReplay](#)
Interface for replaying write commands on a port.

Namespaces

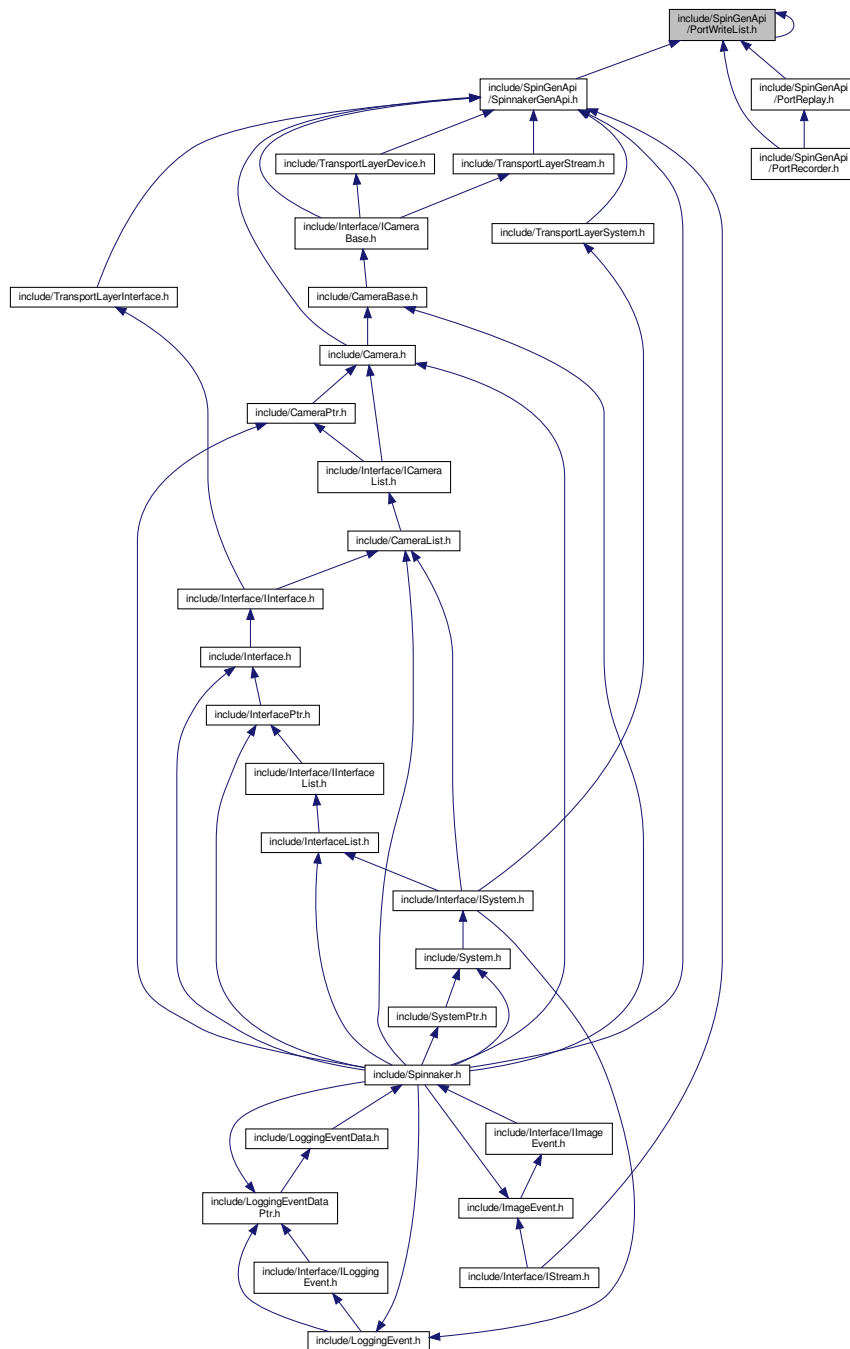
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.114 include/SpinGenApi/PortWriteList.h File Reference

Include dependency graph for PortWriteList.h:



This graph shows which files directly or indirectly include this file:



Classes

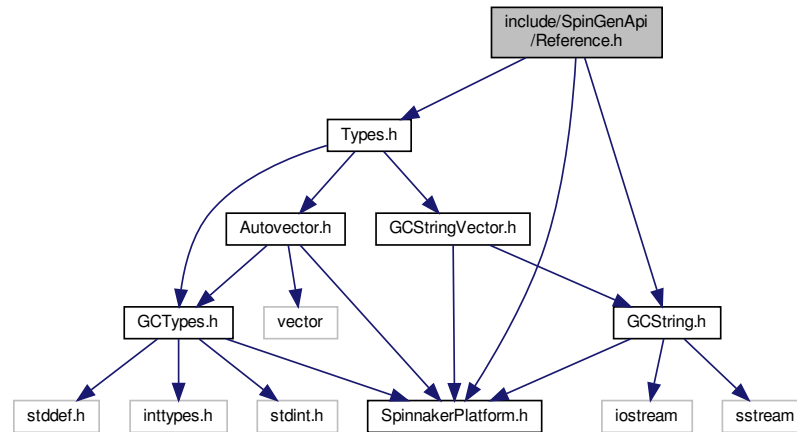
- class [CPortWriteList](#)
Container holding a list of port write commands.

Namespaces

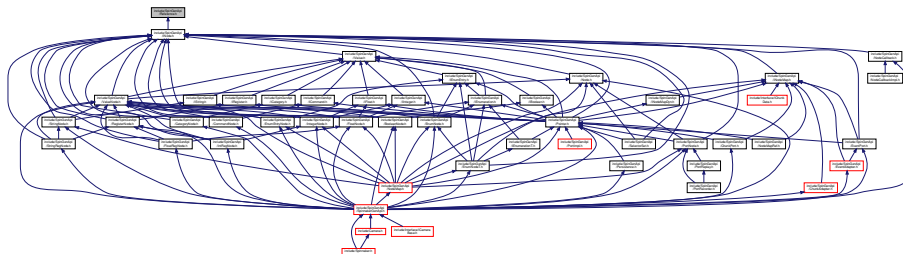
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.115 include/SpinGenApi/Reference.h File Reference

Include dependency graph for Reference.h:



This graph shows which files directly or indirectly include this file:



Namespaces

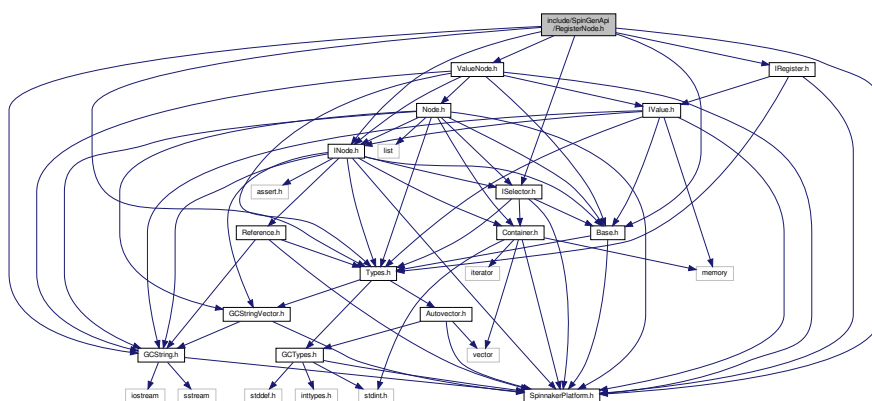
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

Functions

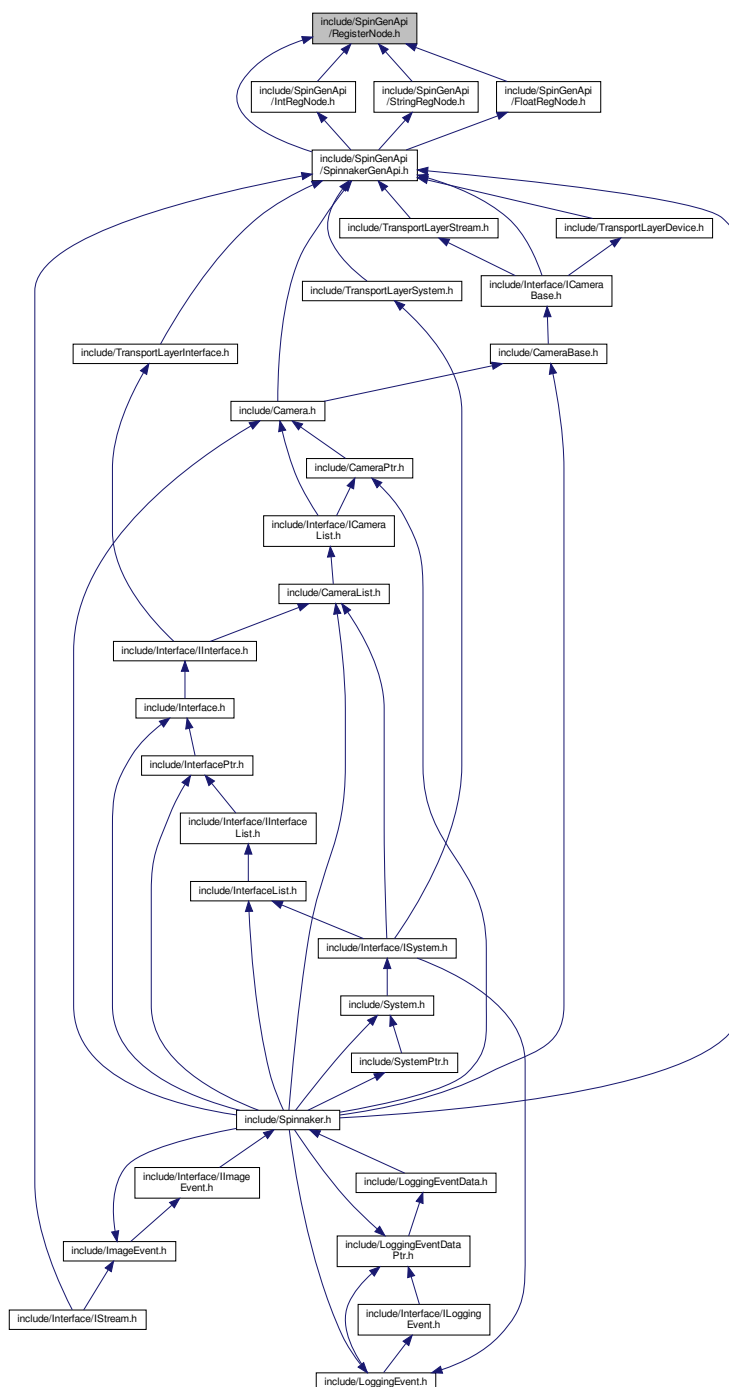
- virtual void [SetNumEnums](#) (int NumEnums)=0
sets the number of enum values

11.116 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.

Classes

- class [CSelectorSet](#)

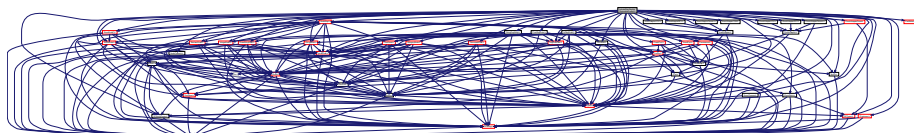
The set of selectors selecting a given node.

Namespaces

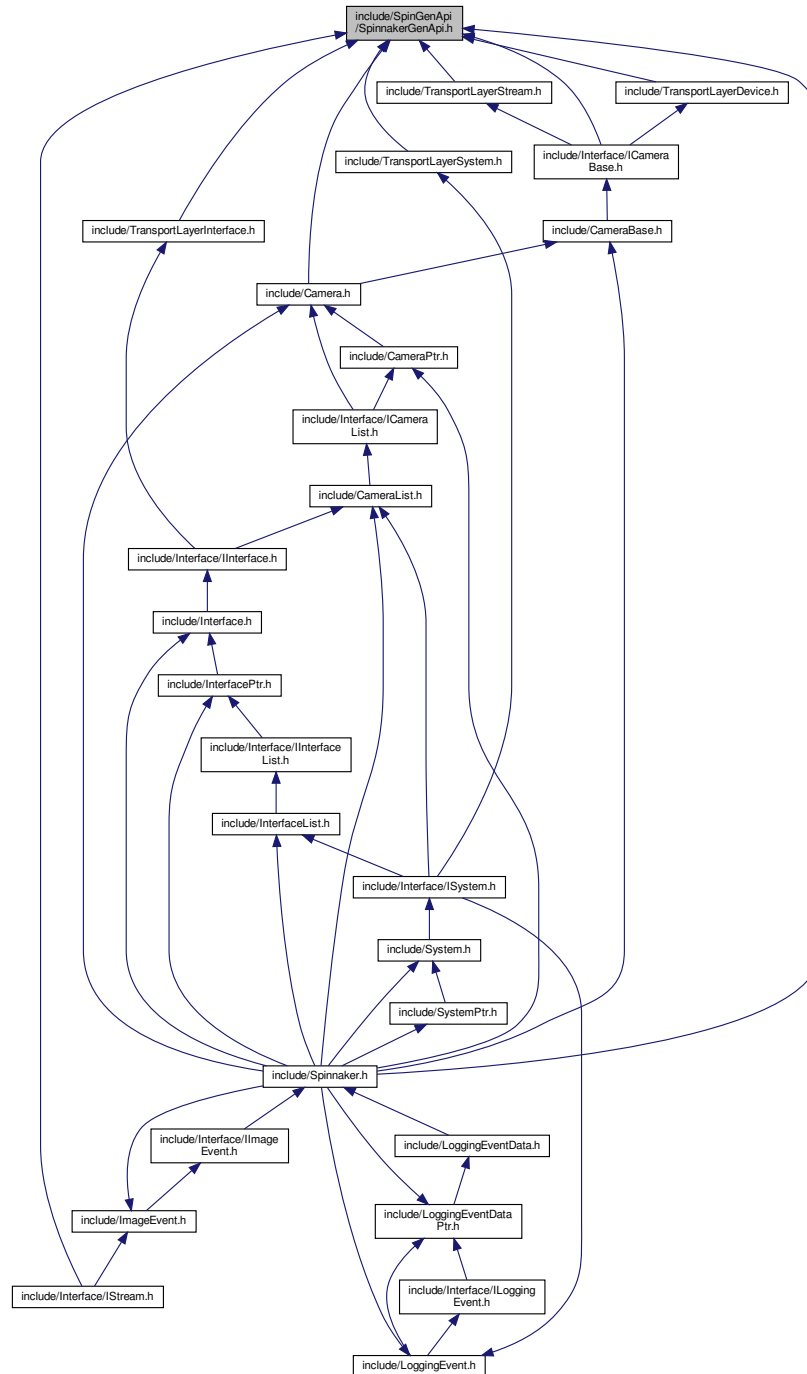
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.119 include/SpinGenApi/SpinnakerGenApi.h File Reference

Include dependency graph for SpinnakerGenApi.h:

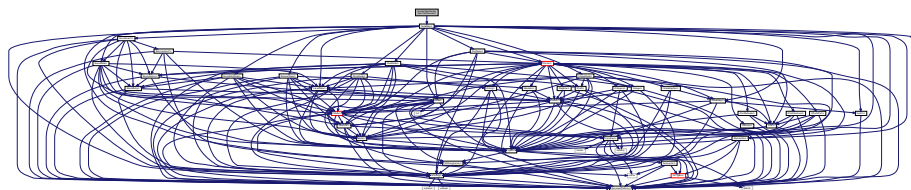


This graph shows which files directly or indirectly include this file:



11.120 include/SpinGenApi/SpinTestCamera.h File Reference

Include dependency graph for SpinTestCamera.h:



Classes

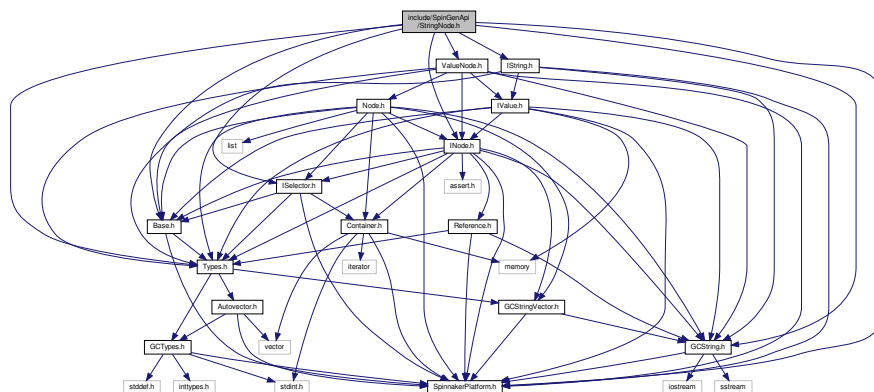
- class [SpinTestCamera](#)

Namespaces

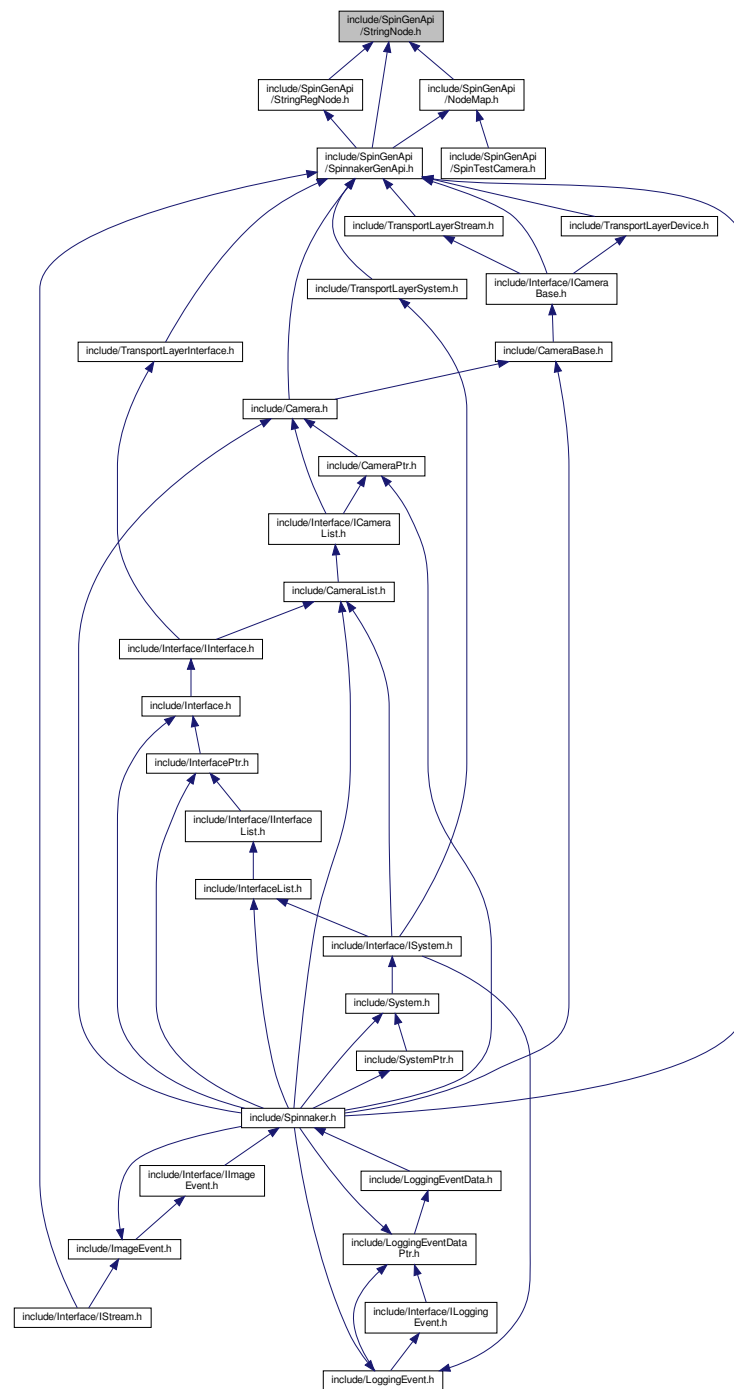
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.121 include/SpinGenApi/StringNode.h File Reference

Include dependency graph for StringNode.h:



This graph shows which files directly or indirectly include this file:

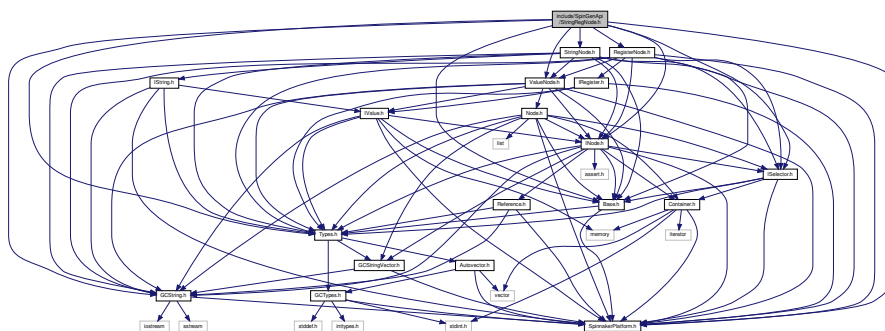


Classes

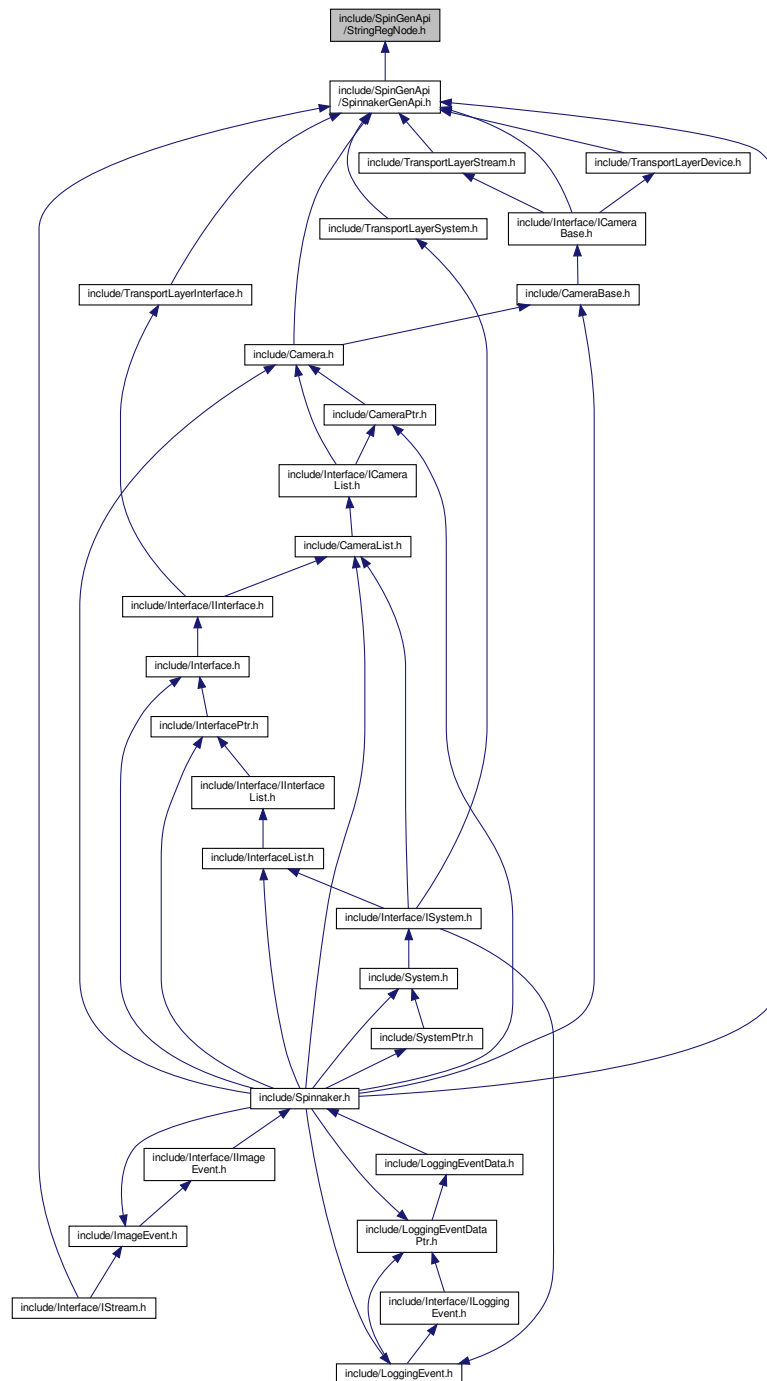
- class [StringNode](#)
Interface for string properties.

- Spinnaker
- Spinnaker::GenApi

- typedef StringNode CStringRef



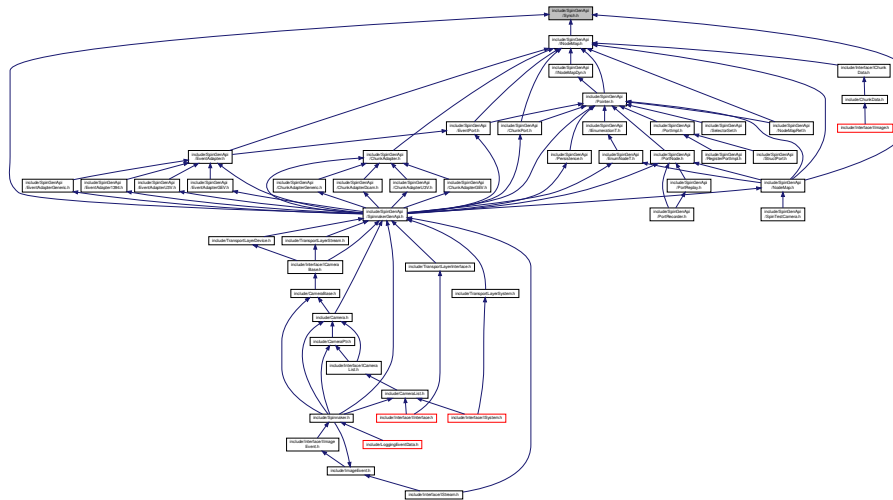
This graph shows which files directly or indirectly include this file:



Classes

- class [StringRegNode](#)
Interface for string properties.

This graph shows which files directly or indirectly include this file:



Classes

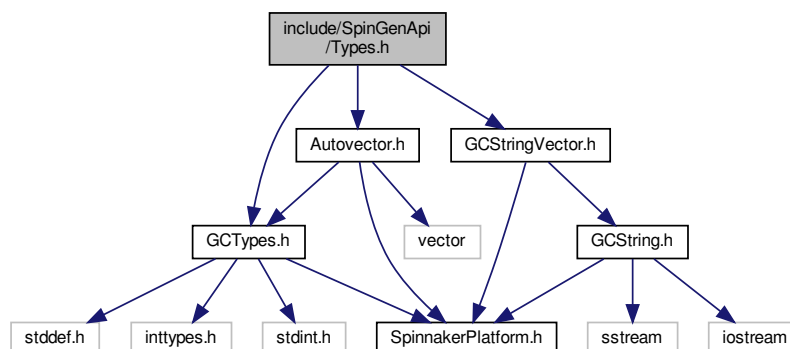
- class [CLock](#)
A lock class.
- class [CLockEx](#)
This class is for testing purposes only.
- class [AutoLock](#)

Namespaces

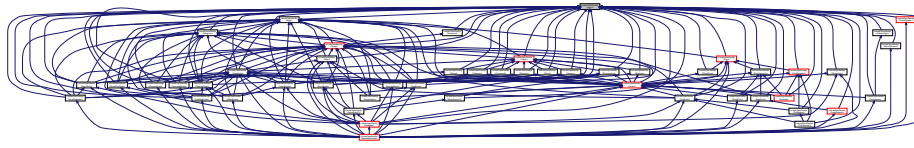
- [Spinnaker](#)
- [Spinnaker::GenApi](#)

11.125 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 [EEndianness](#) {
[BigEndian](#),
[LittleEndian](#),
[_UndefinedEndian](#) }

Endianness of a value in a register.

- enum [ENameSpace](#) {
[Custom](#),
[Standard](#),
[_UndefinedNameSpace](#) }

Defines if a node name is standard or custom.

- enum [EStandardNameSpace](#) {
[None](#),
[GEV](#),
[IIDC](#),
[CL](#),
[USB](#),
[_UndefinedStandardNameSpace](#) }

Defines from which standard namespace a node name comes from.

- enum [EYesNo](#) {
[Yes](#) = 1,
[No](#) = 0,
[_UndefinedYesNo](#) = 2 }

Defines the choices of a Yes/No alternative.

- enum [ESlope](#) {
[Increasing](#),
[Decreasing](#),
[Varying](#),
[Automatic](#),
[_UndefinedESlope](#) }

typedef for formula type

- enum [EXMLValidation](#) {
[xvLoad](#) = 0x00000001L,
[xvCycles](#) = 0x00000002L,
[xvSFNC](#) = 0x00000004L,
[xvDefault](#) = 0x00000000L,
[xvAll](#) = 0xffffffffL,
[_UndefinedEXMLValidation](#) = 0x80000000L }

typedef describing the different validity checks which can be performed on an XML file

- enum [EDisplayNotation](#) {
[fnAutomatic](#),
[fnFixed](#),
[fnScientific](#),
[_UndefinedEDisplayNotation](#) }

typedef for float notation

- enum [EInterfaceType](#) {
[intfIValue](#),
[intfIBase](#),
[intfInteger](#),
[intfBoolean](#),
[intfCommand](#),
[intfFloat](#),
[intfString](#),
[intfRegister](#),
[intfCategory](#),
[intfEnumeration](#),
[intfEnumEntry](#),
[intfIPort](#) }

typedef for interface type

- enum [ELinkType](#) {
[ctParentNodes](#),
[ctReadingChildren](#),
[ctWritingChildren](#),
[ctInvalidatingChildren](#),
[ctDependingNodes](#),
[ctTerminalNodes](#) }

typedef for link type

- enum [EIncMode](#) {
[noIncrement](#),
[fixedIncrement](#),
[listIncrement](#) }

typedef for increment mode

- enum [EInputDirection](#) {
[idFrom](#),
[idTo](#),
[idNone](#) }

typedef for link type

- enum [EGenApiSchemaVersion](#) {
[v1_0](#) = 1,
[v1_1](#) = 2,
[_Undefined](#) = -1 }

GenApi schema version.

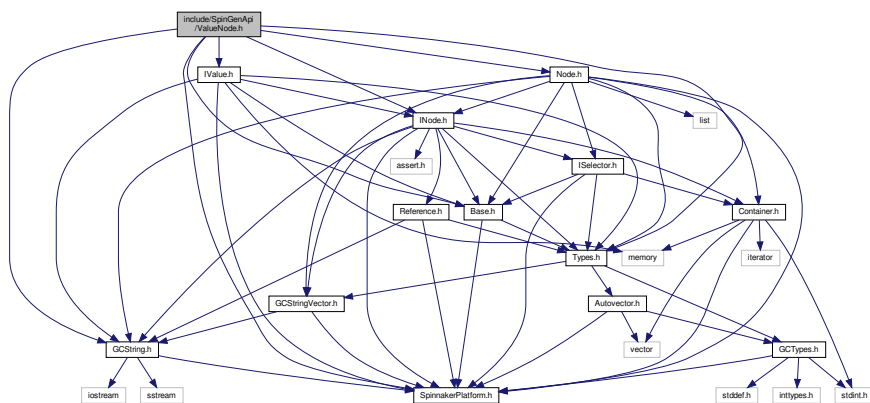
11.125.1 Macro Definition Documentation

11.125.1.1 interface

```
#define interface struct
```

11.126 include/SpinGenApi/ValueNode.h File Reference

Include dependency graph for ValueNode.h:



[illegible]

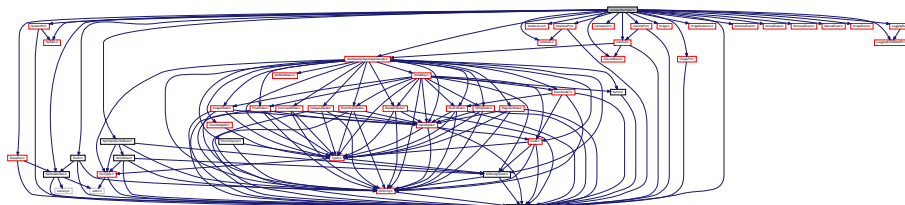
- class `ValueNode`
Interface for value properties.

- Spinnaker
- Spinnaker::GenApi

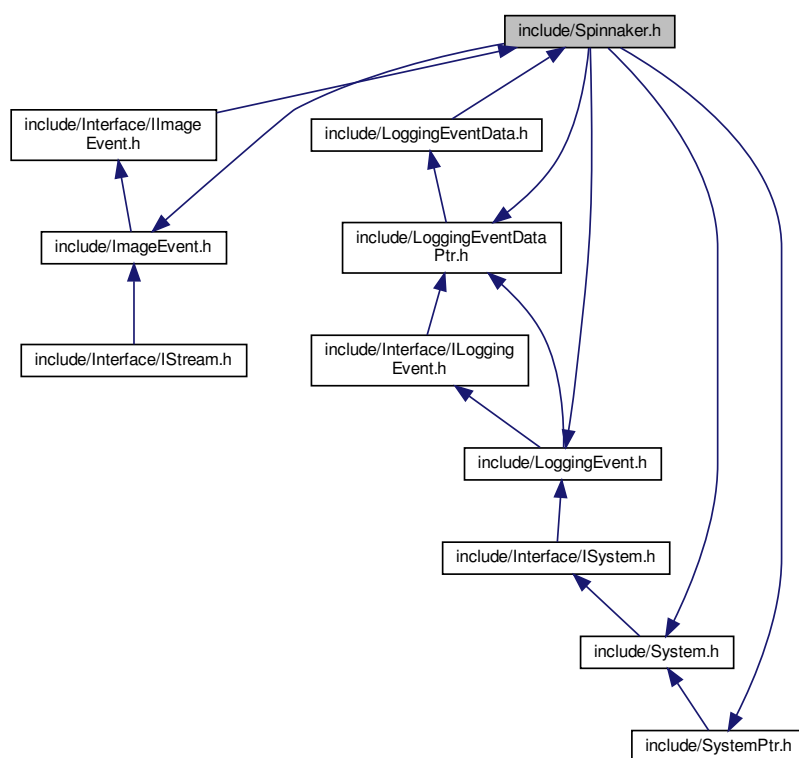
- typedef ValueNode CValueRef

11.127 include/Spinnaker.h File Reference

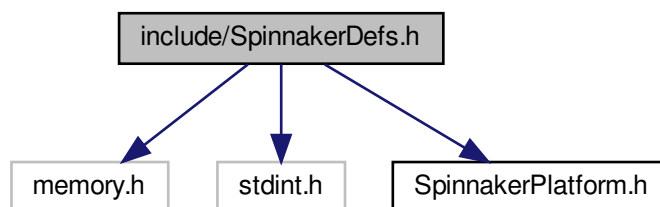
Include dependency graph for Spinnaker.h:



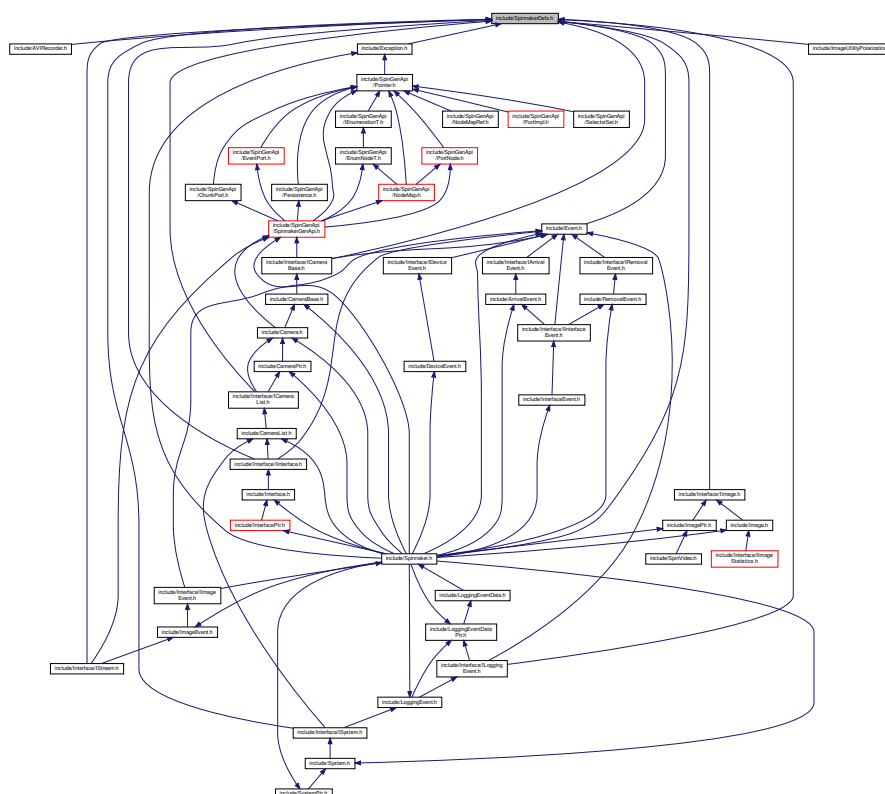
This graph shows which files directly or indirectly include this file:



Include dependency graph for SpinnakerDefs.h:



This graph shows which files directly or indirectly include this file:



- 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 }

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 = 0x7FFFFFFF }

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` = 0x8013,
`ACTION_COMMAND_STATUS_OVERFLOW` = 0x8015,
`ACTION_COMMAND_STATUS_ACTION_LATE` = 0x8016,
`ACTION_COMMAND_STATUS_ERROR` = 0x8FFF }

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](#) }

Functions

- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. Polarization images are now created through specific functions the ImageUtilityPolarization class.") [PolarizationAlgorithm](#)
- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. Image scaling can now be applied through specific functions defined in the ImageUtility class.") [PolarizationResolution](#)
- enum [DEPRECATED_CLASS](#) ("This enum has been deprecated. Please use HeatmapColor in the ImageUtilityHeatmap class.") [HeatMapColor](#)

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

11.129 include/SpinnakerPlatform.h File Reference

Macros

- #define [SPINNAKER_API_ABSTRACT](#) /*nothing*/
- #define [SPINNAKER_API](#) __attribute__((visibility ("default"))))
- #define [SPINNAKER_LOCAL](#) __attribute__((visibility ("hidden"))))

11.130 include/SpinUpdate.h File Reference

Macros

- #define [SPINUPDATE_API](#) SPINUPDATE_IMPORT_EXPORT

Functions

- `SPINUPDATE_API` int `UpdateFirmwareConsole` (int argc, char **argv)
Updates the firmware for the device.
- `SPINUPDATE_API` int `UpdateFirmware` (const char *args)
- `SPINUPDATE_API` void `SetMessageCallback` (`UpdaterMessageCallback` messageCallbackFunction)
- `SPINUPDATE_API` void `SetProgressCallback` (`UpdaterProgressCallback` progressCallbackFunction)
- `SPINUPDATE_API` const char * `GetErrorMessage` ()

Variables

- `SPINUPDATE_API` typedef int(* `UpdaterMessageCallback`)(const char *message)
- `SPINUPDATE_API` typedef int(* `UpdaterProgressCallback`)(const char *action, unsigned int address, int globalPercent, int currPercent)

11.130.1 Macro Definition Documentation

11.130.1.1 SPINUPDATE_API

```
#define SPINUPDATE_API SPINUPDATE_IMPORT_EXPORT
```

11.130.2 Function Documentation

11.130.2.1 GetErrorMessage()

```
SPINUPDATE_API const char* GetErrorMessage ( )
```

11.130.2.2 SetMessageCallback()

```
SPINUPDATE_API void SetMessageCallback (
    UpdaterMessageCallback messageCallbackFunction )
```

11.130.2.3 SetProgressCallback()

```
SPINUPDATE_API void SetProgressCallback (
    UpdaterProgressCallback progressCallbackFunction )
```

11.130.2.4 UpdateFirmware()

```
SPINUPDATE_API int UpdateFirmware (
    const char * args )
```

11.130.2.5 UpdateFirmwareConsole()

```
SPINUPDATE_API int UpdateFirmwareConsole (
    int argc,
    char ** argv )
```

Updates the firmware for the device.

Parameters

| | |
|-------------|---|
| <i>argc</i> | Number of strings pointed to by argv |
| <i>argv</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 the EEPROM to a file.

11.130.3 Variable Documentation

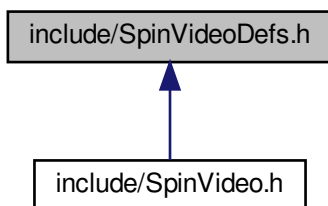
11.130.3.1 UpdatorMessageCallback

```
SPINUPDATE_API typedef int(* UpdatorMessageCallback) (const char *message)
```

11.130.3.2 UpdatorProgressCallback

```
SPINUPDATE_API typedef int(* UpdatorProgressCallback) (const char *action, unsigned int address,
int globalPercent, int currPercent)
```


This graph shows which files directly or indirectly include this file:



Classes

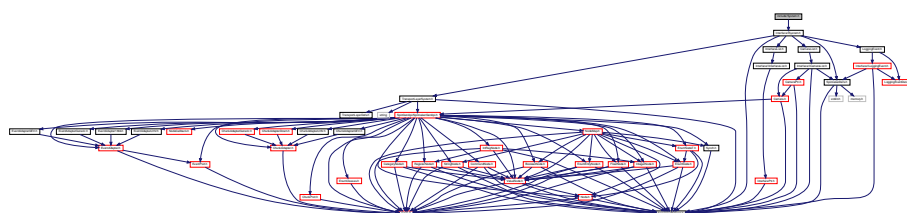
- struct [MJPGOption](#)
Options for saving MJPG files.
- struct [H264Option](#)
Options for saving H264 files.
- struct [AVIOption](#)
Options for saving AVI files.

Namespaces

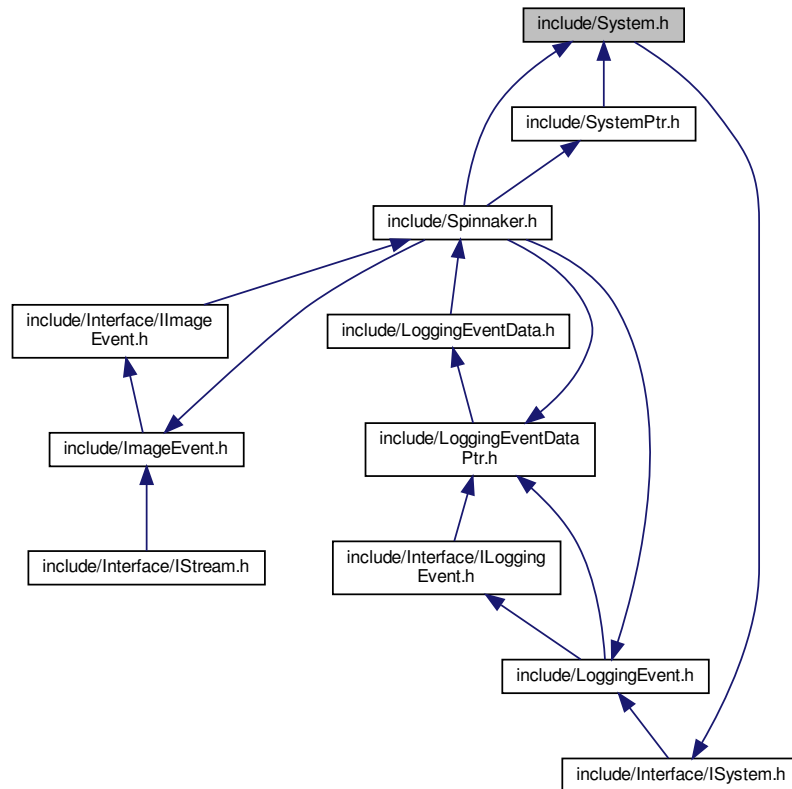
- [Spinnaker](#)
- [Spinnaker::Video](#)

11.133 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 1`
- `#define FLIR_SPINNAKER_VERSION_MINOR 23`
- `#define FLIR_SPINNAKER_VERSION_TYPE 0`
- `#define FLIR_SPINNAKER_VERSION_BUILD 27`

11.133.1 Macro Definition Documentation

11.133.1.1 FLIR_SPINNAKER_VERSION_BUILD

```
#define FLIR_SPINNAKER_VERSION_BUILD 27
```

11.133.1.2 FLIR_SPINNAKER_VERSION_MAJOR

```
#define FLIR_SPINNAKER_VERSION_MAJOR 1
```

11.133.1.3 FLIR_SPINNAKER_VERSION_MINOR

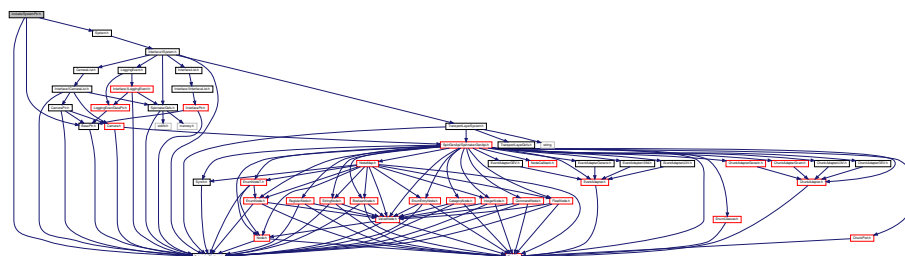
```
#define FLIR_SPINNAKER_VERSION_MINOR 23
```

11.133.1.4 FLIR_SPINNAKER_VERSION_TYPE

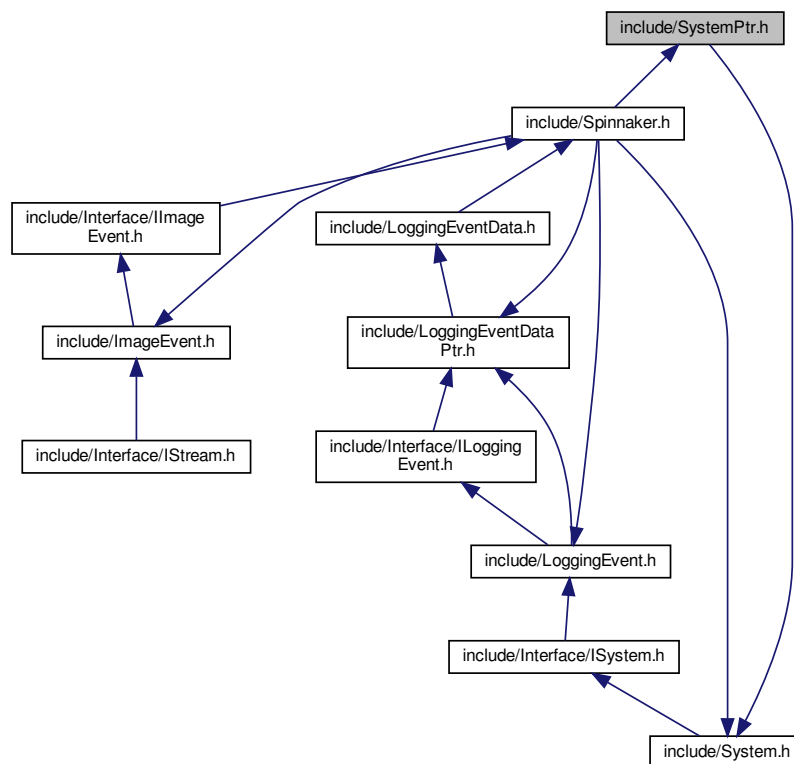
```
#define FLIR_SPINNAKER_VERSION_TYPE 0
```

11.134 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](#)

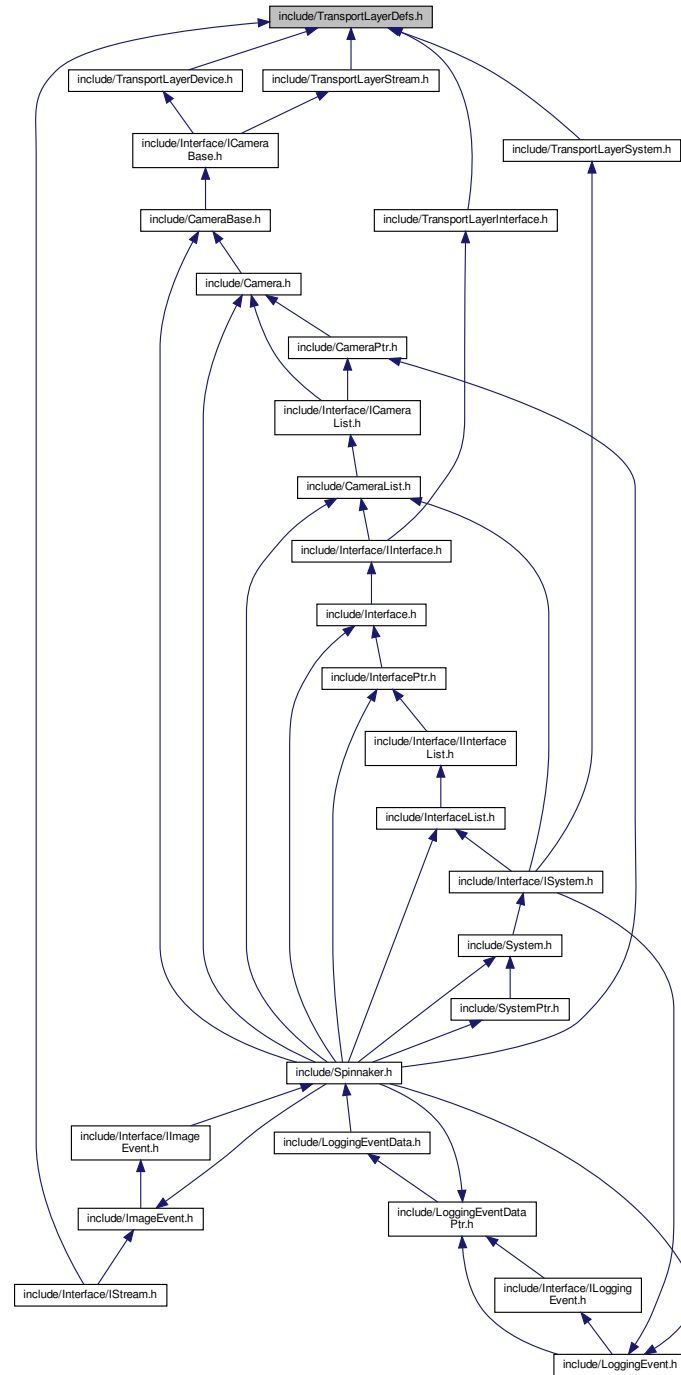
A reference tracked pointer to a system object.

Namespaces

- [Spinnaker](#)

11.135 include/TransportLayerDefs.h File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [Spinnaker](#)

Enumerations

- enum [StreamTypeEnum](#) {
[StreamType_Mixed](#),
[StreamType_Custom](#),
[StreamType_GEV](#),
[StreamType_CL](#),
[StreamType_IIDC](#),
[StreamType_UVC](#),
[StreamType_CXP](#),
[StreamType_CLHS](#),
[StreamType_U3V](#),
[StreamType_ETHERNET](#),
[StreamType_PCI](#),
[NUMSTREAMTYPE](#) }

The enum definitions for TL Device nodes from the transport layer .xml files.

- enum [StreamDefaultBufferCountModeEnum](#) {
[StreamDefaultBufferCountMode_Manual](#),
[StreamDefaultBufferCountMode_Auto](#),
[NUMSTREAMDEFAULTBUFFERCOUNTMODE](#) }
- enum [StreamBufferCountModeEnum](#) {
[StreamBufferCountMode_Manual](#),
[StreamBufferCountMode_Auto](#),
[NUMSTREAMBUFFERCOUNTMODE](#) }
- enum [StreamBufferHandlingModeEnum](#) {
[StreamBufferHandlingMode_OldestFirst](#),
[StreamBufferHandlingMode_OldestFirstOverwrite](#),
[StreamBufferHandlingMode_NewestFirst](#),
[StreamBufferHandlingMode_NewestFirstOverwrite](#),
[StreamBufferHandlingMode_NewestOnly](#),
[NUMSTREAMBUFFERHANDLINGMODE](#) }
- enum [DeviceTypeEnum](#) {
[DeviceType_Mixed](#),
[DeviceType_Custom](#),
[DeviceType_GEV](#),
[DeviceType_CL](#),
[DeviceType_IIDC](#),
[DeviceType_UVC](#),
[DeviceType_CXP](#),
[DeviceType_CLHS](#),
[DeviceType_U3V](#),
[DeviceType_ETHERNET](#),
[DeviceType_PCI](#),
[NUMDEVICETYPE](#) }
- enum [DeviceAccessStatusEnum](#) {
[DeviceAccessStatus_Unknown](#),
[DeviceAccessStatus_ReadWrite](#),
[DeviceAccessStatus_ReadOnly](#),
[DeviceAccessStatus_NoAccess](#),
[NUMDEVICEACCESSSTATUS](#) }
- enum [GevCCPEnum](#) {
[GevCCP_EnumEntry_GevCCP_OpenAccess](#),
[GevCCP_EnumEntry_GevCCP_ExclusiveAccess](#),
[GevCCP_EnumEntry_GevCCP_ControlAccess](#),
[NUMGEVCCP](#) }
- enum [GUIXMLLocationEnum](#) {
[GUIXMLLocation_Device](#),

```

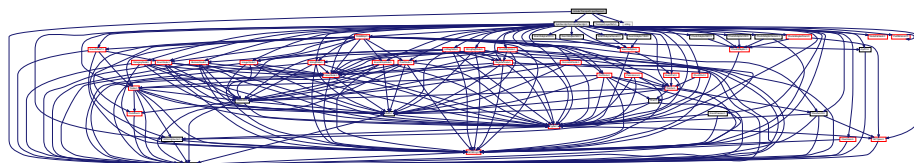
GUIXMLLocation_Host,
NUMGUIXMLLOCATION }

```

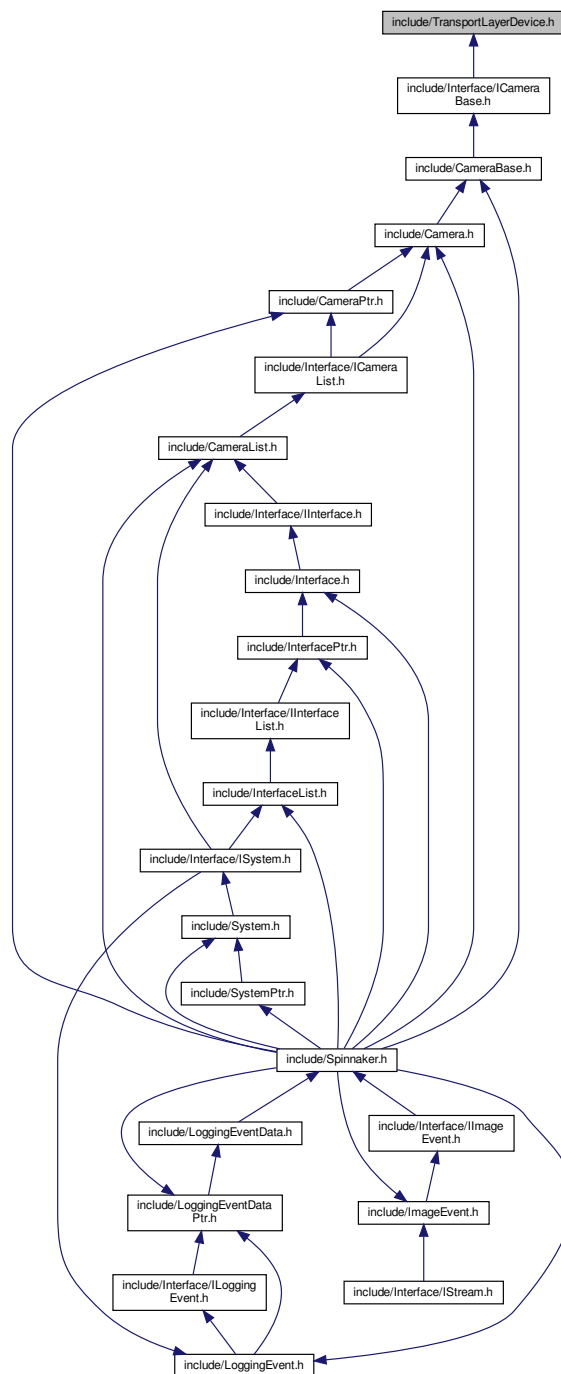
- enum `GenICamXMLLocationEnum` {
`GenICamXMLLocation_Device`,
`GenICamXMLLocation_Host`,
`NUMGENICAMXMLLOCATION` }
- enum `DeviceEndiannessMechanismEnum` {
`DeviceEndiannessMechanism_Legacy`,
`DeviceEndiannessMechanism_Standard`,
`NUMDEVICEENDIANESSMECHANISM` }
- enum `DeviceCurrentSpeedEnum` {
`DeviceCurrentSpeed_UnknownSpeed`,
`DeviceCurrentSpeed_LowSpeed`,
`DeviceCurrentSpeed_FullSpeed`,
`DeviceCurrentSpeed_HighSpeed`,
`DeviceCurrentSpeed_SuperSpeed`,
`NUMDEVICECURRENTSPEED` }
- enum `POEStatusEnum` {
`POEStatus_NotSupported`,
`POEStatus_PowerOff`,
`POEStatus_PowerOn`,
`NUMPOESTATUS` }
- enum `FilterDriverStatusEnum` {
`FilterDriverStatus_NotSupported`,
`FilterDriverStatus_Disabled`,
`FilterDriverStatus_Enabled`,
`NUMFILTERDRIVERSTATUS` }

11.136 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`

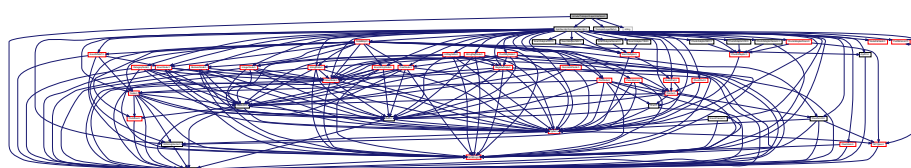
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

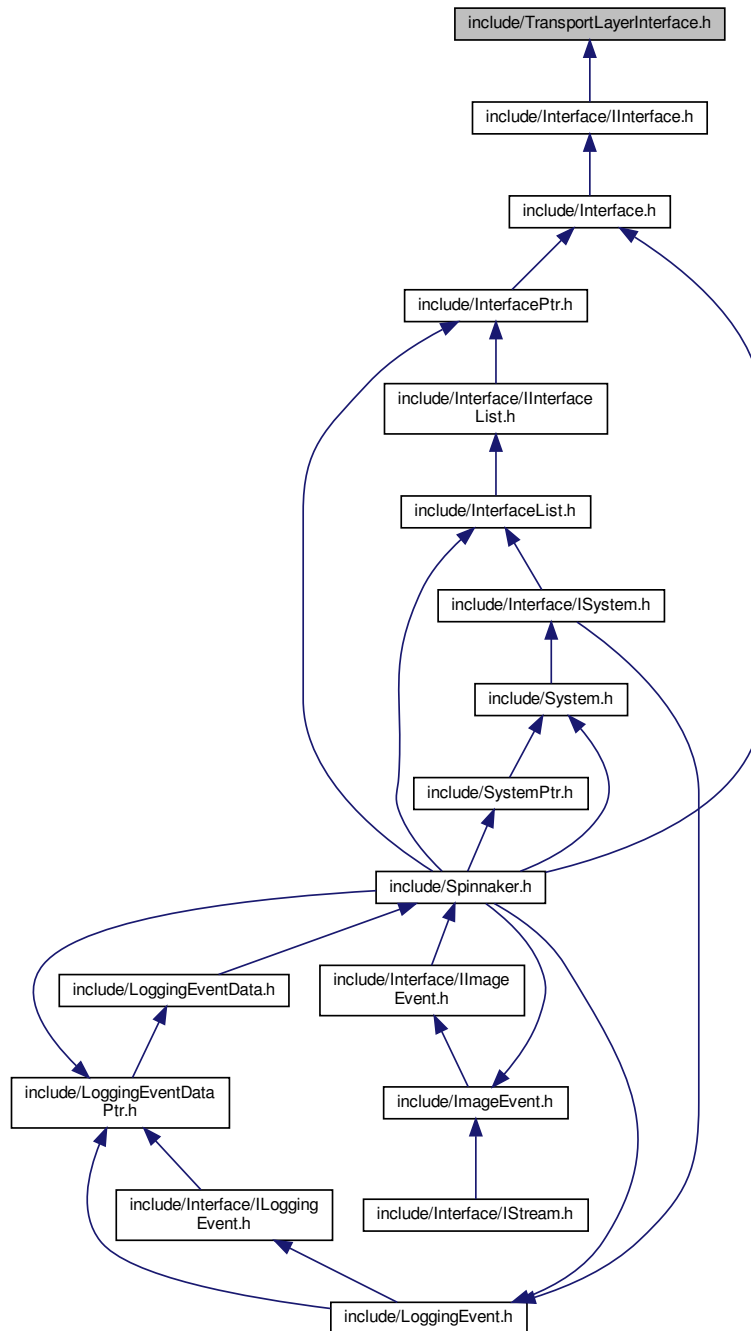
- Spinnaker

11.137 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](#)

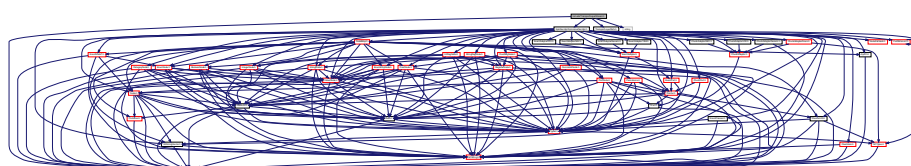
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

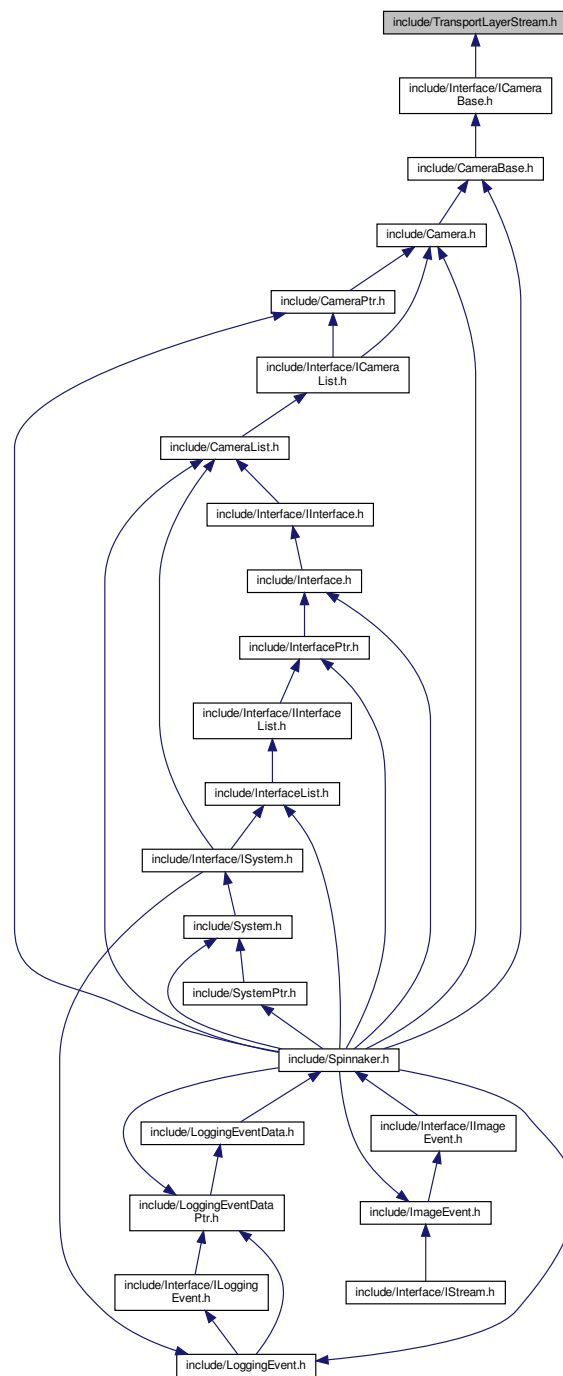
- [Spinnaker](#)

11.138 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](#)

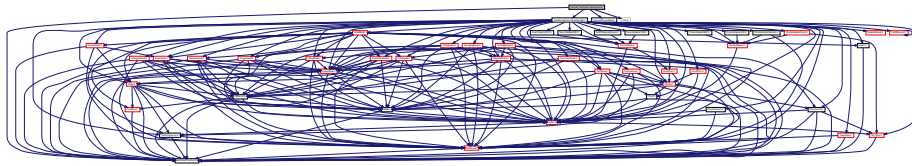
Part of the QuickSpin API to provide access to camera information without having to first initialize the camera.

Namespaces

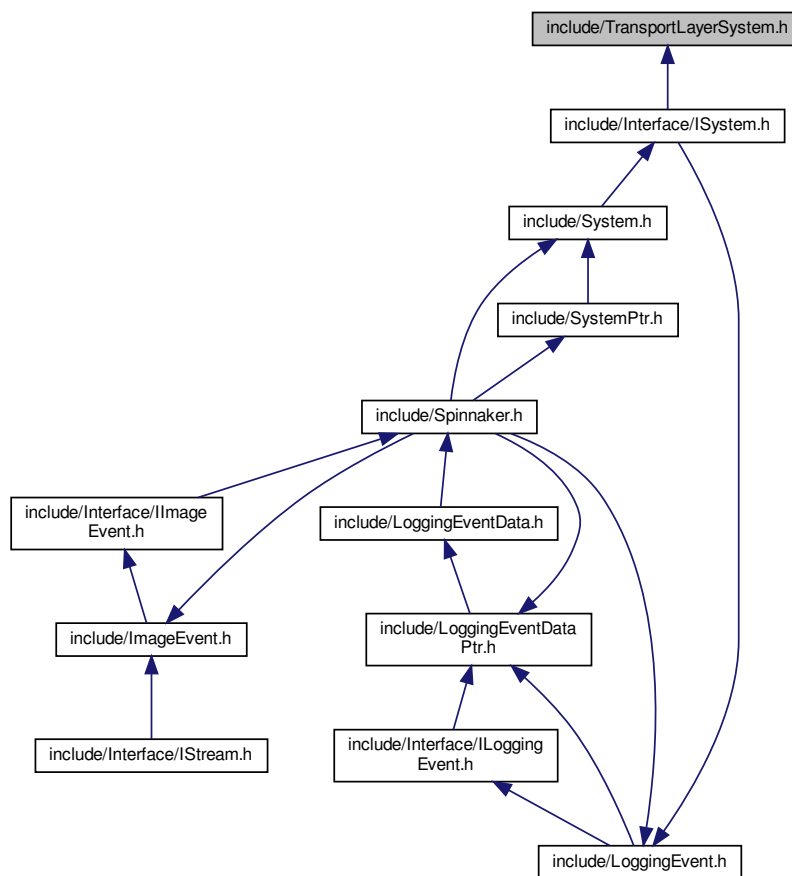
- [Spinnaker](#)

11.139 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](#)

Index

- _ClearXMLCache
 - Spinnaker GenApi Classes, [220](#)
 - Spinnaker::GenApi::CNodeMapRefT, [699](#)
 - _Connect
 - Spinnaker GenApi Classes, [220](#), [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [699](#)
 - _Destroy
 - Spinnaker GenApi Classes, [221](#)
 - _GetDeviceName
 - Spinnaker GenApi Classes, [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [699](#)
 - _GetNode
 - Spinnaker GenApi Classes, [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [699](#)
 - _GetNodes
 - Spinnaker GenApi Classes, [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [700](#)
 - _GetSupportedSchemaVersions
 - Spinnaker GenApi Classes, [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [700](#)
 - _Initialize
 - Spinnaker::GenApi::CGeneric_XMLLoader↔
Params, [659](#)
 - _InvalidateNodes
 - Spinnaker GenApi Classes, [221](#)
 - Spinnaker::GenApi::CNodeMapRefT, [700](#)
 - _LoadXMLFromFile
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [700](#)
 - _LoadXMLFromFileInject
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [700](#)
 - _LoadXMLFromString
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [701](#)
 - _LoadXMLFromStringInject
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [701](#)
 - _LoadXMLFromZIPData
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [701](#)
 - _LoadXMLFromZIPFile
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [701](#)
 - _Poll
 - Spinnaker GenApi Classes, [222](#)
 - Spinnaker::GenApi::CNodeMapRefT, [701](#)
 - _Ptr
 - Spinnaker::GenApi::CNodeMapRefT, [702](#)
 - Spinnaker::GenApi::NodeMap, [990](#)
 - _TO_STRING
 - GCUtilities.h, [1241](#)
 - _UndefinedRepresentation
 - Types Enums, [354](#)
 - __ERR__
 - GCUtilities.h, [1241](#)
 - __LINE_STR__
 - GCUtilities.h, [1241](#)
 - __LOCATION__
 - GCUtilities.h, [1241](#)
 - __OUTPUT_FORMATER__
 - GCUtilities.h, [1241](#)
 - __STDC_CONSTANT_MACROS
 - GCTypes.h, [1236](#)
 - __STDC_LIMIT_MACROS
 - GCTypes.h, [1236](#)
 - __TODO__
 - GCUtilities.h, [1241](#)
 - __WARN__
 - GCUtilities.h, [1241](#)
 - _npos
 - Spinnaker::GenICam::gcstring, [784](#)
 - _pCount
 - Spinnaker::GenApi::double_autovector_t, [735](#)
 - Spinnaker::GenApi::int64_autovector_t, [923](#)
 - _pv
 - Spinnaker::GenApi::double_autovector_t, [735](#)
 - Spinnaker::GenApi::int64_autovector_t, [923](#)
 - ~ArrivalEvent
 - Spinnaker::ArrivalEvent, [427](#)
 - ~AutoLock
 - Spinnaker::GenApi::AutoLock, [429](#)
 - Spinnaker::GenICam::AutoLock, [430](#)
 - ~BasePtr
 - Spinnaker::BasePtr, [432](#)
 - ~BooleanNode
 - Spinnaker::GenApi::BooleanNode, [438](#)
 - ~CChunkAdapter
 - Spinnaker::GenApi::CChunkAdapter, [616](#)
 - ~CChunkAdapterDcam
 - Spinnaker::GenApi::CChunkAdapterDcam, [620](#)
 - ~CChunkAdapterGEV
 - Spinnaker::GenApi::CChunkAdapterGEV, [625](#)
 - ~CChunkAdapterGeneric
 - Spinnaker::GenApi::CChunkAdapterGeneric, [622](#)
 - ~CChunkAdapterU3V
 - Spinnaker::GenApi::CChunkAdapterU3V, [627](#)
 - ~CChunkPort

- Spinnaker::GenApi::CChunkPort, 629
- ~CEnumerationTRef
 - Spinnaker::GenApi::CEnumerationTRef, 635
- ~CEventAdapter
 - Spinnaker::GenApi::CEventAdapter, 639
- ~CEventAdapter1394
 - Spinnaker::GenApi::CEventAdapter1394, 641
- ~CEventAdapterGEV
 - Spinnaker::GenApi::CEventAdapterGEV, 646
- ~CEventAdapterGeneric
 - Spinnaker::GenApi::CEventAdapterGeneric, 643
- ~CEventAdapterU3V
 - Spinnaker::GenApi::CEventAdapterU3V, 648
- ~CEventPort
 - Spinnaker::GenApi::CEventPort, 651
- ~CFeatureBag
 - Spinnaker::GenApi::CFeatureBag, 655
- ~CGlobalLock
 - Spinnaker::GenICam::CGlobalLock, 661
- ~CGlobalLockUnlocker
 - Spinnaker::GenICam::CGlobalLockUnlocker, 663
- ~CLock
 - Spinnaker::GenApi::CLock, 677
 - Spinnaker::GenICam::CLock, 679
- ~CNodeCallback
 - Spinnaker::GenApi::CNodeCallback, 684
- ~CNodeMapFactory
 - Spinnaker::GenApi::CNodeMapFactory, 688
- ~CNodeMapRefT
 - Spinnaker GenApi Classes, 224
- ~CPointer
 - Spinnaker::GenApi::CPointer, 709
- ~CPortImpl
 - Spinnaker::GenApi::CPortImpl, 713
- ~CPortWriteList
 - Spinnaker::GenApi::CPortWriteList, 717
- ~CRegisterPortImpl
 - Spinnaker::GenApi::CRegisterPortImpl, 720
- ~CSelectorSet
 - Spinnaker::GenApi::CSelectorSet, 724
- ~Camera
 - Spinnaker::Camera, 470
- ~CameraBase
 - Spinnaker::CameraBase, 593
- ~CameraList
 - Spinnaker::CameraList, 607
- ~CategoryNode
 - Spinnaker::GenApi::CategoryNode, 615
- ~ChunkData
 - Spinnaker::ChunkData, 667
- ~CommandNode
 - Spinnaker::GenApi::CommandNode, 704
- ~DeviceEvent
 - Spinnaker::DeviceEvent, 732
- ~EnumEntryNode
 - Spinnaker::GenApi::EnumEntryNode, 744
- ~EnumNode
 - Spinnaker::GenApi::EnumNode, 748
- ~Event
 - Spinnaker::Event, 757
- ~Exception
 - Spinnaker::Exception, 762
- ~FileProtocolAdapter
 - Spinnaker::GenApi::FileProtocolAdapter, 766
- ~FloatNode
 - Spinnaker::GenApi::FloatNode, 773
- ~FloatRegNode
 - Spinnaker::GenApi::FloatRegNode, 779
- ~IArrivalEvent
 - Spinnaker::IArrivalEvent, 804
- ~ICameraBase
 - Spinnaker::ICameraBase, 807
- ~ICameraList
 - Spinnaker::ICameraList, 815
- ~IChunkData
 - Spinnaker::IChunkData, 818
- ~IDataStream
 - Spinnaker::IDataStream, 826
- ~IDevFileStreamBuf
 - Spinnaker::GenApi::IDevFileStreamBuf, 833
- ~IDeviceEvent
 - Spinnaker::IDeviceEvent, 835
- ~IImage
 - Spinnaker::IImage, 838
- ~IImageEvent
 - Spinnaker::IImageEvent, 849
- ~IImageStatistics
 - Spinnaker::IImageStatistics, 851
- ~IInterface
 - Spinnaker::IInterface, 855
- ~IInterfaceEvent
 - Spinnaker::IInterfaceEvent, 859
- ~IInterfaceList
 - Spinnaker::IInterfaceList, 861
- ~ILoggingEvent
 - Spinnaker::ILoggingEvent, 864
- ~IRemovalEvent
 - Spinnaker::IRemovalEvent, 947
- ~ISystem
 - Spinnaker::ISystem, 949
- ~Image
 - Spinnaker::Image, 869
- ~ImageEvent
 - Spinnaker::ImageEvent, 895
- ~ImagePtr
 - Spinnaker::ImagePtr, 897
- ~ImageStatistics
 - Spinnaker::ImageStatistics, 900
- ~IntRegNode
 - Spinnaker::GenApi::IntRegNode, 945
- ~IntegerNode
 - Spinnaker::GenApi::IntegerNode, 926
- ~Interface
 - Spinnaker::Interface, 931
- ~InterfaceEvent
 - Spinnaker::InterfaceEvent, 936

- ~InterfaceList
 - Spinnaker::InterfaceList, [939](#)
- ~Lock
 - Spinnaker::GenICam::LockableObject::Lock, [958](#)
- ~LoggingEvent
 - Spinnaker::LoggingEvent, [961](#)
- ~LoggingEventData
 - Spinnaker::LoggingEventData, [963](#)
- ~Node
 - Spinnaker::GenApi::Node, [974](#)
- ~NodeMap
 - Spinnaker::GenApi::NodeMap, [985](#)
- ~ODevFileStreamBuf
 - Spinnaker::GenApi::ODevFileStreamBuf, [995](#)
- ~PortNode
 - Spinnaker::GenApi::PortNode, [1001](#)
- ~PortRecorder
 - Spinnaker::GenApi::PortRecorder, [1006](#)
- ~PortReplay
 - Spinnaker::GenApi::PortReplay, [1008](#)
- ~RegisterNode
 - Spinnaker::GenApi::RegisterNode, [1013](#)
- ~RemovalEvent
 - Spinnaker::RemovalEvent, [1016](#)
- ~SpinVideo
 - Spinnaker::Video::SpinVideo, [1020](#)
- ~StringNode
 - Spinnaker::GenApi::StringNode, [1026](#)
- ~StringRegNode
 - Spinnaker::GenApi::StringRegNode, [1030](#)
- ~System
 - Spinnaker::System, [1032](#)
- ~SystemPtr
 - Spinnaker::SystemPtr, [1042](#)
- ~TransportLayerDevice
 - Spinnaker::TransportLayerDevice, [1046](#)
- ~TransportLayerInterface
 - Spinnaker::TransportLayerInterface, [1056](#)
- ~TransportLayerStream
 - Spinnaker::TransportLayerStream, [1065](#)
- ~TransportLayerSystem
 - Spinnaker::TransportLayerSystem, [1071](#)
- ~ValueNode
 - Spinnaker::GenApi::ValueNode, [1077](#)
- ~double_autovector_t
 - Spinnaker::GenApi::double_autovector_t, [734](#)
- ~gcstring
 - Spinnaker::GenICam::gcstring, [784](#)
- ~int64_autovector_t
 - Spinnaker::GenApi::int64_autovector_t, [922](#)
- aPAUSEMACCtrlFramesReceived
 - Spinnaker::Camera, [475](#)
- aPAUSEMACCtrlFramesTransmitted
 - Spinnaker::Camera, [475](#)
- AVI Recorder Class, [33](#)
 - DEPRECATED_CLASS, [33](#)
- AVIOption, [430](#)
 - Spinnaker::Video::AVIOption, [430](#)
- AasRoiEnable
 - Spinnaker::Camera, [470](#)
- AasRoiHeight
 - Spinnaker::Camera, [470](#)
- AasRoiOffsetX
 - Spinnaker::Camera, [471](#)
- AasRoiOffsetY
 - Spinnaker::Camera, [471](#)
- AasRoiWidth
 - Spinnaker::Camera, [471](#)
- AcquisitionAbort
 - Spinnaker::Camera, [471](#)
- AcquisitionArm
 - Spinnaker::Camera, [472](#)
- AcquisitionBurstFrameCount
 - Spinnaker::Camera, [472](#)
- AcquisitionFrameCount
 - Spinnaker::Camera, [472](#)
- AcquisitionFrameRate
 - Spinnaker::Camera, [472](#)
- AcquisitionFrameRateEnable
 - Spinnaker::Camera, [472](#)
- AcquisitionLineRate
 - Spinnaker::Camera, [473](#)
- AcquisitionMode
 - Spinnaker::Camera, [473](#)
- AcquisitionModeEnums
 - CameraDefs Class, [71](#)
- AcquisitionResultingFrameRate
 - Spinnaker::Camera, [473](#)
- AcquisitionStart
 - Spinnaker::Camera, [473](#)
- AcquisitionStatus
 - Spinnaker::Camera, [473](#)
- AcquisitionStatusSelector
 - Spinnaker::Camera, [473](#)
- AcquisitionStatusSelectorEnums
 - CameraDefs Class, [71](#)
- AcquisitionStop
 - Spinnaker::Camera, [474](#)
- ActionCommand
 - Spinnaker::TransportLayerInterface, [1056](#)
- ActionCommandResult, [425](#)
- ActionCommandStatus
 - Spinnaker Definitions, [182](#)
- ActionDeviceKey
 - Spinnaker::Camera, [474](#)
- ActionGroupKey
 - Spinnaker::Camera, [474](#)
- ActionGroupMask
 - Spinnaker::Camera, [474](#)
- ActionQueueSize
 - Spinnaker::Camera, [474](#)
- ActionSelector
 - Spinnaker::Camera, [474](#)
- ActionUnconditionalMode
 - Spinnaker::Camera, [475](#)
- ActionUnconditionalModeEnums

- CameraDefs Class, [72](#)
- AdaptiveCompressionEnable
 - Spinnaker::Camera, [475](#)
- AdcBitDepth
 - Spinnaker::Camera, [475](#)
- AdcBitDepthEnums
 - CameraDefs Class, [72](#)
- AddChunks
 - Spinnaker::IDataStream, [826](#)
- AddInjectionData
 - Spinnaker::GenApi::CNodeMapFactory, [690](#)
- Address
 - IPort Interface, [308](#)
- AnnouncelImage
 - Spinnaker::IDataStream, [826](#)
- Append
 - Spinnaker::CameraList, [608](#)
 - Spinnaker::ICameraList, [815](#)
 - Spinnaker::Video::SpinVideo, [1020](#)
- append
 - Spinnaker::GenICam::gcstring, [784](#)
- ApplyStyleSheet
 - Spinnaker::GenApi::CNodeMapFactory, [690](#)
- ArrivalEvent, [426](#)
 - Spinnaker::ArrivalEvent, [427](#)
- ArrivalEvent Class, [29](#)
- assign
 - Spinnaker::GenICam::gcstring, [784](#), [785](#)
- attach
 - Spinnaker::GenApi::FileProtocolAdapter, [767](#)
- AttachBuffer
 - Spinnaker::GenApi::CChunkAdapter, [617](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [620](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [625](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [623](#)
 - Spinnaker::GenApi::CChunkAdapterU3V, [627](#)
- AttachChunk
 - Spinnaker::GenApi::CChunkPort, [630](#)
- AttachEvent
 - Spinnaker::GenApi::CEventPort, [651](#)
- AttachNode
 - Spinnaker::GenApi::CEventPort, [651](#)
- AttachNodeMap
 - Spinnaker::GenApi::CChunkAdapter, [617](#)
 - Spinnaker::GenApi::CEventAdapter, [639](#)
- AttachPort
 - Spinnaker::GenApi::CChunkPort, [630](#)
- AttachStatistics_t, [428](#)
 - NumAttachedChunks, [428](#)
 - NumChunkPorts, [428](#)
 - NumChunks, [428](#)
- AutoAlgorithmSelector
 - Spinnaker::Camera, [476](#)
- AutoAlgorithmSelectorEnums
 - CameraDefs Class, [72](#)
- AutoExposureControlLoopDamping
 - Spinnaker::Camera, [476](#)
- AutoExposureControlPriority
 - Spinnaker::Camera, [476](#)
- AutoExposureControlPriorityEnums
 - CameraDefs Class, [73](#)
- AutoExposureEVCompensation
 - Spinnaker::Camera, [476](#)
- AutoExposureExposureTimeLowerLimit
 - Spinnaker::Camera, [477](#)
- AutoExposureExposureTimeUpperLimit
 - Spinnaker::Camera, [477](#)
- AutoExposureGainLowerLimit
 - Spinnaker::Camera, [477](#)
- AutoExposureGainUpperLimit
 - Spinnaker::Camera, [477](#)
- AutoExposureGreyValueLowerLimit
 - Spinnaker::Camera, [478](#)
- AutoExposureGreyValueUpperLimit
 - Spinnaker::Camera, [478](#)
- AutoExposureLightingMode
 - Spinnaker::Camera, [478](#)
- AutoExposureLightingModeEnums
 - CameraDefs Class, [73](#)
- AutoExposureMeteringMode
 - Spinnaker::Camera, [478](#)
- AutoExposureMeteringModeEnums
 - CameraDefs Class, [73](#)
- AutoExposureTargetGreyValue
 - Spinnaker::Camera, [479](#)
- AutoExposureTargetGreyValueAuto
 - Spinnaker::Camera, [479](#)
- AutoExposureTargetGreyValueAutoEnums
 - CameraDefs Class, [74](#)
- AutoForceIP
 - Spinnaker::TransportLayerInterface, [1057](#)
- AutoLock, [429](#)
 - Spinnaker::GenApi::AutoLock, [429](#)
 - Spinnaker::GenICam::AutoLock, [430](#)
- AutoVector Class, [225](#)
- Automatic
 - Types Enums, [360](#)
- BMPOption, [435](#)
 - Spinnaker::BMPOption, [436](#)
- BalanceRatio
 - Spinnaker::Camera, [479](#)
- BalanceRatioSelector
 - Spinnaker::Camera, [480](#)
- BalanceRatioSelectorEnums
 - CameraDefs Class, [74](#)
- BalanceWhiteAuto
 - Spinnaker::Camera, [480](#)
- BalanceWhiteAutoDamping
 - Spinnaker::Camera, [480](#)
- BalanceWhiteAutoEnums
 - CameraDefs Class, [75](#)
- BalanceWhiteAutoLowerLimit
 - Spinnaker::Camera, [480](#)
- BalanceWhiteAutoProfile
 - Spinnaker::Camera, [481](#)
- BalanceWhiteAutoProfileEnums

- CameraDefs Class, [75](#)
- BalanceWhiteAutoUpperLimit
 - Spinnaker::Camera, [481](#)
- BasePtr
 - Spinnaker::BasePtr, [432](#)
- BasePtr Class, [36](#)
- BasePtr< T, B >, [431](#)
- BeginAcquisition
 - Spinnaker::CameraBase, [594](#)
 - Spinnaker::ICameraBase, [807](#)
- Beginner
 - Types Enums, [361](#)
- binaryFile
 - Spinnaker::PGMOption, [997](#)
 - Spinnaker::PPMOption, [1010](#)
- BinningHorizontal
 - Spinnaker::Camera, [481](#)
- BinningHorizontalMode
 - Spinnaker::Camera, [481](#)
- BinningHorizontalModeEnums
 - CameraDefs Class, [75](#)
- BinningSelector
 - Spinnaker::Camera, [482](#)
- BinningSelectorEnums
 - CameraDefs Class, [76](#)
- BinningVertical
 - Spinnaker::Camera, [482](#)
- BinningVerticalMode
 - Spinnaker::Camera, [482](#)
- BinningVerticalModeEnums
 - CameraDefs Class, [76](#)
- bitrate
 - Spinnaker::Video::H264Option, [801](#)
- BlackLevel
 - Spinnaker::Camera, [482](#)
- BlackLevelAuto
 - Spinnaker::Camera, [482](#)
- BlackLevelAutoBalance
 - Spinnaker::Camera, [483](#)
- BlackLevelAutoBalanceEnums
 - CameraDefs Class, [76](#)
- BlackLevelAutoEnums
 - CameraDefs Class, [77](#)
- BlackLevelClampingEnable
 - Spinnaker::Camera, [483](#)
- BlackLevelRaw
 - Spinnaker::Camera, [483](#)
- BlackLevelSelector
 - Spinnaker::Camera, [483](#)
- BlackLevelSelectorEnums
 - CameraDefs Class, [77](#)
- BlockId
 - GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
 - GVCP_EVENT_ITEM, [792](#)
- BlockId64High
 - GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
- BlockId64Low
 - GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
- Boolean
 - Types Enums, [359](#)
- BooleanNode, [437](#)
 - Spinnaker::GenApi::BooleanNode, [438](#)
- BooleanNode Class, [229](#)
 - CBooleanRef, [229](#)
- BufferOwnership
 - Spinnaker Definitions, [183](#)
- build
 - Spinnaker::LibraryVersion, [956](#)
- c_str
 - Spinnaker::GenICam::gcstring, [785](#)
- CBasePtr
 - Pointer Class, [332](#)
- CBooleanPtr
 - Pointer Class, [332](#)
- CBooleanRef
 - BooleanNode Class, [229](#)
- CCategoryPtr
 - Pointer Class, [333](#)
- CCategoryRef
 - CategoryNode Class, [230](#)
- CChunkAdapter, [615](#)
 - Spinnaker::GenApi::CChunkAdapter, [616](#)
- CChunkAdapterDcam, [619](#)
 - Spinnaker::GenApi::CChunkAdapterDcam, [620](#)
- CChunkAdapterGEV, [624](#)
 - Spinnaker::GenApi::CChunkAdapterGEV, [625](#)
- CChunkAdapterGeneric, [621](#)
 - Spinnaker::GenApi::CChunkAdapterGeneric, [622](#)
- CChunkAdapterU3V, [626](#)
 - Spinnaker::GenApi::CChunkAdapterU3V, [627](#)
- CChunkPort, [628](#)
 - Spinnaker::GenApi::CChunkPort, [629](#)
- CChunkPortPtr
 - Pointer Class, [333](#)
- CCommandPtr
 - Pointer Class, [333](#)
- CCommandRef
 - CommandNode Class, [236](#)
- CDeviceInfoPtr
 - Pointer Class, [333](#)
- CEnumEntryPtr
 - Pointer Class, [333](#)
- CEnumEntryRef
 - EnumEntryNode Class, [241](#)
- CEnumerationPtr
 - Pointer Class, [333](#)
- CEnumerationRef
 - EnumNode Class, [242](#)
- CEnumerationTRef
 - Spinnaker::GenApi::CEnumerationTRef, [635](#)
- CEnumerationTRef< EnumT >, [633](#)
- CEventAdapter, [638](#)
 - Spinnaker::GenApi::CEventAdapter, [639](#)
- CEventAdapter1394, [640](#)
 - Spinnaker::GenApi::CEventAdapter1394, [641](#)
- CEventAdapterGEV, [645](#)

- Spinnaker::GenApi::CEventAdapterGEV, 646
- CEventAdapterGeneric, 642
 - Spinnaker::GenApi::CEventAdapterGeneric, 643
- CEventAdapterU3V, 647
 - Spinnaker::GenApi::CEventAdapterU3V, 648
- CEventPort, 649
 - Spinnaker::GenApi::CEventPort, 650
- CFeatureBag, 654
 - Spinnaker::GenApi::CFeatureBag, 655
- CFloatPtr, 657
 - Spinnaker::GenApi::CFloatPtr, 658
- CFloatRef
 - FloatNode Class, 251
- CGeneric_XMLLoaderParams, 659
- CGlobalLock, 660
 - Spinnaker::GenICam::CGlobalLock, 660, 661
- CGlobalLockUnlocker, 662
 - Spinnaker::GenICam::CGlobalLockUnlocker, 663
- CHUNK_BASE_ADDRESS_REGISTER_LEN
 - ICChunkPort Interface, 266
- CHUNK_BASE_ADDRESS_REGISTER
 - ICChunkPort Interface, 266
- CHUNK_LENGTH_REGISTER_LEN
 - ICChunkPort Interface, 267
- CHUNK_LENGTH_REGISTER
 - ICChunkPort Interface, 267
- CIntegerPtr
 - Pointer Class, 334
- CIntegerRef
 - IntegerNode Class, 306
- CLock, 676, 679
 - Spinnaker::GenApi::CLock, 677
 - Spinnaker::GenICam::CLock, 679
- CLockEx, 681, 682
- CNodeCallback, 683
 - Spinnaker::GenApi::CNodeCallback, 684
- CNodeMapDynPtr
 - Pointer Class, 334
- CNodeMapFactory, 686
 - Spinnaker::GenApi::CNodeMapFactory, 688, 689
- CNodeMapFactory::NodeStatistics_t, 991
- CNodeMapPtr
 - Pointer Class, 334
- CNodeMapRef, 694
 - Spinnaker GenApi Classes, 220
 - Spinnaker::GenApi::CNodeMapRef, 695, 696
- CNodeMapRefT< TCameraParams >, 697
- CNodeMapRefT
 - Spinnaker GenApi Classes, 223
- CNodePtr
 - Pointer Class, 334
- CNodeRef
 - Spinnaker GenApi Classes, 220
- COMMAND_MAGIC
 - Spinnaker::GenApi, 420
- CPointer
 - Spinnaker::GenApi::CPointer, 708
- CPointer< T, B >, 707
- CPortConstructPtr
 - Pointer Class, 334
- CPortImpl, 712
 - Spinnaker::GenApi::CPortImpl, 713
- CPortPtr
 - Pointer Class, 334
- CPortRecorderPtr
 - Pointer Class, 335
- CPortRecorderRef
 - PortRecorder Class, 339
- CPortRef
 - PortNode Class, 338
- CPortReplayPtr
 - Pointer Class, 335
- CPortWriteList, 716
 - Spinnaker::GenApi::CPortWriteList, 717
- CPortWriteListPtr
 - Pointer Class, 335
- CRCChecksum
 - DCAM_CHECKSUM, 729
- CRegisterPortImpl, 719
 - Spinnaker::GenApi::CRegisterPortImpl, 720
- CRegisterPtr
 - Pointer Class, 335
- CRegisterRef
 - RegisterNode Class, 343
- CSelectorPtr
 - Pointer Class, 335
- CSelectorRef
 - Spinnaker GenApi Classes, 220
- CSelectorSet, 722
 - Spinnaker::GenApi::CSelectorSet, 723
- CStringPtr
 - Pointer Class, 335
- CStringRef
 - StringNode Class, 347
- CTestPortStruct
 - Spinnaker::GenApi::CTestPortStruct, 727
- CTestPortStruct< CDataStruct >, 725
- CValuePtr
 - Pointer Class, 336
- CValueRef
 - ValueNode Class, 362
- CacheChunkData
 - ICChunkPort Interface, 267
 - Spinnaker::GenApi::PortNode, 1001
- CalculateStatistics
 - Spinnaker::Image, 839
 - Spinnaker::Image, 870
- CallbackHandleType
 - Spinnaker GenApi Interfaces, 227
- Camera, 440
 - Spinnaker::Camera, 470
- Camera Base Class, 38
- Camera Base Interface Class, 207
- Camera Class, 37
- Camera List Class, 154
- CameraBase, 591

- Spinnaker::CameraBase, [593](#), [594](#)
- Spinnaker::TransportLayerDevice, [1046](#)
- Spinnaker::TransportLayerStream, [1065](#)
- CameraDefs Class, [39](#)
 - AcquisitionModeEnums, [71](#)
 - AcquisitionStatusSelectorEnums, [71](#)
 - ActionUnconditionalModeEnums, [72](#)
 - AdcBitDepthEnums, [72](#)
 - AutoAlgorithmSelectorEnums, [72](#)
 - AutoExposureControlPriorityEnums, [73](#)
 - AutoExposureLightingModeEnums, [73](#)
 - AutoExposureMeteringModeEnums, [73](#)
 - AutoExposureTargetGreyValueAutoEnums, [74](#)
 - BalanceRatioSelectorEnums, [74](#)
 - BalanceWhiteAutoEnums, [75](#)
 - BalanceWhiteAutoProfileEnums, [75](#)
 - BinningHorizontalModeEnums, [75](#)
 - BinningSelectorEnums, [76](#)
 - BinningVerticalModeEnums, [76](#)
 - BlackLevelAutoBalanceEnums, [76](#)
 - BlackLevelAutoEnums, [77](#)
 - BlackLevelSelectorEnums, [77](#)
 - ChunkBlackLevelSelectorEnums, [77](#)
 - ChunkCounterSelectorEnums, [77](#)
 - ChunkEncoderSelectorEnums, [78](#)
 - ChunkEncoderStatusEnums, [78](#)
 - ChunkExposureTimeSelectorEnums, [78](#)
 - ChunkGainSelectorEnums, [79](#)
 - ChunkImageComponentEnums, [79](#)
 - ChunkPixelFormatEnums, [80](#)
 - ChunkRegionIDEnums, [80](#)
 - ChunkScan3dCoordinateReferenceSelector↔
Enums, [80](#)
 - ChunkScan3dCoordinateSelectorEnums, [81](#)
 - ChunkScan3dCoordinateSystemEnums, [81](#)
 - ChunkScan3dCoordinateSystemReferenceEnums,
[81](#)
 - ChunkScan3dCoordinateTransformSelector↔
Enums, [82](#)
 - ChunkScan3dDistanceUnitEnums, [82](#)
 - ChunkScan3dOutputModeEnums, [83](#)
 - ChunkSelectorEnums, [83](#)
 - ChunkSourceIDEnums, [84](#)
 - ChunkTimerSelectorEnums, [84](#)
 - ChunkTransferStreamIDEnums, [85](#)
 - CIConfigurationEnums, [85](#)
 - CITimeSlotsCountEnums, [85](#)
 - ColorTransformationSelectorEnums, [86](#)
 - ColorTransformationValueSelectorEnums, [86](#)
 - CounterEventActivationEnums, [87](#)
 - CounterEventSourceEnums, [87](#)
 - CounterResetActivationEnums, [88](#)
 - CounterResetSourceEnums, [88](#)
 - CounterSelectorEnums, [88](#)
 - CounterStatusEnums, [89](#)
 - CounterTriggerActivationEnums, [89](#)
 - CounterTriggerSourceEnums, [89](#)
 - CxpConnectionTestModeEnums, [90](#)
 - CxpLinkConfigurationEnums, [90](#)
 - CxpLinkConfigurationPreferredEnums, [91](#)
 - CxpLinkConfigurationStatusEnums, [92](#)
 - CxpPoCxpStatusEnums, [93](#)
 - DecimationHorizontalModeEnums, [94](#)
 - DecimationSelectorEnums, [94](#)
 - DecimationVerticalModeEnums, [94](#)
 - DefectCorrectionModeEnums, [94](#)
 - DeinterlacingEnums, [95](#)
 - DeviceCharacterSetEnums, [95](#)
 - DeviceClockSelectorEnums, [95](#)
 - DeviceConnectionStatusEnums, [96](#)
 - DeviceIndicatorModeEnums, [96](#)
 - DeviceLinkHeartbeatModeEnums, [96](#)
 - DeviceLinkThroughputLimitModeEnums, [98](#)
 - DevicePowerSupplySelectorEnums, [98](#)
 - DeviceRegistersEndiannessEnums, [98](#)
 - DeviceScanTypeEnums, [99](#)
 - DeviceSerialPortBaudRateEnums, [99](#)
 - DeviceSerialPortSelectorEnums, [99](#)
 - DeviceStreamChannelEndiannessEnums, [99](#)
 - DeviceStreamChannelTypeEnums, [100](#)
 - DeviceTLTypeEnums, [102](#)
 - DeviceTapGeometryEnums, [100](#)
 - DeviceTemperatureSelectorEnums, [101](#)
 - DeviceTypeEnums, [102](#)
 - EncoderModeEnums, [102](#)
 - EncoderOutputModeEnums, [103](#)
 - EncoderResetActivationEnums, [103](#)
 - EncoderResetSourceEnums, [104](#)
 - EncoderSelectorEnums, [105](#)
 - EncoderSourceAEnums, [105](#)
 - EncoderSourceBEnums, [105](#)
 - EncoderStatusEnums, [106](#)
 - EventNotificationEnums, [106](#)
 - EventSelectorEnums, [106](#)
 - ExposureActiveModeEnums, [107](#)
 - ExposureAutoEnums, [107](#)
 - ExposureModeEnums, [107](#)
 - ExposureTimeModeEnums, [108](#)
 - ExposureTimeSelectorEnums, [108](#)
 - FileOpenModeEnums, [109](#)
 - FileOperationSelectorEnums, [109](#)
 - FileOperationStatusEnums, [109](#)
 - FileSelectorEnums, [110](#)
 - GainAutoBalanceEnums, [110](#)
 - GainAutoEnums, [110](#)
 - GainSelectorEnums, [111](#)
 - GevCCPEnums, [111](#)
 - GevCurrentPhysicalLinkConfigurationEnums, [111](#)
 - GevGVCPExtendedStatusCodesSelectorEnums,
[111](#)
 - GevGVSPExtendedIDModeEnums, [112](#)
 - GevIEEE1588ClockAccuracyEnums, [112](#)
 - GevIEEE1588ModeEnums, [112](#)
 - GevIEEE1588StatusEnums, [113](#)
 - GevIPConfigurationStatusEnums, [113](#)
 - GevPhysicalLinkConfigurationEnums, [113](#)

- GevSupportedOptionSelectorEnums, 114
- ImageComponentSelectorEnums, 115
- ImageCompressionJPEGFormatOptionEnums, 115
- ImageCompressionModeEnums, 116
- ImageCompressionRateOptionEnums, 116
- LUTSelectorEnums, 120
- LineFormatEnums, 116
- LineInputFilterSelectorEnums, 117
- LineModeEnums, 117
- LineSelectorEnums, 117
- LineSourceEnums, 118
- LogicBlockLUTInputActivationEnums, 118
- LogicBlockLUTInputSelectorEnums, 119
- LogicBlockLUTInputSourceEnums, 119
- LogicBlockLUTSelectorEnums, 120
- LogicBlockSelectorEnums, 120
- PixelColorFilterEnums, 121
- PixelFormatEnums, 121
- PixelFormatInfoSelectorEnums, 127
- PixelSizeEnums, 132
- RegionDestinationEnums, 133
- RegionModeEnums, 133
- RegionSelectorEnums, 134
- RgbTransformLightSourceEnums, 134
- Scan3dCoordinateReferenceSelectorEnums, 134
- Scan3dCoordinateSelectorEnums, 135
- Scan3dCoordinateSystemEnums, 135
- Scan3dCoordinateSystemReferenceEnums, 135
- Scan3dCoordinateTransformSelectorEnums, 136
- Scan3dDistanceUnitEnums, 136
- Scan3dOutputModeEnums, 137
- SensorDigitizationTapsEnums, 137
- SensorShutterModeEnums, 138
- SensorTapsEnums, 138
- SequencerConfigurationModeEnums, 139
- SequencerConfigurationValidEnums, 139
- SequencerModeEnums, 139
- SequencerSetValidEnums, 139
- SequencerTriggerActivationEnums, 140
- SequencerTriggerSourceEnums, 140
- SerialPortBaudRateEnums, 140
- SerialPortParityEnums, 141
- SerialPortSelectorEnums, 141
- SerialPortSourceEnums, 142
- SerialPortStopBitsEnums, 142
- SoftwareSignalSelectorEnums, 142
- SourceSelectorEnums, 143
- TestPatternEnums, 143
- TestPatternGeneratorSelectorEnums, 143
- TimerSelectorEnums, 144
- TimerStatusEnums, 144
- TimerTriggerActivationEnums, 144
- TimerTriggerSourceEnums, 145
- TransferComponentSelectorEnums, 146
- TransferControlModeEnums, 146
- TransferOperationModeEnums, 147
- TransferQueueModeEnums, 147
- TransferSelectorEnums, 147
- TransferStatusSelectorEnums, 148
- TransferTriggerActivationEnums, 148
- TransferTriggerModeEnums, 148
- TransferTriggerSelectorEnums, 149
- TransferTriggerSourceEnums, 149
- TriggerActivationEnums, 150
- TriggerModeEnums, 151
- TriggerOverlapEnums, 151
- TriggerSelectorEnums, 151
- TriggerSourceEnums, 151
- UserOutputSelectorEnums, 152
- UserSetDefaultEnums, 152
- UserSetSelectorEnums, 153
- WhiteClipSelectorEnums, 153
- CameraInternal
 - Spinnaker::ICameraBase, 813
 - Spinnaker::TransportLayerDevice, 1046
 - Spinnaker::TransportLayerStream, 1065
- CameraList, 606
 - Spinnaker::CameraList, 607
- CameraListImpl
 - Spinnaker::ICameraList, 817
- CameraPtr, 612
 - CameraPtr Class, 155, 156
- CameraPtr Class, 155
 - CameraPtr, 155, 156
- CastToIDestroy
 - Spinnaker GenApi Classes, 223
- CategoryNode, 613
 - Spinnaker::GenApi::CategoryNode, 614
- CategoryNode Class, 230
 - CCategoryRef, 230
- CheckBufferLayout
 - Spinnaker::GenApi::CChunkAdapter, 617
 - Spinnaker::GenApi::CChunkAdapterDcam, 620
 - Spinnaker::GenApi::CChunkAdapterGEV, 625
 - Spinnaker::GenApi::CChunkAdapterGeneric, 623
 - Spinnaker::GenApi::CChunkAdapterU3V, 627
- CheckCRC
 - Spinnaker::GenApi::CChunkAdapterDcam, 621
 - Spinnaker::Image, 839
 - Spinnaker::Image, 871
- CheckChunkID
 - Spinnaker::GenApi::CChunkPort, 630
- CheckEventID
 - Spinnaker::GenApi::CEventPort, 651
- ChunkAdapter Class, 231
- ChunkAdapterDcam Class, 232
- ChunkAdapterGEV Class, 234
- ChunkAdapterGeneric Class, 233
- ChunkAdapterU3V Class, 363
- ChunkBlackLevel
 - Spinnaker::Camera, 483
- ChunkBlackLevelSelector
 - Spinnaker::Camera, 484
- ChunkBlackLevelSelectorEnums
 - CameraDefs Class, 77

- ChunkCRC
 - Spinnaker::Camera, [484](#)
- ChunkCounterSelector
 - Spinnaker::Camera, [484](#)
- ChunkCounterSelectorEnums
 - CameraDefs Class, [77](#)
- ChunkCounterValue
 - Spinnaker::Camera, [484](#)
- ChunkData, [664](#)
 - Spinnaker::ChunkData, [666](#)
- ChunkData Class, [157](#)
- ChunkEnable
 - Spinnaker::Camera, [484](#)
- ChunkEncoderSelector
 - Spinnaker::Camera, [484](#)
- ChunkEncoderSelectorEnums
 - CameraDefs Class, [78](#)
- ChunkEncoderStatus
 - Spinnaker::Camera, [485](#)
- ChunkEncoderStatusEnums
 - CameraDefs Class, [78](#)
- ChunkEncoderValue
 - Spinnaker::Camera, [485](#)
- ChunkExposureEndLineStatusAll
 - Spinnaker::Camera, [485](#)
- ChunkExposureTime
 - Spinnaker::Camera, [485](#)
- ChunkExposureTimeSelector
 - Spinnaker::Camera, [485](#)
- ChunkExposureTimeSelectorEnums
 - CameraDefs Class, [78](#)
- ChunkFrameID
 - Spinnaker::Camera, [485](#)
- ChunkGain
 - Spinnaker::Camera, [486](#)
- ChunkGainSelector
 - Spinnaker::Camera, [486](#)
- ChunkGainSelectorEnums
 - CameraDefs Class, [79](#)
- ChunkHeight
 - Spinnaker::Camera, [486](#)
- ChunkID
 - DCAM_CHUNK_TRAILER, [730](#)
 - GVCP_CHUNK_TRAILER, [791](#)
 - SingleChunkData_t, [1017](#)
 - SingleChunkDataStr_t, [1018](#)
 - U3V_CHUNK_TRAILER, [1072](#)
- ChunkImage
 - Spinnaker::Camera, [486](#)
- ChunkImageComponent
 - Spinnaker::Camera, [486](#)
- ChunkImageComponentEnums
 - CameraDefs Class, [79](#)
- ChunkInferenceConfidence
 - Spinnaker::Camera, [486](#)
- ChunkInferenceResult
 - Spinnaker::Camera, [487](#)
- ChunkLength
 - DCAM_CHUNK_TRAILER, [730](#)
 - GVCP_CHUNK_TRAILER, [791](#)
 - SingleChunkData_t, [1017](#)
 - SingleChunkDataStr_t, [1018](#)
 - U3V_CHUNK_TRAILER, [1072](#)
- ChunkLinePitch
 - Spinnaker::Camera, [487](#)
- ChunkLineStatusAll
 - Spinnaker::Camera, [487](#)
- ChunkModeActive
 - Spinnaker::Camera, [487](#)
- ChunkOffset
 - SingleChunkData_t, [1017](#)
 - SingleChunkDataStr_t, [1018](#)
- ChunkOffsetX
 - Spinnaker::Camera, [487](#)
- ChunkOffsetY
 - Spinnaker::Camera, [487](#)
- ChunkPartSelector
 - Spinnaker::Camera, [488](#)
- ChunkPixelDynamicRangeMax
 - Spinnaker::Camera, [488](#)
- ChunkPixelDynamicRangeMin
 - Spinnaker::Camera, [488](#)
- ChunkPixelFormat
 - Spinnaker::Camera, [488](#)
- ChunkPixelFormatEnums
 - CameraDefs Class, [80](#)
- ChunkPort Class, [235](#)
- ChunkRegionIDEnums
 - CameraDefs Class, [80](#)
- ChunkRegionID
 - Spinnaker::Camera, [488](#)
- ChunkScan3dAxisMax
 - Spinnaker::Camera, [488](#)
- ChunkScan3dAxisMin
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateOffset
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateReferenceSelector
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateReferenceSelectorEnums
 - CameraDefs Class, [80](#)
- ChunkScan3dCoordinateReferenceValue
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateScale
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateSelector
 - Spinnaker::Camera, [489](#)
- ChunkScan3dCoordinateSelectorEnums
 - CameraDefs Class, [81](#)
- ChunkScan3dCoordinateSystem
 - Spinnaker::Camera, [490](#)
- ChunkScan3dCoordinateSystemEnums
 - CameraDefs Class, [81](#)
- ChunkScan3dCoordinateSystemReference
 - Spinnaker::Camera, [490](#)
- ChunkScan3dCoordinateSystemReferenceEnums

- CameraDefs Class, [81](#)
- ChunkScan3dCoordinateTransformSelector
 - Spinnaker::Camera, [490](#)
- ChunkScan3dCoordinateTransformSelectorEnums
 - CameraDefs Class, [82](#)
- ChunkScan3dDistanceUnit
 - Spinnaker::Camera, [490](#)
- ChunkScan3dDistanceUnitEnums
 - CameraDefs Class, [82](#)
- ChunkScan3dInvalidDataFlag
 - Spinnaker::Camera, [490](#)
- ChunkScan3dInvalidDataValue
 - Spinnaker::Camera, [490](#)
- ChunkScan3dOutputMode
 - Spinnaker::Camera, [491](#)
- ChunkScan3dOutputModeEnums
 - CameraDefs Class, [83](#)
- ChunkScan3dTransformValue
 - Spinnaker::Camera, [491](#)
- ChunkScanLineSelector
 - Spinnaker::Camera, [491](#)
- ChunkSelector
 - Spinnaker::Camera, [491](#)
- ChunkSelectorEnums
 - CameraDefs Class, [83](#)
- ChunkSequencerSetActive
 - Spinnaker::Camera, [491](#)
- ChunkSerialData
 - Spinnaker::Camera, [491](#)
- ChunkSerialDataLength
 - Spinnaker::Camera, [492](#)
- ChunkSerialReceiveOverflow
 - Spinnaker::Camera, [492](#)
- ChunkSourceIDenums
 - CameraDefs Class, [84](#)
- ChunkSourceID
 - Spinnaker::Camera, [492](#)
- ChunkStreamChannelID
 - Spinnaker::Camera, [492](#)
- ChunkTimerSelector
 - Spinnaker::Camera, [492](#)
- ChunkTimerSelectorEnums
 - CameraDefs Class, [84](#)
- ChunkTimerValue
 - Spinnaker::Camera, [492](#)
- ChunkTimestamp
 - Spinnaker::Camera, [493](#)
- ChunkTimestampLatchValue
 - Spinnaker::Camera, [493](#)
- ChunkTransferBlockID
 - Spinnaker::Camera, [493](#)
- ChunkTransferQueueCurrentBlockCount
 - Spinnaker::Camera, [493](#)
- ChunkTransferStreamIDenums
 - CameraDefs Class, [85](#)
- ChunkTransferStreamID
 - Spinnaker::Camera, [493](#)
- ChunkWidth
 - Spinnaker::Camera, [493](#)
- CL
 - Types Enums, [360](#)
- CIConfiguration
 - Spinnaker::Camera, [494](#)
- CIConfigurationEnums
 - CameraDefs Class, [85](#)
- CITimeSlotsCount
 - Spinnaker::Camera, [494](#)
- CITimeSlotsCountEnums
 - CameraDefs Class, [85](#)
- CleanupChunkAdapter
 - Spinnaker::IDataStream, [827](#)
- Clear
 - Spinnaker::CameraList, [609](#)
 - Spinnaker::ICameraList, [815](#)
 - Spinnaker::IInterfaceList, [862](#)
 - Spinnaker::InterfaceList, [939](#)
- ClearCache
 - Spinnaker::GenApi::CChunkPort, [630](#)
 - Spinnaker::GenApi::CNodeMapFactory, [690](#)
- ClearCaches
 - Spinnaker::GenApi::CChunkAdapter, [617](#)
- ClearXMLCache
 - Spinnaker::GenApi::NodeMap, [985](#)
- Close
 - Spinnaker::Video::SpinVideo, [1021](#)
- close
 - Spinnaker::GenApi::IDevFileStreamBase, [831](#)
 - Spinnaker::GenApi::IDevFileStreamBuf, [833](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [993](#)
 - Spinnaker::GenApi::ODevFileStreamBuf, [995](#)
- closeFile
 - Spinnaker::GenApi::FileProtocolAdapter, [768](#)
- ColorProcessingAlgorithm
 - Spinnaker Definitions, [183](#)
- ColorTransformationEnable
 - Spinnaker::Camera, [494](#)
- ColorTransformationSelector
 - Spinnaker::Camera, [494](#)
- ColorTransformationSelectorEnums
 - CameraDefs Class, [86](#)
- ColorTransformationValue
 - Spinnaker::Camera, [494](#)
- ColorTransformationValueSelector
 - Spinnaker::Camera, [495](#)
- ColorTransformationValueSelectorEnums
 - CameraDefs Class, [86](#)
- Combine
 - INode Interface, [289](#)
- Command
 - GVCP_REQUEST_HEADER, [800](#)
- CommandHeader
 - U3V_EVENT_MESSAGE, [1075](#)
- CommandId
 - U3V_COMMAND_HEADER, [1073](#)
- CommandNode, [702](#)
 - Spinnaker::GenApi::CommandNode, [703](#), [704](#)

- CommandNode Class, [236](#)
 - CCommandRef, [236](#)
- compare
 - Spinnaker::GenICam::gcstring, [785](#)
- Compatibility.h
 - FMT_I64, [1203](#)
- compression
 - Spinnaker::TIFFOption, [1043](#)
- compressionLevel
 - Spinnaker::PNGOption, [998](#)
- CompressionMethod
 - Spinnaker::TIFFOption, [1043](#)
- CompressionRatio
 - Spinnaker::Camera, [495](#)
- Connect
 - INodeMap Interface, [298](#), [299](#)
 - Spinnaker::GenApi::NodeMap, [985](#)
- Container Class, [237](#)
- Convert
 - Spinnaker::IImage, [839](#)
 - Spinnaker::Image, [871](#)
- Counter, [705](#)
 - Spinnaker::GenApi::Counter, [706](#)
- Counter Class, [238](#)
- CounterDelay
 - Spinnaker::Camera, [495](#)
- CounterDuration
 - Spinnaker::Camera, [495](#)
- CounterEventActivation
 - Spinnaker::Camera, [495](#)
- CounterEventActivationEnums
 - CameraDefs Class, [87](#)
- CounterEventSource
 - Spinnaker::Camera, [496](#)
- CounterEventSourceEnums
 - CameraDefs Class, [87](#)
- CounterReset
 - Spinnaker::Camera, [496](#)
- CounterResetActivation
 - Spinnaker::Camera, [496](#)
- CounterResetActivationEnums
 - CameraDefs Class, [88](#)
- CounterResetSource
 - Spinnaker::Camera, [496](#)
- CounterResetSourceEnums
 - CameraDefs Class, [88](#)
- CounterSelector
 - Spinnaker::Camera, [496](#)
- CounterSelectorEnums
 - CameraDefs Class, [88](#)
- CounterStatus
 - Spinnaker::Camera, [496](#)
- CounterStatusEnums
 - CameraDefs Class, [89](#)
- CounterTriggerActivation
 - Spinnaker::Camera, [497](#)
- CounterTriggerActivationEnums
 - CameraDefs Class, [89](#)
- CounterTriggerSource
 - Spinnaker::Camera, [497](#)
- CounterTriggerSourceEnums
 - CameraDefs Class, [89](#)
- CounterValue
 - Spinnaker::Camera, [497](#)
- CounterValueAtReset
 - Spinnaker::Camera, [497](#)
- Create
 - Spinnaker::Image, [871](#), [872](#)
- CreateAolp
 - Spinnaker::ImageUtilityPolarization, [915](#)
- CreateDolp
 - Spinnaker::ImageUtilityPolarization, [916](#)
- CreateEmptyNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [691](#)
- CreateHeatmap
 - Spinnaker::ImageUtilityHeatmap, [910](#), [911](#)
- CreateNodeDataFromNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [691](#)
- CreateNodeMap
 - Spinnaker::GenApi::CNodeMapFactory, [691](#)
- CreateNormalized
 - Spinnaker::ImageUtility, [906–908](#)
- CreateScaled
 - Spinnaker::ImageUtility, [908](#), [909](#)
- CreateShared
 - Spinnaker::Image, [872](#)
- CreateStokesS0
 - Spinnaker::ImageUtilityPolarization, [916](#), [917](#)
- CreateStokesS1
 - Spinnaker::ImageUtilityPolarization, [917](#), [919](#)
- CreateStokesS2
 - Spinnaker::ImageUtilityPolarization, [919](#), [920](#)
- Custom
 - Types Enums, [359](#)
- CxpConnectionSelector
 - Spinnaker::Camera, [497](#)
- CxpConnectionTestErrorCount
 - Spinnaker::Camera, [497](#)
- CxpConnectionTestMode
 - Spinnaker::Camera, [498](#)
- CxpConnectionTestModeEnums
 - CameraDefs Class, [90](#)
- CxpConnectionTestPacketCount
 - Spinnaker::Camera, [498](#)
- CxpLinkConfiguration
 - Spinnaker::Camera, [498](#)
- CxpLinkConfigurationEnums
 - CameraDefs Class, [90](#)
- CxpLinkConfigurationPreferred
 - Spinnaker::Camera, [498](#)
- CxpLinkConfigurationPreferredEnums
 - CameraDefs Class, [91](#)
- CxpLinkConfigurationStatus
 - Spinnaker::Camera, [498](#)
- CxpLinkConfigurationStatusEnums
 - CameraDefs Class, [92](#)

- CxpPoCxpAuto
 - Spinnaker::Camera, [498](#)
- CxpPoCxpStatus
 - Spinnaker::Camera, [499](#)
- CxpPoCxpStatusEnums
 - CameraDefs Class, [93](#)
- CxpPoCxpTripReset
 - Spinnaker::Camera, [499](#)
- CxpPoCxpTurnOff
 - Spinnaker::Camera, [499](#)
- DCAM_CHECKSUM, [729](#)
 - CRCChecksum, [729](#)
- DCAM_CHUNK_TRAILER, [729](#)
 - ChunkID, [730](#)
 - ChunkLength, [730](#)
 - InverseChunkLength, [730](#)
- DEPRECATED_CLASS
 - AVI Recorder Class, [33](#)
 - Spinnaker Definitions, [189](#)
- DEPRECATED_FUNC
 - Spinnaker::Image, [839](#), [840](#)
 - Spinnaker::Image, [873–876](#)
- Data
 - GVCP_EVENTDATA_REQUEST_EXTENDED_
ID, [799](#)
 - GVCP_EVENTDATA_REQUEST, [798](#)
- Delnit
 - Spinnaker::CameraBase, [594](#)
 - Spinnaker::ICameraBase, [807](#)
- DecimationHorizontal
 - Spinnaker::Camera, [499](#)
- DecimationHorizontalMode
 - Spinnaker::Camera, [499](#)
- DecimationHorizontalModeEnums
 - CameraDefs Class, [94](#)
- DecimationSelector
 - Spinnaker::Camera, [500](#)
- DecimationSelectorEnums
 - CameraDefs Class, [94](#)
- DecimationVertical
 - Spinnaker::Camera, [500](#)
- DecimationVerticalMode
 - Spinnaker::Camera, [500](#)
- DecimationVerticalModeEnums
 - CameraDefs Class, [94](#)
- Decreasing
 - Types Enums, [360](#)
- DeepCopy
 - Spinnaker::Image, [839](#)
 - Spinnaker::Image, [872](#), [873](#)
- DefectCorrectStaticEnable
 - Spinnaker::Camera, [501](#)
- DefectCorrectionMode
 - Spinnaker::Camera, [500](#)
- DefectCorrectionModeEnums
 - CameraDefs Class, [94](#)
- DefectTableApply
 - Spinnaker::Camera, [501](#)
- DefectTableCoordinateX
 - Spinnaker::Camera, [501](#)
- DefectTableCoordinateY
 - Spinnaker::Camera, [501](#)
- DefectTableFactoryRestore
 - Spinnaker::Camera, [501](#)
- DefectTableIndex
 - Spinnaker::Camera, [502](#)
- DefectTablePixelCount
 - Spinnaker::Camera, [502](#)
- DefectTableSave
 - Spinnaker::Camera, [502](#)
- Deinterlacing
 - Spinnaker::Camera, [502](#)
- DeinterlacingEnums
 - CameraDefs Class, [95](#)
- deleteFile
 - Spinnaker::GenApi::FileProtocolAdapter, [768](#)
- DeliverEventMessage
 - Spinnaker::GenApi::CEventAdapter1394, [642](#)
 - Spinnaker::GenApi::CEventAdapterGEV, [646](#)
 - Spinnaker::GenApi::CEventAdapterU3V, [648](#)
- DeliverMessage
 - Spinnaker::GenApi::CEventAdapter, [639](#)
 - Spinnaker::GenApi::CEventAdapter1394, [642](#)
 - Spinnaker::GenApi::CEventAdapterGEV, [646](#)
 - Spinnaker::GenApi::CEventAdapterGeneric, [644](#)
 - Spinnaker::GenApi::CEventAdapterU3V, [648](#)
- Deregister
 - NodeCallback Class, [324](#)
- DeregisterCallback
 - INode Interface, [289](#)
 - Spinnaker::GenApi::Node, [974](#)
- Destroy
 - Spinnaker::GenApi::CNodeCallback, [684](#)
 - Spinnaker::GenApi::Function_NodeCallback, [781](#)
 - Spinnaker::GenApi::Member_NodeCallback, [969](#)
 - Spinnaker::GenApi::NodeMap, [985](#)
- DetachBuffer
 - Spinnaker::GenApi::CChunkAdapter, [618](#)
- DetachChunk
 - Spinnaker::GenApi::CChunkPort, [631](#)
- DetachEvent
 - Spinnaker::GenApi::CEventPort, [652](#)
- DetachNode
 - Spinnaker::GenApi::CEventPort, [652](#)
- DetachNodeMap
 - Spinnaker::GenApi::CChunkAdapter, [618](#)
 - Spinnaker::GenApi::CEventAdapter, [640](#)
- DetachPort
 - Spinnaker::GenApi::CChunkPort, [631](#)
- DeviceAccessStatus
 - Spinnaker::TransportLayerDevice, [1047](#)
 - Spinnaker::TransportLayerInterface, [1057](#)
- DeviceAccessStatusEnum
 - TransportLayerDefs Class, [197](#)
- DeviceAddress
 - Spinnaker::ActionCommandResult, [425](#)

- DeviceCharacterSet
 - Spinnaker::Camera, [502](#)
- DeviceCharacterSetEnums
 - CameraDefs Class, [95](#)
- DeviceClockFrequency
 - Spinnaker::Camera, [503](#)
- DeviceClockSelector
 - Spinnaker::Camera, [503](#)
- DeviceClockSelectorEnums
 - CameraDefs Class, [95](#)
- DeviceConnectionSelector
 - Spinnaker::Camera, [503](#)
- DeviceConnectionSpeed
 - Spinnaker::Camera, [503](#)
- DeviceConnectionStatus
 - Spinnaker::Camera, [503](#)
- DeviceConnectionStatusEnums
 - CameraDefs Class, [96](#)
- DeviceCount
 - Spinnaker::TransportLayerInterface, [1057](#)
- DeviceCurrentSpeed
 - Spinnaker::TransportLayerDevice, [1047](#)
- DeviceCurrentSpeedEnum
 - TransportLayerDefs Class, [198](#)
- DeviceDisplayName
 - Spinnaker::TransportLayerDevice, [1047](#)
- DeviceDriverVersion
 - Spinnaker::TransportLayerDevice, [1047](#)
- DeviceEndianessMechanism
 - Spinnaker::TransportLayerDevice, [1047](#)
- DeviceEndianessMechanismEnum
 - TransportLayerDefs Class, [198](#)
- DeviceEvent, [730](#)
 - Spinnaker::DeviceEvent, [731](#)
- DeviceEvent Class, [158](#)
- DeviceEventChannelCount
 - Spinnaker::Camera, [503](#)
- DeviceFamilyName
 - Spinnaker::Camera, [504](#)
- DeviceFeaturePersistenceEnd
 - Spinnaker::Camera, [504](#)
- DeviceFeaturePersistenceStart
 - Spinnaker::Camera, [504](#)
- DeviceFirmwareVersion
 - Spinnaker::Camera, [504](#)
- DeviceGenCPVersionMajor
 - Spinnaker::Camera, [504](#)
- DeviceGenCPVersionMinor
 - Spinnaker::Camera, [504](#)
- DeviceID
 - Spinnaker::Camera, [505](#)
 - Spinnaker::TransportLayerDevice, [1048](#)
 - Spinnaker::TransportLayerInterface, [1057](#)
- DeviceIndicatorMode
 - Spinnaker::Camera, [505](#)
- DeviceIndicatorModeEnums
 - CameraDefs Class, [96](#)
- DeviceInstanceId
 - Spinnaker::TransportLayerDevice, [1048](#)
- DevicesUpdater
 - Spinnaker::TransportLayerDevice, [1048](#)
- DeviceLinkBandwidthReserve
 - Spinnaker::Camera, [505](#)
- DeviceLinkCommandTimeout
 - Spinnaker::Camera, [505](#)
- DeviceLinkConnectionCount
 - Spinnaker::Camera, [505](#)
- DeviceLinkCurrentThroughput
 - Spinnaker::Camera, [505](#)
- DeviceLinkHeartbeatMode
 - Spinnaker::Camera, [506](#)
- DeviceLinkHeartbeatModeEnums
 - CameraDefs Class, [96](#)
- DeviceLinkHeartbeatTimeout
 - Spinnaker::Camera, [506](#)
- DeviceLinkSelector
 - Spinnaker::Camera, [506](#)
- DeviceLinkSpeed
 - Spinnaker::Camera, [506](#)
 - Spinnaker::TransportLayerDevice, [1048](#)
- DeviceLinkThroughputLimit
 - Spinnaker::Camera, [506](#)
- DeviceLinkThroughputLimitMode
 - Spinnaker::Camera, [507](#)
- DeviceLinkThroughputLimitModeEnums
 - CameraDefs Class, [98](#)
- DeviceLocation
 - Spinnaker::TransportLayerDevice, [1048](#)
- DeviceManifestEntrySelector
 - Spinnaker::Camera, [507](#)
- DeviceManifestPrimaryURL
 - Spinnaker::Camera, [507](#)
- DeviceManifestSchemaMajorVersion
 - Spinnaker::Camera, [507](#)
- DeviceManifestSchemaMinorVersion
 - Spinnaker::Camera, [507](#)
- DeviceManifestSecondaryURL
 - Spinnaker::Camera, [508](#)
- DeviceManifestXMLMajorVersion
 - Spinnaker::Camera, [508](#)
- DeviceManifestXMLMinorVersion
 - Spinnaker::Camera, [508](#)
- DeviceManifestXMLSubMinorVersion
 - Spinnaker::Camera, [508](#)
- DeviceManufacturerInfo
 - Spinnaker::Camera, [508](#)
- DeviceMaxThroughput
 - Spinnaker::Camera, [508](#)
- DeviceModelName
 - Spinnaker::Camera, [509](#)
 - Spinnaker::TransportLayerDevice, [1048](#)
 - Spinnaker::TransportLayerInterface, [1057](#)
- DeviceMulticastMonitorMode
 - Spinnaker::TransportLayerDevice, [1049](#)
- DevicePowerSupplySelector
 - Spinnaker::Camera, [509](#)

- DevicePowerSupplySelectorEnums
 - CameraDefs Class, [98](#)
- DeviceRegistersCheck
 - Spinnaker::Camera, [509](#)
- DeviceRegistersEndianness
 - Spinnaker::Camera, [509](#)
- DeviceRegistersEndiannessEnums
 - CameraDefs Class, [98](#)
- DeviceRegistersStreamingEnd
 - Spinnaker::Camera, [509](#)
- DeviceRegistersStreamingStart
 - Spinnaker::Camera, [510](#)
- DeviceRegistersValid
 - Spinnaker::Camera, [510](#)
- DeviceReset
 - Spinnaker::Camera, [510](#)
- DeviceSFNCVersionMajor
 - Spinnaker::Camera, [511](#)
- DeviceSFNCVersionMinor
 - Spinnaker::Camera, [511](#)
- DeviceSFNCVersionSubMinor
 - Spinnaker::Camera, [511](#)
- DeviceScanType
 - Spinnaker::Camera, [510](#)
- DeviceScanTypeEnum
 - CameraDefs Class, [99](#)
- DeviceSelector
 - Spinnaker::TransportLayerInterface, [1057](#)
- DeviceSerialNumber
 - Spinnaker::Camera, [510](#)
 - Spinnaker::TransportLayerDevice, [1049](#)
- DeviceSerialPortBaudRate
 - Spinnaker::Camera, [510](#)
- DeviceSerialPortBaudRateEnums
 - CameraDefs Class, [99](#)
- DeviceSerialPortSelector
 - Spinnaker::Camera, [511](#)
- DeviceSerialPortSelectorEnums
 - CameraDefs Class, [99](#)
- DeviceStreamChannelCount
 - Spinnaker::Camera, [511](#)
- DeviceStreamChannelEndianness
 - Spinnaker::Camera, [511](#)
- DeviceStreamChannelEndiannessEnums
 - CameraDefs Class, [99](#)
- DeviceStreamChannelLink
 - Spinnaker::Camera, [512](#)
- DeviceStreamChannelPacketSize
 - Spinnaker::Camera, [512](#)
- DeviceStreamChannelSelector
 - Spinnaker::Camera, [512](#)
- DeviceStreamChannelType
 - Spinnaker::Camera, [512](#)
- DeviceStreamChannelTypeEnum
 - CameraDefs Class, [100](#)
- DeviceTLType
 - Spinnaker::Camera, [513](#)
- DeviceTLTypeEnum
 - CameraDefs Class, [102](#)
- DeviceTLVersionMajor
 - Spinnaker::Camera, [513](#)
- DeviceTLVersionMinor
 - Spinnaker::Camera, [513](#)
- DeviceTLVersionSubMinor
 - Spinnaker::Camera, [513](#)
- DeviceTapGeometry
 - Spinnaker::Camera, [512](#)
- DeviceTapGeometryEnums
 - CameraDefs Class, [100](#)
- DeviceTemperature
 - Spinnaker::Camera, [512](#)
- DeviceTemperatureSelector
 - Spinnaker::Camera, [513](#)
- DeviceTemperatureSelectorEnums
 - CameraDefs Class, [101](#)
- DeviceType
 - Spinnaker::Camera, [514](#)
 - Spinnaker::TransportLayerDevice, [1049](#)
- DeviceTypeEnum
 - TransportLayerDefs Class, [198](#)
- DeviceTypeEnum
 - CameraDefs Class, [102](#)
- DeviceU3VProtocol
 - Spinnaker::TransportLayerDevice, [1049](#)
- DeviceUnlock
 - Spinnaker::TransportLayerInterface, [1058](#)
- DeviceUpdateList
 - Spinnaker::TransportLayerInterface, [1058](#)
- DeviceUptime
 - Spinnaker::Camera, [514](#)
- DeviceUserID
 - Spinnaker::Camera, [514](#)
 - Spinnaker::TransportLayerDevice, [1049](#)
- DeviceVendorName
 - Spinnaker::Camera, [514](#)
 - Spinnaker::TransportLayerDevice, [1049](#)
 - Spinnaker::TransportLayerInterface, [1058](#)
- DeviceVersion
 - Spinnaker::Camera, [514](#)
 - Spinnaker::TransportLayerDevice, [1050](#)
- DisableAll
 - Spinnaker::ImageStatistics, [851](#)
 - Spinnaker::ImageStatistics, [900](#)
- DiscoverMaxPacketSize
 - Spinnaker::CameraBase, [594](#)
 - Spinnaker::ICameraBase, [807](#)
- doc/Doxygen/spindocs/Licensing.dox, [1081](#)
- doc/Doxygen/spindocs/MainPage.dox, [1081](#)
- DoesEnvironmentVariableExist
 - GCUtilities Utility, [258](#)
- double _autovector_t, [733](#)
 - Spinnaker::GenApi::double _autovector_t, [734](#)
- EAccessMode
 - Types Enums, [355](#)
- EAccessModeClass, [736](#)
- ECacheUsage_t

- NodeMapFactory Class, [327](#)
- ECachingMode
 - Types Enums, [355](#)
- ECachingModeClass, [737](#)
- ECallbackType
 - NodeCallback Class, [324](#)
- EContentType_t
 - NodeMapFactory Class, [328](#)
- EDisplayNotation
 - Types Enums, [355](#)
- EDisplayNotationClass, [738](#)
- EEndianess
 - Types Enums, [357](#)
- EEndianessClass, [739](#)
- EGenApiSchemaVersion
 - Types Enums, [357](#)
- EGenApiSchemaVersionClass, [740](#)
- ElncMode
 - Types Enums, [357](#)
- ElncDirection
 - Types Enums, [358](#)
- ElncDirectionClass, [741](#)
- ElncInterfaceType
 - Types Enums, [358](#)
- ElncLinkType
 - Types Enums, [358](#)
- ElncNamespace
 - Types Enums, [359](#)
- ElncNamespaceClass, [742](#)
- ElncRepresentation
 - Types Enums, [359](#)
- ElncRepresentationClass, [751](#)
- ESign
 - Types Enums, [359](#)
- ESignClass, [752](#)
- ESlope
 - Types Enums, [360](#)
- ESlopeClass, [753](#)
- EStandardNameSpace
 - Types Enums, [360](#)
- EStandardNameSpaceClass, [754](#)
- EVENT_TIMEOUT_INFINITE
 - Spinnaker Headers, [177](#)
- EVENT_TIMEOUT_NONE
 - Spinnaker Headers, [177](#)
- EVisibility
 - Types Enums, [360](#)
- EVisibilityClass, [759](#)
- EXMLValidation
 - Types Enums, [361](#)
- EXPAND_TO_STRINGISE
 - GCUtilities.h, [1242](#)
- EYesNo
 - Types Enums, [361](#)
- EYesNoClass, [765](#)
- EatComments
 - Spinnaker GenApi Classes, [223](#)
- empty
 - Spinnaker::GenICam::gcstring, [785](#)
- EnableAll
 - Spinnaker::ImageStatistics, [851](#)
 - Spinnaker::ImageStatistics, [900](#)
- EnableGreyOnly
 - Spinnaker::ImageStatistics, [851](#)
 - Spinnaker::ImageStatistics, [900](#)
- EnableHSLOnly
 - Spinnaker::ImageStatistics, [851](#)
 - Spinnaker::ImageStatistics, [901](#)
- EnableRGBOnly
 - Spinnaker::ImageStatistics, [852](#)
 - Spinnaker::ImageStatistics, [901](#)
- EncoderDivider
 - Spinnaker::Camera, [514](#)
- EncoderMode
 - Spinnaker::Camera, [515](#)
- EncoderModeEnums
 - CameraDefs Class, [102](#)
- EncoderOutputMode
 - Spinnaker::Camera, [515](#)
- EncoderOutputModeEnums
 - CameraDefs Class, [103](#)
- EncoderReset
 - Spinnaker::Camera, [515](#)
- EncoderResetActivation
 - Spinnaker::Camera, [515](#)
- EncoderResetActivationEnums
 - CameraDefs Class, [103](#)
- EncoderResetSource
 - Spinnaker::Camera, [515](#)
- EncoderResetSourceEnums
 - CameraDefs Class, [104](#)
- EncoderSelector
 - Spinnaker::Camera, [515](#)
- EncoderSelectorEnums
 - CameraDefs Class, [105](#)
- EncoderSourceAEnums
 - CameraDefs Class, [105](#)
- EncoderSourceBEnums
 - CameraDefs Class, [105](#)
- EncoderSourceA
 - Spinnaker::Camera, [516](#)
- EncoderSourceB
 - Spinnaker::Camera, [516](#)
- EncoderStatus
 - Spinnaker::Camera, [516](#)
- EncoderStatusEnums
 - CameraDefs Class, [106](#)
- EncoderTimeout
 - Spinnaker::Camera, [516](#)
- EncoderValue
 - Spinnaker::Camera, [516](#)
- EncoderValueAtReset
 - Spinnaker::Camera, [516](#)
- EndAcquisition
 - Spinnaker::CameraBase, [595](#)
 - Spinnaker::ICameraBase, [808](#)

- EnumClasses Class, [239](#)
- EnumEntryNode, [743](#)
 - Spinnaker::GenApi::EnumEntryNode, [744](#)
- EnumEntryNode Class, [241](#)
 - CEnumEntryRef, [241](#)
- EnumNode, [746](#)
 - Spinnaker::GenApi::EnumNode, [748](#)
- EnumNode Class, [242](#)
 - CEnumerationRef, [242](#)
- EnumNodeT Class, [243](#)
- EnumerateGEVInterfaces
 - Spinnaker::TransportLayerSystem, [1071](#)
- EnumerationCount
 - Spinnaker::Camera, [517](#)
- Error
 - Spinnaker Definitions, [184](#)
- Event, [756](#)
 - GVCP_EVENTDATA_REQUEST_EXTENDED_↔ ID, [799](#)
 - GVCP_EVENTDATA_REQUEST, [798](#)
 - Spinnaker::Event, [757](#)
- Event Class, [159](#)
- EventAcquisitionEnd
 - Spinnaker::Camera, [517](#)
- EventAcquisitionEndFrameID
 - Spinnaker::Camera, [517](#)
- EventAcquisitionEndTimestamp
 - Spinnaker::Camera, [517](#)
- EventAcquisitionError
 - Spinnaker::Camera, [517](#)
- EventAcquisitionErrorFrameID
 - Spinnaker::Camera, [517](#)
- EventAcquisitionErrorTimestamp
 - Spinnaker::Camera, [518](#)
- EventAcquisitionStart
 - Spinnaker::Camera, [518](#)
- EventAcquisitionStartFrameID
 - Spinnaker::Camera, [518](#)
- EventAcquisitionStartTimestamp
 - Spinnaker::Camera, [518](#)
- EventAcquisitionTransferEnd
 - Spinnaker::Camera, [518](#)
- EventAcquisitionTransferEndFrameID
 - Spinnaker::Camera, [518](#)
- EventAcquisitionTransferEndTimestamp
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTransferStart
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTransferStartFrameID
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTransferStartTimestamp
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTrigger
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTriggerFrameID
 - Spinnaker::Camera, [519](#)
- EventAcquisitionTriggerTimestamp
 - Spinnaker::Camera, [520](#)
- EventActionLate
 - Spinnaker::Camera, [520](#)
- EventActionLateFrameID
 - Spinnaker::Camera, [520](#)
- EventActionLateTimestamp
 - Spinnaker::Camera, [520](#)
- EventAdapter Class, [244](#)
- EventAdapter1394 Class, [245](#)
- EventAdapterGEV Class, [247](#)
- EventAdapterGeneric Class, [246](#)
- EventAdapterU3V Class, [248](#)
- EventCounter0End
 - Spinnaker::Camera, [520](#)
- EventCounter0EndFrameID
 - Spinnaker::Camera, [520](#)
- EventCounter0EndTimestamp
 - Spinnaker::Camera, [521](#)
- EventCounter0Start
 - Spinnaker::Camera, [521](#)
- EventCounter0StartFrameID
 - Spinnaker::Camera, [521](#)
- EventCounter0StartTimestamp
 - Spinnaker::Camera, [521](#)
- EventCounter1End
 - Spinnaker::Camera, [521](#)
- EventCounter1EndFrameID
 - Spinnaker::Camera, [521](#)
- EventCounter1EndTimestamp
 - Spinnaker::Camera, [522](#)
- EventCounter1Start
 - Spinnaker::Camera, [522](#)
- EventCounter1StartFrameID
 - Spinnaker::Camera, [522](#)
- EventCounter1StartTimestamp
 - Spinnaker::Camera, [522](#)
- EventData
 - U3V_EVENT_MESSAGE, [1075](#)
- EventEncoder0Restarted
 - Spinnaker::Camera, [522](#)
- EventEncoder0RestartedFrameID
 - Spinnaker::Camera, [522](#)
- EventEncoder0RestartedTimestamp
 - Spinnaker::Camera, [523](#)
- EventEncoder0Stopped
 - Spinnaker::Camera, [523](#)
- EventEncoder0StoppedFrameID
 - Spinnaker::Camera, [523](#)
- EventEncoder0StoppedTimestamp
 - Spinnaker::Camera, [523](#)
- EventEncoder1Restarted
 - Spinnaker::Camera, [523](#)
- EventEncoder1RestartedFrameID
 - Spinnaker::Camera, [523](#)
- EventEncoder1RestartedTimestamp
 - Spinnaker::Camera, [524](#)
- EventEncoder1Stopped
 - Spinnaker::Camera, [524](#)
- EventEncoder1StoppedFrameID

- Spinnaker::Camera, [524](#)
- EventEncoder1StoppedTimestamp
 - Spinnaker::Camera, [524](#)
- EventError
 - Spinnaker::Camera, [524](#)
- EventErrorCode
 - Spinnaker::Camera, [524](#)
- EventErrorFrameID
 - Spinnaker::Camera, [525](#)
- EventErrorTimestamp
 - Spinnaker::Camera, [525](#)
- EventExposureEnd
 - Spinnaker::Camera, [525](#)
- EventExposureEndFrameID
 - Spinnaker::Camera, [525](#)
- EventExposureEndTimestamp
 - Spinnaker::Camera, [525](#)
- EventExposureStart
 - Spinnaker::Camera, [525](#)
- EventExposureStartFrameID
 - Spinnaker::Camera, [526](#)
- EventExposureStartTimestamp
 - Spinnaker::Camera, [526](#)
- EventFrameBurstEnd
 - Spinnaker::Camera, [526](#)
- EventFrameBurstEndFrameID
 - Spinnaker::Camera, [526](#)
- EventFrameBurstEndTimestamp
 - Spinnaker::Camera, [526](#)
- EventFrameBurstStart
 - Spinnaker::Camera, [526](#)
- EventFrameBurstStartFrameID
 - Spinnaker::Camera, [527](#)
- EventFrameBurstStartTimestamp
 - Spinnaker::Camera, [527](#)
- EventFrameEnd
 - Spinnaker::Camera, [527](#)
- EventFrameEndFrameID
 - Spinnaker::Camera, [527](#)
- EventFrameEndTimestamp
 - Spinnaker::Camera, [527](#)
- EventFrameStart
 - Spinnaker::Camera, [527](#)
- EventFrameStartFrameID
 - Spinnaker::Camera, [528](#)
- EventFrameStartTimestamp
 - Spinnaker::Camera, [528](#)
- EventFrameTransferEnd
 - Spinnaker::Camera, [528](#)
- EventFrameTransferEndFrameID
 - Spinnaker::Camera, [528](#)
- EventFrameTransferEndTimestamp
 - Spinnaker::Camera, [528](#)
- EventFrameTransferStart
 - Spinnaker::Camera, [528](#)
- EventFrameTransferStartFrameID
 - Spinnaker::Camera, [529](#)
- EventFrameTransferStartTimestamp
 - Spinnaker::Camera, [529](#)
- EventFrameTrigger
 - Spinnaker::Camera, [529](#)
- EventFrameTriggerFrameID
 - Spinnaker::Camera, [529](#)
- EventFrameTriggerTimestamp
 - Spinnaker::Camera, [529](#)
- EventId
 - GVCP_EVENT_ITEM_BASIC, [793](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
 - GVCP_EVENT_ITEM, [792](#)
 - U3V_EVENT_DATA, [1074](#)
- EventLine0AnyEdge
 - Spinnaker::Camera, [529](#)
- EventLine0AnyEdgeFrameID
 - Spinnaker::Camera, [530](#)
- EventLine0AnyEdgeTimestamp
 - Spinnaker::Camera, [530](#)
- EventLine0FallingEdge
 - Spinnaker::Camera, [530](#)
- EventLine0FallingEdgeFrameID
 - Spinnaker::Camera, [530](#)
- EventLine0FallingEdgeTimestamp
 - Spinnaker::Camera, [530](#)
- EventLine0RisingEdge
 - Spinnaker::Camera, [530](#)
- EventLine0RisingEdgeFrameID
 - Spinnaker::Camera, [531](#)
- EventLine0RisingEdgeTimestamp
 - Spinnaker::Camera, [531](#)
- EventLine1AnyEdge
 - Spinnaker::Camera, [531](#)
- EventLine1AnyEdgeFrameID
 - Spinnaker::Camera, [531](#)
- EventLine1AnyEdgeTimestamp
 - Spinnaker::Camera, [531](#)
- EventLine1FallingEdge
 - Spinnaker::Camera, [531](#)
- EventLine1FallingEdgeFrameID
 - Spinnaker::Camera, [532](#)
- EventLine1FallingEdgeTimestamp
 - Spinnaker::Camera, [532](#)
- EventLine1RisingEdge
 - Spinnaker::Camera, [532](#)
- EventLine1RisingEdgeFrameID
 - Spinnaker::Camera, [532](#)
- EventLine1RisingEdgeTimestamp
 - Spinnaker::Camera, [532](#)
- EventLinkSpeedChange
 - Spinnaker::Camera, [532](#)
- EventLinkSpeedChangeFrameID
 - Spinnaker::Camera, [533](#)
- EventLinkSpeedChangeTimestamp
 - Spinnaker::Camera, [533](#)
- EventLinkTrigger0
 - Spinnaker::Camera, [533](#)
- EventLinkTrigger0FrameID
 - Spinnaker::Camera, [533](#)

- EventLinkTrigger0Timestamp
 - Spinnaker::Camera, [533](#)
- EventLinkTrigger1
 - Spinnaker::Camera, [533](#)
- EventLinkTrigger1FrameID
 - Spinnaker::Camera, [534](#)
- EventLinkTrigger1Timestamp
 - Spinnaker::Camera, [534](#)
- EventNotification
 - Spinnaker::Camera, [534](#)
- EventNotificationEnums
 - CameraDefs Class, [106](#)
- EventPort Class, [249](#)
- EventProcessor
 - Spinnaker::Event, [758](#)
- EventSelector
 - Spinnaker::Camera, [534](#)
- EventSelectorEnums
 - CameraDefs Class, [106](#)
- EventSequencerSetChange
 - Spinnaker::Camera, [534](#)
- EventSequencerSetChangeFrameID
 - Spinnaker::Camera, [534](#)
- EventSequencerSetChangeTimestamp
 - Spinnaker::Camera, [535](#)
- EventSerialData
 - Spinnaker::Camera, [535](#)
- EventSerialDataLength
 - Spinnaker::Camera, [535](#)
- EventSerialPortReceive
 - Spinnaker::Camera, [535](#)
- EventSerialPortReceiveTimestamp
 - Spinnaker::Camera, [535](#)
- EventSerialReceiveOverflow
 - Spinnaker::Camera, [535](#)
- EventStream0TransferBlockEnd
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockEndFrameID
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockEndTimestamp
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockStart
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockStartFrameID
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockStartTimestamp
 - Spinnaker::Camera, [536](#)
- EventStream0TransferBlockTrigger
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBlockTriggerFrameID
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBlockTriggerTimestamp
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBurstEnd
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBurstEndFrameID
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBurstEndTimestamp
 - Spinnaker::Camera, [537](#)
- EventStream0TransferBurstStart
 - Spinnaker::Camera, [538](#)
- EventStream0TransferBurstStartFrameID
 - Spinnaker::Camera, [538](#)
- EventStream0TransferBurstStartTimestamp
 - Spinnaker::Camera, [538](#)
- EventStream0TransferEnd
 - Spinnaker::Camera, [538](#)
- EventStream0TransferEndFrameID
 - Spinnaker::Camera, [538](#)
- EventStream0TransferEndTimestamp
 - Spinnaker::Camera, [538](#)
- EventStream0TransferOverflow
 - Spinnaker::Camera, [539](#)
- EventStream0TransferOverflowFrameID
 - Spinnaker::Camera, [539](#)
- EventStream0TransferOverflowTimestamp
 - Spinnaker::Camera, [539](#)
- EventStream0TransferPause
 - Spinnaker::Camera, [539](#)
- EventStream0TransferPauseFrameID
 - Spinnaker::Camera, [539](#)
- EventStream0TransferPauseTimestamp
 - Spinnaker::Camera, [539](#)
- EventStream0TransferResume
 - Spinnaker::Camera, [540](#)
- EventStream0TransferResumeFrameID
 - Spinnaker::Camera, [540](#)
- EventStream0TransferResumeTimestamp
 - Spinnaker::Camera, [540](#)
- EventStream0TransferStart
 - Spinnaker::Camera, [540](#)
- EventStream0TransferStartFrameID
 - Spinnaker::Camera, [540](#)
- EventStream0TransferStartTimestamp
 - Spinnaker::Camera, [540](#)
- EventTest
 - Spinnaker::Camera, [541](#)
- EventTestTimestamp
 - Spinnaker::Camera, [541](#)
- EventTimer0End
 - Spinnaker::Camera, [541](#)
- EventTimer0EndFrameID
 - Spinnaker::Camera, [541](#)
- EventTimer0EndTimestamp
 - Spinnaker::Camera, [541](#)
- EventTimer0Start
 - Spinnaker::Camera, [541](#)
- EventTimer0StartFrameID
 - Spinnaker::Camera, [542](#)
- EventTimer0StartTimestamp
 - Spinnaker::Camera, [542](#)
- EventTimer1End
 - Spinnaker::Camera, [542](#)
- EventTimer1EndFrameID
 - Spinnaker::Camera, [542](#)
- EventTimer1EndTimestamp

- Spinnaker::Camera, [542](#)
- EventTimer1 Start
 - Spinnaker::Camera, [542](#)
- EventTimer1 StartFrameID
 - Spinnaker::Camera, [543](#)
- EventTimer1 StartTimestamp
 - Spinnaker::Camera, [543](#)
- EventType
 - Spinnaker Definitions, [185](#)
- Exception, [760](#)
 - Spinnaker::Exception, [762](#)
- Exception Class, [160](#)
- Execute
 - Spinnaker::GenApi::CommandNode, [704](#)
- Expert
 - Types Enums, [361](#)
- ExposureActiveMode
 - Spinnaker::Camera, [543](#)
- ExposureActiveModeEnums
 - CameraDefs Class, [107](#)
- ExposureAuto
 - Spinnaker::Camera, [543](#)
- ExposureAutoEnums
 - CameraDefs Class, [107](#)
- ExposureMode
 - Spinnaker::Camera, [543](#)
- ExposureModeEnums
 - CameraDefs Class, [107](#)
- ExposureTime
 - Spinnaker::Camera, [543](#)
- ExposureTimeMode
 - Spinnaker::Camera, [544](#)
- ExposureTimeModeEnums
 - CameraDefs Class, [108](#)
- ExposureTimeSelector
 - Spinnaker::Camera, [544](#)
- ExposureTimeSelectorEnums
 - CameraDefs Class, [108](#)
- ExtractIndependentSubtree
 - INodeMapDyn Interface, [302](#)
- ExtractPolarQuadrant
 - Spinnaker::ImageUtilityPolarization, [920](#)
- ExtractSubtree
 - Spinnaker::GenApi::CNodeMapFactory, [691](#)
- FLIR_SPINNAKER_VERSION_BUILD
 - System.h, [1334](#)
- FLIR_SPINNAKER_VERSION_MAJOR
 - System.h, [1335](#)
- FLIR_SPINNAKER_VERSION_MINOR
 - System.h, [1335](#)
- FLIR_SPINNAKER_VERSION_TYPE
 - System.h, [1335](#)
- FMT_I64
 - Compatibility.h, [1203](#)
- FactoryReset
 - Spinnaker::Camera, [544](#)
- FileAccessBuffer
 - Spinnaker::Camera, [544](#)
- FileAccessLength
 - Spinnaker::Camera, [544](#)
- FileAccessOffset
 - Spinnaker::Camera, [544](#)
- FileOpenMode
 - Spinnaker::Camera, [545](#)
- FileOpenModeEnums
 - CameraDefs Class, [109](#)
- FileOperationExecute
 - Spinnaker::Camera, [545](#)
- FileOperationResult
 - Spinnaker::Camera, [545](#)
- FileOperationSelector
 - Spinnaker::Camera, [545](#)
- FileOperationSelectorEnums
 - CameraDefs Class, [109](#)
- FileOperationStatus
 - Spinnaker::Camera, [545](#)
- FileOperationStatusEnums
 - CameraDefs Class, [109](#)
- FileProtocolAdapter, [766](#)
 - Spinnaker::GenApi::FileProtocolAdapter, [766](#)
- FileSelector
 - Spinnaker::Camera, [546](#)
- FileSelectorEnums
 - CameraDefs Class, [110](#)
- FileSize
 - Spinnaker::Camera, [546](#)
- filebuf_type
 - Spinnaker::GenApi::IDevFileStreamBase, [831](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [993](#)
- Filestream Class, [250](#)
- FillCRCInfo
 - Spinnaker::IDataStream, [827](#)
- FilterDriverStatus
 - Spinnaker::TransportLayerInterface, [1058](#)
- FilterDriverStatusEnum
 - TransportLayerDefs Class, [199](#)
- find
 - Spinnaker::GenICam::gcstring, [785](#), [786](#)
- find_first_not_of
 - Spinnaker::GenICam::gcstring, [786](#)
- find_first_of
 - Spinnaker::GenICam::gcstring, [786](#)
- Flags
 - GVCP_REQUEST_HEADER, [800](#)
 - U3V_COMMAND_HEADER, [1073](#)
- float32_t
 - GCTypes Class, [255](#)
- float64_t
 - GCTypes Class, [255](#)
- FloatNode, [770](#)
 - Spinnaker::GenApi::FloatNode, [773](#)
- FloatNode Class, [251](#)
 - CFloatRef, [251](#)
- FloatRegNode, [777](#)
 - Spinnaker::GenApi::FloatRegNode, [778](#), [779](#)
- FloatRegNode Class, [252](#)

- FlushQueueAllDiscard
 - Spinnaker::IDataStream, [827](#)
- ForceIP
 - Spinnaker::CameraBase, [595](#)
 - Spinnaker::ICameraBase, [808](#)
- frameRate
 - Spinnaker::Video::AVIOption, [431](#)
 - Spinnaker::Video::H264Option, [801](#)
 - Spinnaker::Video::MJPGOption, [970](#)
- FromString
 - IValue Class, [320](#)
 - Spinnaker::GenApi::EAccessModeClass, [736](#)
 - Spinnaker::GenApi::ECachingModeClass, [737](#)
 - Spinnaker::GenApi::EDisplayNotationClass, [738](#)
 - Spinnaker::GenApi::EEndianessClass, [739](#)
 - Spinnaker::GenApi::EGenApiSchemaVersion↔
Class, [740](#)
 - Spinnaker::GenApi::EInputDirectionClass, [741](#)
 - Spinnaker::GenApi::ENameSpaceClass, [742](#)
 - Spinnaker::GenApi::ERepresentationClass, [751](#)
 - Spinnaker::GenApi::ESignClass, [753](#)
 - Spinnaker::GenApi::ESlopeClass, [754](#)
 - Spinnaker::GenApi::EStandardNameSpaceClass,
[755](#)
 - Spinnaker::GenApi::EVisibilityClass, [759](#)
 - Spinnaker::GenApi::EYesNoClass, [765](#)
 - Spinnaker::GenApi::ValueNode, [1078](#)
- Function_NodeCallback
 - Spinnaker::GenApi::Function_NodeCallback, [781](#)
- Function_NodeCallback< Function >, [780](#)
- GC_COUNTOF
 - GCUtilities.h, [1242](#)
- GC_INT32_MAX
 - GCTypes.h, [1236](#)
- GC_INT32_MIN
 - GCTypes.h, [1236](#)
- GC_INT64_MAX
 - GCTypes.h, [1236](#)
- GC_INT64_MIN
 - GCTypes.h, [1236](#)
- GC_INT8_MAX
 - GCTypes.h, [1237](#)
- GC_INT8_MIN
 - GCTypes.h, [1237](#)
- GC_UINT32_MAX
 - GCTypes.h, [1237](#)
- GC_UINT64_MAX
 - GCTypes.h, [1237](#)
- GC_UINT8_MAX
 - GCTypes.h, [1237](#)
- GCSTRING_NPOS
 - GCString.h, [1232](#)
- GCString Class, [253](#)
- GCString.h
 - GCSTRING_NPOS, [1232](#)
 - operator<<, [1232](#)
 - operator>>, [1233](#)
- GCSynch Class, [254](#)
- GCTypes Class, [255](#)
 - float32_t, [255](#)
 - float64_t, [255](#)
- GCTypes.h
 - __STDC_CONSTANT_MACROS, [1236](#)
 - __STDC_LIMIT_MACROS, [1236](#)
 - GC_INT32_MAX, [1236](#)
 - GC_INT32_MIN, [1236](#)
 - GC_INT64_MAX, [1236](#)
 - GC_INT64_MIN, [1236](#)
 - GC_INT8_MAX, [1237](#)
 - GC_INT8_MIN, [1237](#)
 - GC_UINT32_MAX, [1237](#)
 - GC_UINT64_MAX, [1237](#)
 - GC_UINT8_MAX, [1237](#)
- GCUtilities Utility, [257](#)
 - DoesEnvironmentVariableExist, [258](#)
 - GetFiles, [258](#)
 - GetGenICamCLProtocolFolder, [259](#)
 - GetGenICamCacheFolder, [258](#)
 - GetGenICamLogConfig, [259](#)
 - GetModulePathFromFunction, [259](#)
 - GetValueOfEnvironmentVariable, [259](#), [260](#)
 - INTEGRAL_CAST2, [260](#)
 - INTEGRAL_CAST, [260](#)
 - ReplaceEnvironmentVariables, [260](#)
 - SetGenICamCLProtocolFolder, [261](#)
 - SetGenICamCacheFolder, [260](#)
 - SetGenICamLogConfig, [261](#)
 - Tokenize, [261](#)
 - UrlDecode, [261](#)
 - UrlEncode, [262](#)
- GCUtilities.h
 - _TO_STRING, [1241](#)
 - __ERR__, [1241](#)
 - __LINE_STR__, [1241](#)
 - __LOCATION__, [1241](#)
 - __OUTPUT_FORMATER__, [1241](#)
 - __TODO__, [1241](#)
 - __WARN__, [1241](#)
 - EXPAND_TO_STRINGISE, [1242](#)
 - GC_COUNTOF, [1242](#)
 - GENICAM_DEPRECATED, [1242](#)
 - GENICAM_UNUSED, [1242](#)
 - USE_TEMP_CACHE_FILE, [1242](#)
- GENCP_COMMAND_HEADER_SIZE
 - Spinnaker::GenApi, [420](#)
- GENCP_EVENT_BASIC_SIZE
 - Spinnaker::GenApi, [420](#)
- GENCP_EVENT_CMD_ID
 - Spinnaker::GenApi, [420](#)
- GENICAM_DEPRECATED
 - GCUtilities.h, [1242](#)
- GENICAM_UNUSED
 - GCUtilities.h, [1242](#)
- GUIXMLLocation
 - Spinnaker::TransportLayerDevice, [1053](#)
- GUIXMLLocationEnum

- TransportLayerDefs Class, [200](#)
- GUIXMLPath
 - Spinnaker::TransportLayerDevice, [1053](#)
- GVCP_CHUNK_TRAILER, [791](#)
 - ChunkID, [791](#)
 - ChunkLength, [791](#)
- GVCP_EVENT_ITEM_BASIC, [793](#)
 - EventId, [793](#)
 - ReservedOrEventSize, [793](#)
- GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
 - BlockId, [794](#)
 - BlockId64High, [794](#)
 - BlockId64Low, [794](#)
 - EventId, [794](#)
 - ReservedOrEventSize, [794](#)
 - StreamChannelId, [795](#)
 - TimestampHigh, [795](#)
 - TimestampLow, [795](#)
- GVCP_EVENT_ITEM, [792](#)
 - BlockId, [792](#)
 - EventId, [792](#)
 - ReservedOrEventSize, [792](#)
 - StreamChannelId, [792](#)
 - TimestampHigh, [792](#)
 - TimestampLow, [793](#)
- GVCP_EVENT_REQUEST_EXTENDED_ID, [796](#)
 - Header, [797](#)
 - Items, [797](#)
- GVCP_EVENT_REQUEST, [795](#)
 - Header, [796](#)
 - Items, [796](#)
- GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [798](#)
 - Data, [799](#)
 - Event, [799](#)
 - Header, [799](#)
- GVCP_EVENTDATA_REQUEST, [797](#)
 - Data, [798](#)
 - Event, [798](#)
 - Header, [798](#)
- GVCP_MESSAGE_TAGS
 - Spinnaker::GenApi, [419](#)
- GVCP_REQUEST_HEADER, [799](#)
 - Command, [800](#)
 - Flags, [800](#)
 - Length, [800](#)
 - Magic, [800](#)
 - ReqId, [800](#)
- Gain
 - Spinnaker::Camera, [546](#)
- GainAuto
 - Spinnaker::Camera, [546](#)
- GainAutoBalance
 - Spinnaker::Camera, [546](#)
- GainAutoBalanceEnums
 - CameraDefs Class, [110](#)
- GainAutoEnums
 - CameraDefs Class, [110](#)
- GainSelector
 - Spinnaker::Camera, [547](#)
- GainSelectorEnums
 - CameraDefs Class, [111](#)
- Gamma
 - Spinnaker::Camera, [547](#)
- GammaEnable
 - Spinnaker::Camera, [547](#)
- gcstring, [782](#)
 - Spinnaker::GenICam::gcstring, [783](#)
- GenICamXMLLocation
 - Spinnaker::TransportLayerDevice, [1050](#)
- GenICamXMLLocationEnum
 - TransportLayerDefs Class, [199](#)
- GenICamXMLPath
 - Spinnaker::TransportLayerDevice, [1050](#)
- Get
 - IRegister Interfaces, [313](#)
 - Spinnaker::GenApi::RegisterNode, [1013](#)
- get
 - Spinnaker::BasePtr, [433](#)
- GetAccessMode
 - Spinnaker::CameraBase, [595](#)
 - Spinnaker::GenApi::CChunkPort, [631](#)
 - Spinnaker::GenApi::CEventPort, [652](#)
 - Spinnaker::GenApi::CPortImpl, [714](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [720](#)
 - Spinnaker::GenApi::CTestPortStruct, [727](#)
 - Spinnaker::GenApi::Node, [975](#)
 - Spinnaker::GenApi::PortRecorder, [1006](#)
 - Spinnaker::ICameraBase, [808](#)
- GetAddress
 - IRegister Interfaces, [314](#)
 - Spinnaker::GenApi::RegisterNode, [1013](#)
- GetAlias
 - INode Interface, [290](#)
 - Spinnaker::GenApi::Node, [975](#)
- GetBitsPerPixel
 - Spinnaker::Image, [840](#)
 - Spinnaker::Image, [877](#)
- GetBlackLevel
 - Spinnaker::ChunkData, [667](#)
 - Spinnaker::IChunkData, [819](#)
- getBufSize
 - Spinnaker::GenApi::FileProtocolAdapter, [768](#)
- GetBufferOwnership
 - Spinnaker::CameraBase, [596](#)
 - Spinnaker::ICameraBase, [808](#)
- GetBufferSize
 - Spinnaker::Image, [840](#)
 - Spinnaker::Image, [877](#)
- GetBuildDate
 - Spinnaker::Exception, [763](#)
- GetBuildTime
 - Spinnaker::Exception, [763](#)
- GetByIndex
 - Spinnaker::CameraList, [609](#)
 - Spinnaker::ICameraList, [815](#)
 - Spinnaker::IInterfaceList, [862](#)

- Spinnaker::InterfaceList, [939](#)
- GetBySerial
 - Spinnaker::CameraList, [609](#)
 - Spinnaker::ICameraList, [816](#)
- GetCRC
 - Spinnaker::ChunkData, [667](#)
 - Spinnaker::IChunkData, [819](#)
- GetCachingMode
 - INode Interface, [290](#)
 - Spinnaker::GenApi::Node, [975](#)
- GetCallbackType
 - Spinnaker::GenApi::CNodeCallback, [684](#)
- GetCameras
 - Spinnaker::IInterface, [856](#)
 - Spinnaker::ISystem, [949](#)
 - Spinnaker::Interface, [931](#)
 - Spinnaker::System, [1033](#)
- GetCastAlias
 - INode Interface, [290](#)
 - Spinnaker::GenApi::Node, [975](#)
- GetCategoryName
 - Spinnaker::LoggingEventData, [963](#)
- GetChannelStatus
 - Spinnaker::IImageStatistics, [852](#)
 - Spinnaker::ImageStatistics, [901](#)
- GetChildren
 - INode Interface, [290](#)
 - Spinnaker::GenApi::Node, [975](#)
- GetChunkData
 - Spinnaker::IImage, [840](#)
 - Spinnaker::Image, [877](#)
- GetChunkIDLength
 - Spinnaker::GenApi::CChunkPort, [631](#)
- GetChunkID
 - Spinnaker::GenApi::PortNode, [1001](#)
- GetChunkLayoutId
 - Spinnaker::IImage, [840](#)
 - Spinnaker::Image, [878](#)
- GetColorProcessing
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [878](#)
- GetCookie
 - IPortRecorder Interface, [311](#)
 - Spinnaker::GenApi::CPortWriteList, [717](#)
- GetCounterValue
 - Spinnaker::ChunkData, [667](#)
 - Spinnaker::IChunkData, [819](#)
- GetCurrentEntry
 - IEnumeration Interface, [276](#)
 - Spinnaker::GenApi::CEnumerationTRef, [635](#)
 - Spinnaker::GenApi::EnumNode, [748](#)
- GetData
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [878](#)
- GetDefaultColorProcessing
 - Spinnaker::Image, [879](#)
- GetDescription
 - INode Interface, [291](#)
- Spinnaker::GenApi::Node, [976](#)
- GetDeviceEventId
 - Spinnaker::DeviceEvent, [732](#)
 - Spinnaker::IDeviceEvent, [836](#)
- GetDeviceEventName
 - Spinnaker::DeviceEvent, [732](#)
 - Spinnaker::IDeviceEvent, [836](#)
- GetDeviceName
 - INodeMap Interface, [299](#)
 - Spinnaker::GenApi::Node, [976](#)
 - Spinnaker::GenApi::NodeMap, [986](#)
- GetDeviceVersion
 - IDeviceInfo Interface, [271](#)
 - Spinnaker::GenApi::NodeMap, [986](#)
- GetDisplayName
 - INode Interface, [291](#)
 - Spinnaker::GenApi::Node, [976](#)
- GetDisplayNotation
 - IFloat Interface, [282](#)
 - Spinnaker::GenApi::FloatNode, [773](#)
- GetDisplayPrecision
 - IFloat Interface, [282](#)
 - Spinnaker::GenApi::FloatNode, [773](#)
- GetDocuURL
 - INode Interface, [291](#)
 - Spinnaker::GenApi::Node, [976](#)
- GetEncoderValue
 - Spinnaker::ChunkData, [667](#)
 - Spinnaker::IChunkData, [819](#)
- GetEntries
 - IEnumeration Interface, [276](#)
 - Spinnaker::GenApi::EnumNode, [748](#)
- GetEntry
 - IEnumeration Interface, [277](#)
 - IEnumerationT Interface, [279](#)
 - Spinnaker::GenApi::CEnumerationTRef, [636](#)
 - Spinnaker::GenApi::EnumNode, [749](#)
- GetEntryByName
 - IEnumeration Interface, [277](#)
 - Spinnaker::GenApi::EnumNode, [749](#)
- GetEnumAlias
 - Spinnaker::GenApi::CFloatPtr, [658](#)
 - Spinnaker::GenApi::FloatNode, [773](#)
- GetError
 - Spinnaker::Exception, [763](#)
- GetErrorMessage
 - SpinUpdate.h, [1330](#)
 - Spinnaker::Exception, [763](#)
- GetEventIDLength
 - Spinnaker::GenApi::CEventPort, [652](#)
- GetEventID
 - INode Interface, [291](#)
 - Spinnaker::GenApi::Node, [976](#)
- GetEventPayloadData
 - Spinnaker::Event, [757](#)
- GetEventPayloadDataSize
 - Spinnaker::Event, [757](#)
- GetEventType

- Spinnaker::Event, [757](#)
- GetExposureEndLineStatusAll
 - Spinnaker::ChunkData, [668](#)
 - Spinnaker::IChunkData, [819](#)
- GetExposureTime
 - Spinnaker::ChunkData, [668](#)
 - Spinnaker::IChunkData, [820](#)
- GetFeatureBagHandle
 - Spinnaker::GenApi::CFeatureBag, [655](#)
- GetFeatures
 - Spinnaker::GenApi::CategoryNode, [615](#)
- GetFileName
 - Spinnaker::Exception, [763](#)
- GetFiles
 - GCUtilities Utility, [258](#)
- GetFloatAlias
 - Spinnaker::GenApi::IntegerNode, [926](#)
- GetFramelD
 - Spinnaker::ChunkData, [668](#)
 - Spinnaker::IChunkData, [820](#)
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [879](#)
- GetFullErrorMessage
 - Spinnaker::Exception, [763](#)
- GetFunctionName
 - Spinnaker::Exception, [763](#)
- GetGain
 - Spinnaker::ChunkData, [668](#)
 - Spinnaker::IChunkData, [820](#)
- GetGenApiVersion
 - IDeviceInfo Interface, [271](#)
 - Spinnaker::GenApi::NodeMap, [986](#)
- GetGenICamCLProtocolFolder
 - GCUtilities Utility, [259](#)
- GetGenICamCacheFolder
 - GCUtilities Utility, [258](#)
- GetGenICamLogConfig
 - GCUtilities Utility, [259](#)
- GetGuiXml
 - Spinnaker::CameraBase, [596](#)
 - Spinnaker::ICameraBase, [808](#)
- GetHeatmapColorGradient
 - Spinnaker::ImageUtilityHeatmap, [911](#)
- GetHeatmapRange
 - Spinnaker::ImageUtilityHeatmap, [912](#)
- GetHeight
 - Spinnaker::ChunkData, [669](#)
 - Spinnaker::IChunkData, [820](#)
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [879](#)
- GetHistogram
 - Spinnaker::IImageStatistics, [852](#)
 - Spinnaker::ImageStatistics, [901](#)
- GetID
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [880](#)
- GetImage
 - Spinnaker::ChunkData, [669](#)
 - Spinnaker::IChunkData, [820](#)
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [880](#)
- GetImageSize
 - Spinnaker::IImage, [841](#)
 - Spinnaker::Image, [880](#)
- GetImageStatus
 - Spinnaker::IImage, [842](#)
 - Spinnaker::Image, [880](#)
- GetImageStatusDescription
 - Spinnaker::Image, [881](#)
- GetInc
 - IFloat Interface, [282](#)
 - Spinnaker::GenApi::FloatNode, [773](#)
 - Spinnaker::GenApi::IntegerNode, [927](#)
- GetIncMode
 - IFloat Interface, [282](#)
 - Spinnaker::GenApi::FloatNode, [774](#)
 - Spinnaker::GenApi::IntegerNode, [927](#)
- GetInferenceConfidence
 - Spinnaker::ChunkData, [669](#)
 - Spinnaker::IChunkData, [820](#)
- GetInferenceResult
 - Spinnaker::ChunkData, [669](#)
 - Spinnaker::IChunkData, [821](#)
- GetInstance
 - Spinnaker::System, [1033](#)
- GetIntAlias
 - Spinnaker::GenApi::CFloatPtr, [658](#)
 - Spinnaker::GenApi::FloatNode, [774](#)
- GetIntValue
 - IEnumeration Interface, [277](#)
 - Spinnaker::GenApi::EnumNode, [749](#)
- GetInterfaceName
 - Pointer Class, [336](#)
- GetInterfaces
 - Spinnaker::ISystem, [950](#)
 - Spinnaker::System, [1034](#)
- GetLength
 - IRegister Interfaces, [314](#)
 - Spinnaker::GenApi::RegisterNode, [1014](#)
- GetLibraryVersion
 - Spinnaker::ISystem, [950](#)
 - Spinnaker::System, [1034](#)
- GetLineNumber
 - Spinnaker::Exception, [764](#)
- GetLinePitch
 - Spinnaker::ChunkData, [669](#)
 - Spinnaker::IChunkData, [821](#)
- GetLineStatusAll
 - Spinnaker::ChunkData, [670](#)
 - Spinnaker::IChunkData, [821](#)
- GetListOfValidValues
 - IFloat Interface, [282](#)
 - Spinnaker::GenApi::FloatNode, [774](#)
 - Spinnaker::GenApi::IntegerNode, [927](#)
- GetLock
 - INodeMap Interface, [299](#)
 - Spinnaker::GenApi::NodeMap, [986](#)
 - Spinnaker::GenICam::LockableObject, [959](#)

- GetLogMessage
 - Spinnaker::LoggingEventData, 963
- GetLoggingEventPriorityLevel
 - Spinnaker::ISystem, 950
 - Spinnaker::System, 1035
- GetMax
 - IFloat Interface, 282
 - Spinnaker::GenApi::FloatNode, 774
 - Spinnaker::GenApi::IntegerNode, 927
- GetMaxLength
 - IStrng Class, 319
 - Spinnaker::GenApi::StringNode, 1026
- GetMean
 - Spinnaker::IImageStatistics, 852
 - Spinnaker::ImageStatistics, 902
- GetMin
 - IFloat Interface, 283
 - Spinnaker::GenApi::FloatNode, 774
 - Spinnaker::GenApi::IntegerNode, 927
- GetModelName
 - Spinnaker::GenApi::NodeMap, 986
- GetModulePathFromFunction
 - GCUtilities Utility, 259
- GetNDC
 - Spinnaker::LoggingEventData, 964
- GetName
 - Spinnaker::GenApi::Node, 976
- GetNameSpace
 - INode Interface, 291
 - Spinnaker::GenApi::Node, 977
- GetNextImage
 - Spinnaker::CameraBase, 596
 - Spinnaker::ICameraBase, 808
 - Spinnaker::IDataStream, 827
- GetNextImageInternal
 - Spinnaker::IDataStream, 827
- GetNode
 - INodeMap Interface, 299
 - Spinnaker::GenApi::CNodeCallback, 685
 - Spinnaker::GenApi::NodeMap, 986
 - Spinnaker::GenApi::ValueNode, 1079
- GetNodeHandle
 - Spinnaker::GenApi::Node, 977
- GetNodeMap
 - INode Interface, 291
 - Spinnaker::CameraBase, 597
 - Spinnaker::GenApi::Node, 977
 - Spinnaker::ICameraBase, 809
 - Spinnaker::IDataStream, 827
- GetNodeMapHandle
 - Spinnaker::GenApi::NodeMap, 987
- GetNodeStatistics
 - Spinnaker::GenApi::CNodeMapFactory, 692
- GetNodes
 - Spinnaker::GenApi::NodeMap, 987
- GetNumChannels
 - Spinnaker::IImage, 842
 - Spinnaker::Image, 881
- GetNumDataStreams
 - Spinnaker::CameraBase, 597
 - Spinnaker::ICameraBase, 809
- GetNumImagesInUse
 - Spinnaker::CameraBase, 598
 - Spinnaker::ICameraBase, 809
 - Spinnaker::IDataStream, 827
- GetNumNodes
 - INodeMap Interface, 299
 - Spinnaker::GenApi::NodeMap, 987
- GetNumPixelValues
 - Spinnaker::IImageStatistics, 852
 - Spinnaker::ImageStatistics, 902
- GetNumReads
 - Spinnaker::GenApi::CTestPortStruct, 727
- GetNumWrites
 - Spinnaker::GenApi::CTestPortStruct, 727
- GetNumericValue
 - IEnumEntry Interface, 274
 - Spinnaker::GenApi::EnumEntryNode, 744
- GetOffsetX
 - Spinnaker::ChunkData, 670
 - Spinnaker::IChunkData, 821
- GetOffsetY
 - Spinnaker::ChunkData, 670
 - Spinnaker::IChunkData, 821
- GetParents
 - INode Interface, 291
 - Spinnaker::GenApi::Node, 977
- GetPartSelector
 - Spinnaker::ChunkData, 670
 - Spinnaker::IChunkData, 821
- GetPayloadType
 - Spinnaker::IImage, 842
 - Spinnaker::Image, 881
- GetPixelDynamicRangeMax
 - Spinnaker::ChunkData, 671
 - Spinnaker::IChunkData, 822
- GetPixelDynamicRangeMin
 - Spinnaker::ChunkData, 671
 - Spinnaker::IChunkData, 822
- GetPixelFormat
 - Spinnaker::IImage, 842
 - Spinnaker::Image, 882
- GetPixelFormatIntType
 - Spinnaker::IImage, 842
 - Spinnaker::Image, 882
- GetPixelFormatName
 - Spinnaker::IImage, 842
 - Spinnaker::Image, 882
- GetPixelValueRange
 - Spinnaker::IImageStatistics, 853
 - Spinnaker::ImageStatistics, 902
- GetPollingTime
 - INode Interface, 292
 - Spinnaker::GenApi::Node, 977
- GetPort
 - Spinnaker::IDataStream, 828

- GetPortHandle
 - Spinnaker::GenApi::PortNode, [1002](#)
- GetPortReplayHandle
 - Spinnaker::GenApi::PortReplay, [1009](#)
- GetPortWriteListHandle
 - Spinnaker::GenApi::CPortWriteList, [717](#)
- GetPrincipalInterfaceType
 - INode Interface, [292](#)
 - Spinnaker::GenApi::CChunkPort, [631](#)
 - Spinnaker::GenApi::CEventPort, [652](#)
 - Spinnaker::GenApi::CTestPortStruct, [727](#)
 - Spinnaker::GenApi::Node, [978](#)
- GetPriority
 - Spinnaker::LoggingEventData, [964](#)
- GetPriorityName
 - Spinnaker::LoggingEventData, [964](#)
- GetPrivateData
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [883](#)
- GetProductGuid
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::NodeMap, [987](#)
- GetProperty
 - INode Interface, [292](#)
 - Spinnaker::GenApi::Node, [978](#)
- GetPropertyNames
 - INode Interface, [292](#)
 - Spinnaker::GenApi::Node, [978](#)
- GetRange
 - Spinnaker::ImageStatistics, [853](#)
 - Spinnaker::ImageStatistics, [903](#)
- GetRepresentation
 - IFloat Interface, [283](#)
 - Spinnaker::GenApi::FloatNode, [774](#)
 - Spinnaker::GenApi::IntegerNode, [927](#)
- GetScan3dAxisMax
 - Spinnaker::ChunkData, [671](#)
 - Spinnaker::IChunkData, [822](#)
- GetScan3dAxisMin
 - Spinnaker::ChunkData, [671](#)
 - Spinnaker::IChunkData, [822](#)
- GetScan3dCoordinateOffset
 - Spinnaker::ChunkData, [672](#)
 - Spinnaker::IChunkData, [822](#)
- GetScan3dCoordinateReferenceValue
 - Spinnaker::ChunkData, [672](#)
 - Spinnaker::IChunkData, [822](#)
- GetScan3dCoordinateScale
 - Spinnaker::ChunkData, [672](#)
 - Spinnaker::IChunkData, [823](#)
- GetScan3dInvalidDataValue
 - Spinnaker::ChunkData, [672](#)
 - Spinnaker::IChunkData, [823](#)
- GetScan3dTransformValue
 - Spinnaker::ChunkData, [673](#)
 - Spinnaker::IChunkData, [823](#)
- GetScanLineSelector
 - Spinnaker::ChunkData, [673](#)
- Spinnaker::IChunkData, [823](#)
- GetSchemaVersion
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::NodeMap, [987](#)
- GetSelectedFeatures
 - ISelector Interface, [315](#)
 - Spinnaker::GenApi::Node, [978](#)
- GetSelectingFeatures
 - ISelector Interface, [315](#)
 - Spinnaker::GenApi::Node, [978](#)
- GetSelectorList
 - ISelectorDigit Interface, [316](#)
 - Spinnaker::GenApi::CSelectorSet, [724](#)
- GetSequencerSetActive
 - Spinnaker::ChunkData, [673](#)
 - Spinnaker::IChunkData, [823](#)
- GetSerialDataLength
 - Spinnaker::ChunkData, [673](#)
 - Spinnaker::IChunkData, [823](#)
- GetSize
 - Spinnaker::CameraList, [610](#)
 - Spinnaker::ICameraList, [816](#)
 - Spinnaker::IInterfaceList, [862](#)
 - Spinnaker::InterfaceList, [940](#)
- GetStandardNameSpace
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::NodeMap, [987](#)
- GetStatistics
 - Spinnaker::ImageStatistics, [853](#)
 - Spinnaker::ImageStatistics, [903](#)
- GetStreamChannelID
 - Spinnaker::ChunkData, [674](#)
 - Spinnaker::IChunkData, [824](#)
- GetStride
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [883](#)
- GetSupportedSchemaVersions
 - INodeMapDyn Interface, [302](#)
 - Spinnaker::GenApi::CNodeMapFactory, [692](#)
 - Spinnaker::GenApi::NodeMap, [988](#)
- GetSwapEndianness
 - IPortConstruct Interface, [310](#)
 - Spinnaker::GenApi::CChunkPort, [631](#)
 - Spinnaker::GenApi::CEventPort, [652](#)
 - Spinnaker::GenApi::CPortImpl, [714](#)
 - Spinnaker::GenApi::PortNode, [1002](#)
- GetSymbolic
 - IEnumEntry Interface, [274](#)
 - Spinnaker::GenApi::EnumEntryNode, [745](#)
- GetSymbolics
 - Spinnaker::GenApi::EnumNode, [749](#)
- GetTLDeviceNodeMap
 - Spinnaker::CameraBase, [598](#)
 - Spinnaker::ICameraBase, [809](#)
- GetTLNodeMap
 - Spinnaker::IInterface, [856](#)
 - Spinnaker::ISystem, [950](#)
 - Spinnaker::Interface, [932](#)

- Spinnaker::System, [1035](#)
- GetTLPayloadType
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [884](#)
- GetTLPixelFormat
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [884](#)
- GetTLPixelFormatNamespace
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [884](#)
- GetTLStreamNodeMap
 - Spinnaker::CameraBase, [598](#)
 - Spinnaker::ICameraBase, [809](#)
- GetThreadName
 - Spinnaker::LoggingEventData, [964](#)
- GetTimeStamp
 - Spinnaker::Image, [843](#)
 - Spinnaker::Image, [883](#)
- GetTimerValue
 - Spinnaker::ChunkData, [674](#)
 - Spinnaker::IChunkData, [824](#)
- GetTimestamp
 - Spinnaker::ChunkData, [674](#)
 - Spinnaker::IChunkData, [824](#)
 - Spinnaker::LoggingEventData, [965](#)
- GetTimestampLatchValue
 - Spinnaker::ChunkData, [674](#)
 - Spinnaker::IChunkData, [824](#)
- GetToolTip
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::Node, [979](#)
 - Spinnaker::GenApi::NodeMap, [988](#)
- GetTransferBlockID
 - Spinnaker::ChunkData, [675](#)
 - Spinnaker::IChunkData, [824](#)
- GetTransferQueueCurrentBlockCount
 - Spinnaker::ChunkData, [675](#)
 - Spinnaker::IChunkData, [824](#)
- GetUniqueID
 - Spinnaker::CameraBase, [599](#)
 - Spinnaker::ICameraBase, [809](#)
- GetUnit
 - IFloat Interface, [283](#)
 - Spinnaker::GenApi::FloatNode, [775](#)
 - Spinnaker::GenApi::IntegerNode, [928](#)
- GetUserBufferCount
 - Spinnaker::CameraBase, [599](#)
 - Spinnaker::ICameraBase, [810](#)
- GetUserBufferSize
 - Spinnaker::CameraBase, [599](#)
 - Spinnaker::ICameraBase, [810](#)
- GetUserBufferTotalSize
 - Spinnaker::CameraBase, [600](#)
 - Spinnaker::ICameraBase, [810](#)
- GetValidPayloadSize
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [885](#)
- GetValue
 - IBoolean Interface, [263](#)
 - Spinnaker::GenApi::BooleanNode, [438](#)
 - Spinnaker::GenApi::CEnumerationTRef, [636](#)
 - Spinnaker::GenApi::Counter, [706](#)
 - Spinnaker::GenApi::EnumEntryNode, [745](#)
 - Spinnaker::GenApi::FloatNode, [775](#)
 - Spinnaker::GenApi::IntegerNode, [928](#)
 - Spinnaker::GenApi::StringNode, [1026](#)
- GetValueOfEnvironmentVariable
 - GCUtilities Utility, [259](#), [260](#)
- GetVendorName
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::NodeMap, [988](#)
- GetVersionGuid
 - IDeviceInfo Interface, [272](#)
 - Spinnaker::GenApi::NodeMap, [988](#)
- GetVisibility
 - INode Interface, [292](#)
 - Spinnaker::GenApi::Node, [979](#)
- GetWidth
 - Spinnaker::ChunkData, [675](#)
 - Spinnaker::IChunkData, [825](#)
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [885](#)
- GetXOffset
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [885](#)
- GetXPadding
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [886](#)
- GetYOffset
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [886](#)
- GetYPadding
 - Spinnaker::Image, [844](#)
 - Spinnaker::Image, [886](#)
- getline
 - Spinnaker::GenICam, [422](#)
- GevActionDeviceKey
 - Spinnaker::TransportLayerInterface, [1058](#)
- GevActionGroupKey
 - Spinnaker::TransportLayerInterface, [1058](#)
- GevActionGroupMask
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevActionTime
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevActiveLinkCount
 - Spinnaker::Camera, [547](#)
- GevCCPEnum
 - TransportLayerDefs Class, [199](#)
- GevCCPEnums
 - CameraDefs Class, [111](#)
- GevCCP
 - Spinnaker::Camera, [547](#)
 - Spinnaker::TransportLayerDevice, [1050](#)
- GevCurrentDefaultGateway
 - Spinnaker::Camera, [547](#)
- GevCurrentIPAddress

- Spinnaker::Camera, [548](#)
- GevCurrentIPConfigurationDHCP
 - Spinnaker::Camera, [548](#)
- GevCurrentIPConfigurationLLA
 - Spinnaker::Camera, [548](#)
- GevCurrentIPConfigurationPersistentIP
 - Spinnaker::Camera, [548](#)
- GevCurrentPhysicalLinkConfiguration
 - Spinnaker::Camera, [548](#)
- GevCurrentPhysicalLinkConfigurationEnums
 - CameraDefs Class, [111](#)
- GevCurrentSubnetMask
 - Spinnaker::Camera, [548](#)
- GevDeviceDiscoverMaximumPacketSize
 - Spinnaker::TransportLayerDevice, [1050](#)
- GevDeviceForceIP
 - Spinnaker::TransportLayerDevice, [1050](#)
- GevDeviceGateway
 - Spinnaker::TransportLayerDevice, [1051](#)
- GevDeviceIPAddress
 - Spinnaker::TransportLayerDevice, [1051](#)
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevDevicesWrongSubnet
 - Spinnaker::TransportLayerDevice, [1051](#)
- GevDeviceMACAddress
 - Spinnaker::TransportLayerDevice, [1051](#)
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevDeviceMaximumPacketSize
 - Spinnaker::TransportLayerDevice, [1051](#)
- GevDeviceMaximumRetryCount
 - Spinnaker::TransportLayerDevice, [1051](#)
- GevDeviceModelsBigEndian
 - Spinnaker::TransportLayerDevice, [1052](#)
- GevDevicePort
 - Spinnaker::TransportLayerDevice, [1052](#)
- GevDeviceReadAndWriteTimeout
 - Spinnaker::TransportLayerDevice, [1052](#)
- GevDeviceSubnetMask
 - Spinnaker::TransportLayerDevice, [1052](#)
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevDiscoveryAckDelay
 - Spinnaker::Camera, [549](#)
- GevFailedPacketCount
 - Spinnaker::TransportLayerStream, [1066](#)
- GevFirstURL
 - Spinnaker::Camera, [549](#)
- GevGVCPExtendedStatusCodes
 - Spinnaker::Camera, [549](#)
- GevGVCPExtendedStatusCodesSelector
 - Spinnaker::Camera, [549](#)
- GevGVCPExtendedStatusCodesSelectorEnums
 - CameraDefs Class, [111](#)
- GevGVCPHeartbeatDisable
 - Spinnaker::Camera, [549](#)
- GevGVCPPendingAck
 - Spinnaker::Camera, [549](#)
- GevGVCPPendingTimeout
 - Spinnaker::Camera, [550](#)
- GevGVSPExtendedIDMode
 - Spinnaker::Camera, [550](#)
- GevGVSPExtendedIDModeEnums
 - CameraDefs Class, [112](#)
- GevHeartbeatTimeout
 - Spinnaker::Camera, [550](#)
- GevIEEE1588
 - Spinnaker::Camera, [550](#)
- GevIEEE1588ClockAccuracy
 - Spinnaker::Camera, [550](#)
- GevIEEE1588ClockAccuracyEnums
 - CameraDefs Class, [112](#)
- GevIEEE1588Mode
 - Spinnaker::Camera, [550](#)
- GevIEEE1588ModeEnums
 - CameraDefs Class, [112](#)
- GevIEEE1588Status
 - Spinnaker::Camera, [551](#)
- GevIEEE1588StatusEnums
 - CameraDefs Class, [113](#)
- GevIPConfigurationStatus
 - Spinnaker::Camera, [551](#)
- GevIPConfigurationStatusEnums
 - CameraDefs Class, [113](#)
- GevInterfaceGateway
 - Spinnaker::TransportLayerInterface, [1059](#)
- GevInterfaceIPAddress
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevInterfaceMACAddress
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevInterfaceMTU
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevInterfaceReceiveLinkSpeed
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevInterfaceSelector
 - Spinnaker::Camera, [551](#)
- GevInterfaceSubnetMask
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevInterfaceTransmitLinkSpeed
 - Spinnaker::TransportLayerInterface, [1060](#)
- GevMACAddress
 - Spinnaker::Camera, [551](#)
- GevMCDA
 - Spinnaker::Camera, [551](#)
- GevMCPHostPort
 - Spinnaker::Camera, [551](#)
- GevMCRC
 - Spinnaker::Camera, [552](#)
- GevMCSP
 - Spinnaker::Camera, [552](#)
- GevMCTT
 - Spinnaker::Camera, [552](#)
- GevMaximumNumberResendBuffers
 - Spinnaker::TransportLayerStream, [1066](#)
- GevMaximumNumberResendRequests
 - Spinnaker::TransportLayerStream, [1066](#)
- GevNumberOfInterfaces
 - Spinnaker::Camera, [552](#)

- GevPAUSEFrameReception
 - Spinnaker::Camera, [552](#)
- GevPAUSEFrameTransmission
 - Spinnaker::Camera, [552](#)
- GevPacketResendMode
 - Spinnaker::TransportLayerStream, [1066](#)
- GevPacketResendTimeout
 - Spinnaker::TransportLayerStream, [1066](#)
- GevPersistentDefaultGateway
 - Spinnaker::Camera, [553](#)
- GevPersistentIPAddress
 - Spinnaker::Camera, [553](#)
- GevPersistentSubnetMask
 - Spinnaker::Camera, [553](#)
- GevPhysicalLinkConfiguration
 - Spinnaker::Camera, [553](#)
- GevPhysicalLinkConfigurationEnums
 - CameraDefs Class, [113](#)
- GevPrimaryApplicationIPAddress
 - Spinnaker::Camera, [553](#)
- GevPrimaryApplicationSocket
 - Spinnaker::Camera, [553](#)
- GevPrimaryApplicationSwitchoverKey
 - Spinnaker::Camera, [554](#)
- GevResendPacketCount
 - Spinnaker::TransportLayerStream, [1067](#)
- GevResendRequestCount
 - Spinnaker::TransportLayerStream, [1067](#)
- GevSCCFGAllInTransmission
 - Spinnaker::Camera, [554](#)
- GevSCCFGExtendedChunkData
 - Spinnaker::Camera, [554](#)
- GevSCCFGPacketResendDestination
 - Spinnaker::Camera, [554](#)
- GevSCCFGUnconditionalStreaming
 - Spinnaker::Camera, [554](#)
- GevSCDA
 - Spinnaker::Camera, [554](#)
- GevSCPDDirection
 - Spinnaker::Camera, [555](#)
- GevSCPHostPort
 - Spinnaker::Camera, [555](#)
- GevSCPIInterfaceIndex
 - Spinnaker::Camera, [555](#)
- GevSCPSBigEndian
 - Spinnaker::Camera, [555](#)
- GevSCPSDoNotFragment
 - Spinnaker::Camera, [555](#)
- GevSCPSFireTestPacket
 - Spinnaker::Camera, [556](#)
- GevSCPSPacketSize
 - Spinnaker::Camera, [556](#)
- GevSCPD
 - Spinnaker::Camera, [555](#)
- GevSCSP
 - Spinnaker::Camera, [556](#)
- GevSCZoneConfigurationLock
 - Spinnaker::Camera, [556](#)
- GevSCZoneCount
 - Spinnaker::Camera, [556](#)
- GevSCZoneDirectionAll
 - Spinnaker::Camera, [556](#)
- GevSecondURL
 - Spinnaker::Camera, [557](#)
- GevStreamChannelSelector
 - Spinnaker::Camera, [557](#)
- GevSupportedOption
 - Spinnaker::Camera, [557](#)
- GevSupportedOptionSelector
 - Spinnaker::Camera, [557](#)
- GevSupportedOptionSelectorEnums
 - CameraDefs Class, [114](#)
- GevTimestampTickFrequency
 - Spinnaker::Camera, [557](#)
- GevTotalPacketCount
 - Spinnaker::TransportLayerStream, [1067](#)
- GevVersionMajor
 - Spinnaker::TransportLayerDevice, [1052](#)
- GevVersionMinor
 - Spinnaker::TransportLayerDevice, [1052](#)
- GuiXmlManifestAddress
 - Spinnaker::Camera, [557](#)
- Guru
 - Types Enums, [361](#)
- H264Option, [800](#)
 - Spinnaker::Video::H264Option, [801](#)
- HasCRC
 - Spinnaker::GenApi::CChunkAdapterDcam, [621](#)
 - Spinnaker::Image, [845](#)
 - Spinnaker::Image, [887](#)
- HasInc
 - IFloat Interface, [283](#)
 - Spinnaker::GenApi::FloatNode, [775](#)
- Header
 - GVCP_EVENT_REQUEST_EXTENDED_ID, [797](#)
 - GVCP_EVENT_REQUEST, [796](#)
 - GVCP_EVENTDATA_REQUEST_EXTENDED_ID, [799](#)
 - GVCP_EVENTDATA_REQUEST, [798](#)
- HeatmapColor
 - Spinnaker::ImageUtilityHeatmap, [910](#)
- Height
 - Spinnaker::Camera, [558](#)
- height
 - Spinnaker::Video::H264Option, [802](#)
- HeightMax
 - Spinnaker::Camera, [558](#)
- HostAdapterDriverVersion
 - Spinnaker::TransportLayerInterface, [1061](#)
- HostAdapterName
 - Spinnaker::TransportLayerInterface, [1061](#)
- HostAdapterVendor
 - Spinnaker::TransportLayerInterface, [1061](#)
- IArrivalEvent, [803](#)
 - Spinnaker::IArrivalEvent, [804](#)

- IBase
 - IBase Interface, [228](#)
- IBase Interface, [228](#)
 - IBase, [228](#)
- IBoolean
 - IBoolean Interface, [264](#)
- IBoolean Interface, [263](#)
 - GetValue, [263](#)
 - IBoolean, [264](#)
 - operator(), [264](#)
 - operator=, [264](#)
 - Verify, [264](#)
- ICameraBase, [805](#)
 - Spinnaker::ICameraBase, [807](#)
 - Spinnaker::TransportLayerDevice, [1047](#)
 - Spinnaker::TransportLayerStream, [1066](#)
- ICameraList, [814](#)
 - Spinnaker::ICameraList, [815](#)
- ICategory
 - ICategory Interfaces, [265](#)
- ICategory Interfaces, [265](#)
 - ICategory, [265](#)
- IChunkData, [817](#)
 - Spinnaker::IChunkData, [819](#)
- IChunkData Class, [208](#)
- IChunkPort
 - IChunkPort Interface, [267](#)
- IChunkPort Interface, [266](#)
 - CHUNK_BASE_ADDRESS_REGISTER_LEN, [266](#)
 - CHUNK_BASE_ADDRESS_REGISTER, [266](#)
 - CHUNK_LENGTH_REGISTER_LEN, [267](#)
 - CHUNK_LENGTH_REGISTER, [267](#)
 - CacheChunkData, [267](#)
 - IChunkPort, [267](#)
- ICommand
 - ICommand Interface, [268](#)
- ICommand Interface, [268](#)
 - ICommand, [268](#)
 - IsDone, [268](#)
- IDataStream, [825](#)
 - Spinnaker::Event, [758](#)
 - Spinnaker::IDataStream, [826](#)
 - Spinnaker::Image, [892](#)
- IDestroy
 - IDestroy Interface, [270](#)
- IDestroy Interface, [270](#)
 - IDestroy, [270](#)
- IDevFileStream
 - Spinnaker::GenApi, [418](#)
- IDevFileStreamBase< CharType, Traits >, [830](#)
- IDevFileStreamBuf
 - Spinnaker::GenApi::IDevFileStreamBuf, [833](#)
- IDevFileStreamBuf< CharType, Traits >, [832](#)
- IDeviceEvent, [834](#)
 - Spinnaker::IDeviceEvent, [835](#)
- IDeviceInfo
 - IDeviceInfo Interface, [273](#)
- IDeviceInfo Interface, [271](#)
 - GetDeviceVersion, [271](#)
 - GetGenApiVersion, [271](#)
 - GetProductGuid, [272](#)
 - GetSchemaVersion, [272](#)
 - GetStandardNameSpace, [272](#)
 - GetToolTip, [272](#)
 - GetVendorName, [272](#)
 - GetVersionGuid, [272](#)
 - IDeviceInfo, [273](#)
- IEnumEntry
 - IEnumEntry Interface, [275](#)
- IEnumEntry Interface, [274](#)
 - GetNumericValue, [274](#)
 - GetSymbolic, [274](#)
 - IEnumEntry, [275](#)
 - IsSelfClearing, [274](#)
- IEnumReference
 - IEnumerationT Interface, [280](#)
- IEnumeration
 - IEnumeration Interface, [278](#)
- IEnumeration Interface, [276](#)
 - GetCurrentEntry, [276](#)
 - GetEntries, [276](#)
 - GetEntry, [277](#)
 - GetEntryByName, [277](#)
 - GetIntValue, [277](#)
 - IEnumeration, [278](#)
 - operator*, [277](#)
 - SetIntValue, [278](#)
- IEnumerationT Interface, [279](#)
 - GetEntry, [279](#)
 - IEnumReference, [280](#)
 - IEnumerationT, [280](#)
 - operator=, [279](#), [280](#)
- IEnumerationT
 - IEnumerationT Interface, [280](#)
- IFloat
 - IFloat Interface, [284](#)
- IFloat Interface, [281](#)
 - GetDisplayNotation, [282](#)
 - GetDisplayPrecision, [282](#)
 - GetInc, [282](#)
 - GetIncMode, [282](#)
 - GetListOfValidValues, [282](#)
 - GetMax, [282](#)
 - GetMin, [283](#)
 - GetRepresentation, [283](#)
 - GetUnit, [283](#)
 - HasInc, [283](#)
 - IFloat, [284](#)
 - ImposeMax, [283](#)
 - ImposeMin, [283](#)
 - operator=, [284](#)
- Image, [837](#)
 - Spinnaker::Image, [838](#)
- Image Class, [209](#)
- ImageEvent, [848](#)
 - Spinnaker::ImageEvent, [849](#)

- IIImageStatistics, [850](#)
 - Spinnaker::IIImageStatistics, [851](#)
- IIImageStatistics Class, [210](#)
- IIInteger
 - IIInteger Interface, [286](#)
- IIInteger Interface, [285](#)
 - IIInteger, [286](#)
 - ImposeMax, [285](#)
 - ImposeMin, [285](#)
 - operator=, [285](#)
- IIInterface, [854](#)
 - Spinnaker::IIInterface, [855](#), [856](#)
 - Spinnaker::TransportLayerInterface, [1056](#)
- IIInterface Class, [211](#)
- IIInterfaceEvent, [858](#)
 - Spinnaker::IIInterfaceEvent, [859](#)
- IIInterfaceList, [861](#)
 - Spinnaker::IIInterfaceList, [862](#)
- IIInterfaceList Class, [212](#)
- ILoggingEvent, [863](#)
 - Spinnaker::ILoggingEvent, [864](#)
- INTEGRAL_CAST2
 - GCUtilities Utility, [260](#)
- INTEGRAL_CAST
 - GCUtilities Utility, [260](#)
- INode
 - INode Interface, [297](#)
- INode Interface, [287](#)
 - Combine, [289](#)
 - DeregisterCallback, [289](#)
 - GetAlias, [290](#)
 - GetCachingMode, [290](#)
 - GetCastAlias, [290](#)
 - GetChildren, [290](#)
 - GetDescription, [291](#)
 - GetDisplayName, [291](#)
 - GetDocuURL, [291](#)
 - GetEventID, [291](#)
 - GetNameSpace, [291](#)
 - GetNodeMap, [291](#)
 - GetParents, [291](#)
 - GetPollingTime, [292](#)
 - GetPrincipalInterfaceType, [292](#)
 - GetProperty, [292](#)
 - GetPropertyNames, [292](#)
 - GetVisibility, [292](#)
 - INode, [297](#)
 - IRReference, [297](#)
 - ImposeAccessMode, [292](#)
 - ImposeVisibility, [293](#)
 - InvalidateNode, [293](#)
 - IsAccessModeCacheable, [293](#)
 - IsAvailable, [293](#)
 - IsCacheable, [294](#)
 - IsCacheable, [294](#)
 - IsDeprecated, [294](#)
 - IsFeature, [294](#)
 - IsImplemented, [294](#), [295](#)
 - IsReadable, [295](#)
 - IsStreamable, [295](#)
 - IsVisible, [295](#)
 - IsWritable, [296](#)
 - operator!=, [296](#)
 - operator==, [296](#)
 - RegisterCallback, [296](#)
- INodeMap
 - INodeMap Interface, [300](#)
- INodeMap Interface, [298](#)
 - Connect, [298](#), [299](#)
 - GetDeviceName, [299](#)
 - GetLock, [299](#)
 - GetNode, [299](#)
 - GetNumNodes, [299](#)
 - INodeMap, [300](#)
 - InvalidateNodes, [300](#)
 - Poll, [300](#)
- INodeMapDyn
 - INodeMapDyn Interface, [305](#)
- INodeMapDyn Interface, [301](#)
 - ExtractIndependentSubtree, [302](#)
 - GetSupportedSchemaVersions, [302](#)
 - INodeMapDyn, [305](#)
 - LoadXMLFromFile, [302](#)
 - LoadXMLFromFileInject, [302](#)
 - LoadXMLFromString, [303](#)
 - LoadXMLFromStringInject, [303](#)
 - LoadXMLFromZIPData, [303](#)
 - LoadXMLFromZIPFile, [303](#)
 - MergeXMLFiles, [303](#)
 - PreprocessXMLFromFile, [304](#)
 - PreprocessXMLFromZIPFile, [304](#)
- IPersistScript
 - Spinnaker::GenApi, [420](#)
- IPort
 - IPort Interface, [309](#)
- IPort Interface, [308](#)
 - Address, [308](#)
 - IPort, [309](#)
 - Length, [309](#)
 - Write, [308](#)
- IPortConstruct
 - IPortConstruct Interface, [310](#)
- IPortConstruct Interface, [310](#)
 - GetSwapEndianness, [310](#)
 - IPortConstruct, [310](#)
- IPortRecorder
 - IPortRecorder Interface, [312](#)
- IPortRecorder Interface, [311](#)
 - GetCookie, [311](#)
 - IPortRecorder, [312](#)
 - IPortReplay, [312](#)
 - IPortWriteList, [312](#)
 - Invalidate, [312](#)
 - Replay, [311](#)
 - SetCookie, [312](#)
 - StopRecording, [312](#)

- IPortReplay
 - IPortRecorder Interface, [312](#)
- IPortWriteList
 - IPortRecorder Interface, [312](#)
- IReference
 - INode Interface, [297](#)
- IRegister
 - IRegister Interfaces, [314](#)
- IRegister Interfaces, [313](#)
 - Get, [313](#)
 - GetAddress, [314](#)
 - GetLength, [314](#)
 - IRegister, [314](#)
- IRemovalEvent, [946](#)
 - Spinnaker::IRemovalEvent, [947](#)
- ISelector
 - ISelector Interface, [315](#)
- ISelector Interface, [315](#)
 - GetSelectedFeatures, [315](#)
 - GetSelectingFeatures, [315](#)
 - ISelector, [315](#)
- ISelectorDigit
 - ISelectorDigit Interface, [317](#)
- ISelectorDigit Interface, [316](#)
 - GetSelectorList, [316](#)
 - ISelectorDigit, [317](#)
 - Restore, [317](#)
 - SetNext, [317](#)
 - ToString, [317](#)
- IString
 - IString Class, [319](#)
- IString Class, [319](#)
 - GetMaxLength, [319](#)
 - IString, [319](#)
- ISystem, [948](#)
 - Spinnaker::ISystem, [949](#)
 - Spinnaker::TransportLayerSystem, [1071](#)
- ISystem Class, [213](#)
- IValue
 - IValue Class, [321](#)
- IValue Class, [320](#)
 - FromString, [320](#)
 - IValue, [321](#)
 - IsValueCacheValid, [321](#)
 - ToString, [321](#)
- Image, [865](#)
 - Spinnaker::Image, [870](#)
- Image Class, [161](#)
- Image Utility Class, [165](#)
- Image Utility Heatmap Class, [166](#)
- Image Utility Polarization Class, [167](#)
- ImageComponentEnable
 - Spinnaker::Camera, [558](#)
- ImageComponentSelector
 - Spinnaker::Camera, [558](#)
- ImageComponentSelectorEnums
 - CameraDefs Class, [115](#)
- ImageCompressionBitrate
 - Spinnaker::Camera, [558](#)
- ImageCompressionJPEGFormatOption
 - Spinnaker::Camera, [558](#)
- ImageCompressionJPEGFormatOptionEnums
 - CameraDefs Class, [115](#)
- ImageCompressionMode
 - Spinnaker::Camera, [559](#)
- ImageCompressionModeEnums
 - CameraDefs Class, [116](#)
- ImageCompressionQuality
 - Spinnaker::Camera, [559](#)
- ImageCompressionRateOption
 - Spinnaker::Camera, [559](#)
- ImageCompressionRateOptionEnums
 - CameraDefs Class, [116](#)
- ImageConverter
 - Spinnaker::Image, [892](#)
- ImageEvent, [893](#)
 - Spinnaker::ImageEvent, [894](#)
- ImageEvent Class, [162](#)
- ImageFileFormat
 - Spinnaker Definitions, [185](#)
- ImageFiler
 - Spinnaker::Image, [892](#)
- ImagePtr, [896](#)
 - Spinnaker::ImagePtr, [897](#)
- ImagePtr Class, [163](#)
- ImageScalingAlgorithm
 - Spinnaker::ImageUtility, [906](#)
- ImageStatistics, [898](#)
 - Spinnaker::ImageStatistics, [900](#)
- ImageStatistics Class, [164](#)
- ImageStatsCalculator
 - Spinnaker::Image, [892](#)
 - Spinnaker::ImageStatistics, [905](#)
- ImageStatus
 - Spinnaker Definitions, [186](#)
- ImageUtility, [905](#)
- ImageUtilityHeatmap, [909](#)
- ImageUtilityImpl
 - Spinnaker::Image, [893](#)
- ImageUtilityPolarization, [913](#)
- ImposeAccessMode
 - INode Interface, [292](#)
 - Spinnaker::GenApi::Node, [979](#)
- ImposeMax
 - IFloat Interface, [283](#)
 - IInteger Interface, [285](#)
 - Spinnaker::GenApi::FloatNode, [775](#)
 - Spinnaker::GenApi::IntegerNode, [928](#)
- ImposeMin
 - IFloat Interface, [283](#)
 - IInteger Interface, [285](#)
 - Spinnaker::GenApi::FloatNode, [775](#)
 - Spinnaker::GenApi::IntegerNode, [928](#)
- ImposeVisibility
 - INode Interface, [293](#)
 - Spinnaker::GenApi::Node, [979](#)

include/AVIRecorder.h, 1083
include/ArrivalEvent.h, 1081
include/BasePtr.h, 1083
include/Camera.h, 1085
include/CameraBase.h, 1087
include/CameraDefs.h, 1089
include/CameraList.h, 1122
include/CameraPtr.h, 1124
include/ChunkData.h, 1126
include/DeviceEvent.h, 1128
include/Event.h, 1130
include/Exception.h, 1132
include/Image.h, 1133
include/ImageEvent.h, 1135
include/ImagePtr.h, 1136
include/ImageStatistics.h, 1138
include/ImageUtility.h, 1140
include/ImageUtilityHeatmap.h, 1140
include/ImageUtilityPolarization.h, 1141
include/Interface.h, 1141
include/Interface/IArrivalEvent.h, 1143
include/Interface/ICameraBase.h, 1145
include/Interface/ICameraList.h, 1147
include/Interface/IChunkData.h, 1149
include/Interface/IDeviceEvent.h, 1151
include/Interface/IImage.h, 1153
include/Interface/IImageEvent.h, 1155
include/Interface/IImageStatistics.h, 1156
include/Interface/IInterface.h, 1158
include/Interface/IInterfaceEvent.h, 1160
include/Interface/IInterfaceList.h, 1162
include/Interface/ILoggingEvent.h, 1163
include/Interface/IRemovalEvent.h, 1165
include/Interface/IStream.h, 1167
include/Interface/ISystem.h, 1167
include/InterfaceEvent.h, 1169
include/InterfaceList.h, 1171
include/InterfacePtr.h, 1172
include/LoggingEvent.h, 1174
include/LoggingEventData.h, 1175
include/LoggingEventDataPtr.h, 1177
include/RemovalEvent.h, 1179
include/SpinGenApi/Autovector.h, 1181
include/SpinGenApi/Base.h, 1182
include/SpinGenApi/BooleanNode.h, 1183
include/SpinGenApi/CategoryNode.h, 1185
include/SpinGenApi/ChunkAdapter.h, 1187
include/SpinGenApi/ChunkAdapterDcam.h, 1189
include/SpinGenApi/ChunkAdapterGEV.h, 1193
include/SpinGenApi/ChunkAdapterGeneric.h, 1191
include/SpinGenApi/ChunkAdapterU3V.h, 1195
include/SpinGenApi/ChunkPort.h, 1197
include/SpinGenApi/CommandNode.h, 1199
include/SpinGenApi/Compatibility.h, 1202
include/SpinGenApi/Container.h, 1203
include/SpinGenApi/Counter.h, 1203
include/SpinGenApi/EnumClasses.h, 1204
include/SpinGenApi/EnumEntryNode.h, 1206
include/SpinGenApi/EnumNode.h, 1208
include/SpinGenApi/EnumNodeT.h, 1210
include/SpinGenApi/EventAdapter.h, 1212
include/SpinGenApi/EventAdapter1394.h, 1214
include/SpinGenApi/EventAdapterGEV.h, 1218
include/SpinGenApi/EventAdapterGeneric.h, 1216
include/SpinGenApi/EventAdapterU3V.h, 1220
include/SpinGenApi/EventPort.h, 1222
include/SpinGenApi/Filestream.h, 1224
include/SpinGenApi/FloatNode.h, 1226
include/SpinGenApi/FloatRegNode.h, 1228
include/SpinGenApi/GCBase.h, 1230
include/SpinGenApi/GCString.h, 1231
include/SpinGenApi/GCStringVector.h, 1233
include/SpinGenApi/GCSynch.h, 1234
include/SpinGenApi/GCTypes.h, 1235
include/SpinGenApi/GCUtilities.h, 1238
include/SpinGenApi/IBoolean.h, 1243
include/SpinGenApi/ICategory.h, 1245
include/SpinGenApi/IChunkPort.h, 1247
include/SpinGenApi/ICommand.h, 1249
include/SpinGenApi/IDestroy.h, 1251
include/SpinGenApi/IDeviceInfo.h, 1252
include/SpinGenApi/IEnumEntry.h, 1254
include/SpinGenApi/IEnumeration.h, 1256
include/SpinGenApi/IEnumerationT.h, 1257
include/SpinGenApi/IFloat.h, 1259
include/SpinGenApi/IInteger.h, 1261
include/SpinGenApi/INode.h, 1263
include/SpinGenApi/INodeMap.h, 1266
include/SpinGenApi/INodeMapDyn.h, 1267
include/SpinGenApi/IPort.h, 1273
include/SpinGenApi/IPortConstruct.h, 1274
include/SpinGenApi/IPortRecorder.h, 1276
include/SpinGenApi/IRegister.h, 1278
include/SpinGenApi/ISelector.h, 1280
include/SpinGenApi/ISelectorDigit.h, 1281
include/SpinGenApi/IString.h, 1283
include/SpinGenApi/IValue.h, 1285
include/SpinGenApi/IntRegNode.h, 1271
include/SpinGenApi/IntegerNode.h, 1269
include/SpinGenApi/Node.h, 1286
include/SpinGenApi/NodeCallback.h, 1288
include/SpinGenApi/NodeCallbackImpl.h, 1290
include/SpinGenApi/NodeMap.h, 1291
include/SpinGenApi/NodeMapFactory.h, 1293
include/SpinGenApi/NodeMapRef.h, 1294
include/SpinGenApi/Persistence.h, 1295
include/SpinGenApi/Pointer.h, 1297
include/SpinGenApi/PortImpl.h, 1300
include/SpinGenApi/PortNode.h, 1301
include/SpinGenApi/PortRecorder.h, 1303
include/SpinGenApi/PortReplay.h, 1304
include/SpinGenApi/PortWriteList.h, 1305
include/SpinGenApi/Reference.h, 1307
include/SpinGenApi/RegisterNode.h, 1308
include/SpinGenApi/RegisterPortImpl.h, 1310
include/SpinGenApi/SelectorSet.h, 1310

- include/SpinGenApi/SpinTestCamera.h, [1313](#)
- include/SpinGenApi/SpinnakerGenApi.h, [1311](#)
- include/SpinGenApi/StringNode.h, [1313](#)
- include/SpinGenApi/StringRegNode.h, [1315](#)
- include/SpinGenApi/StructPort.h, [1317](#)
- include/SpinGenApi/Synch.h, [1317](#)
- include/SpinGenApi/Types.h, [1318](#)
- include/SpinGenApi/ValueNode.h, [1322](#)
- include/SpinUpdate.h, [1329](#)
- include/SpinVideo.h, [1332](#)
- include/SpinVideoDefs.h, [1332](#)
- include/Spinnaker.h, [1324](#)
- include/SpinnakerDefs.h, [1325](#)
- include/SpinnakerPlatform.h, [1329](#)
- include/System.h, [1333](#)
- include/SystemPtr.h, [1335](#)
- include/TransportLayerDefs.h, [1337](#)
- include/TransportLayerDevice.h, [1339](#)
- include/TransportLayerInterface.h, [1341](#)
- include/TransportLayerStream.h, [1343](#)
- include/TransportLayerSystem.h, [1345](#)
- IncompatibleDeviceCount
 - Spinnaker::TransportLayerInterface, [1061](#)
- IncompatibleDeviceID
 - Spinnaker::TransportLayerInterface, [1061](#)
- IncompatibleDeviceModelName
 - Spinnaker::TransportLayerInterface, [1061](#)
- IncompatibleDeviceSelector
 - Spinnaker::TransportLayerInterface, [1062](#)
- IncompatibleDeviceVendorName
 - Spinnaker::TransportLayerInterface, [1062](#)
- IncompatibleGevDeviceIPAddress
 - Spinnaker::TransportLayerInterface, [1062](#)
- IncompatibleGevDeviceMACAddress
 - Spinnaker::TransportLayerInterface, [1062](#)
- IncompatibleGevDeviceSubnetMask
 - Spinnaker::TransportLayerInterface, [1062](#)
- Increasing
 - Types Enums, [360](#)
- indexedColor_8bit
 - Spinnaker::BMPOption, [436](#)
- Init
 - Spinnaker::Camera, [470](#)
 - Spinnaker::CameraBase, [600](#)
 - Spinnaker::ICameraBase, [810](#)
- InitChunkAdapter
 - Spinnaker::IDataStream, [828](#)
- int64_autovector_t, [921](#)
 - Spinnaker::GenApi::int64_autovector_t, [922](#)
- IntRegNode, [943](#)
 - Spinnaker::GenApi::IntRegNode, [944](#), [945](#)
- IntRegNode Class, [307](#)
- IntegerNode, [924](#)
 - Spinnaker::GenApi::IntegerNode, [926](#)
- IntegerNode Class, [306](#)
 - CIntegerRef, [306](#)
- Interface, [930](#)
 - Spinnaker::TransportLayerInterface, [1056](#)
- interface
 - Types.h, [1321](#)
- Interface Class, [168](#)
- InterfaceDisplayName
 - Spinnaker::TransportLayerInterface, [1062](#)
- InterfaceEvent, [935](#)
 - Spinnaker::InterfaceEvent, [936](#)
- InterfaceEvent Class, [169](#)
- InterfaceID
 - Spinnaker::TransportLayerInterface, [1063](#)
- InterfaceImpl
 - Spinnaker::CameraBase, [605](#)
 - Spinnaker::ICameraBase, [813](#)
 - Spinnaker::ICameraList, [817](#)
- InterfaceInternal
 - Spinnaker::IInterface, [857](#)
 - Spinnaker::Interface, [934](#)
 - Spinnaker::TransportLayerInterface, [1056](#)
- InterfaceList, [937](#)
 - Spinnaker::InterfaceList, [939](#)
- InterfaceList Class, [170](#)
- InterfacePtr, [941](#)
 - Spinnaker::InterfacePtr, [942](#)
- InterfacePtr Class, [171](#)
- InterfaceType
 - Spinnaker::TransportLayerInterface, [1063](#)
- interlaced
 - Spinnaker::PNGOption, [998](#)
- Invalidate
 - IPortRecorder Interface, [312](#)
- InvalidateNode
 - INode Interface, [293](#)
 - Spinnaker::GenApi::CChunkPort, [632](#)
 - Spinnaker::GenApi::CEventPort, [653](#)
 - Spinnaker::GenApi::CPortImpl, [714](#)
 - Spinnaker::GenApi::Node, [979](#)
- InvalidateNodes
 - INodeMap Interface, [300](#)
 - Spinnaker::GenApi::NodeMap, [989](#)
- InverseChunkLength
 - DCAM_CHUNK_TRAILER, [730](#)
- Invisible
 - Types Enums, [361](#)
- ios_type
 - Spinnaker::GenApi::IDevFileStreamBase, [831](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [993](#)
- is_open
 - Spinnaker::GenApi::IDevFileStreamBase, [831](#)
 - Spinnaker::GenApi::IDevFileStreamBuf, [833](#)
 - Spinnaker::GenApi::ODevFileStreamBase, [993](#)
 - Spinnaker::GenApi::ODevFileStreamBuf, [995](#)
- IsAccessModeCacheable
 - INode Interface, [293](#)
 - Spinnaker::GenApi::Node, [979](#)
- IsAvailable
 - INode Interface, [293](#)
 - Pointer Class, [336](#)
- IsCachable

- INode Interface, [294](#)
- Spinnaker::GenApi::Node, [980](#)
- IsCacheable
 - INode Interface, [294](#)
- IsCameraDescriptionFileDataReleased
 - Spinnaker::GenApi::CNodeMapFactory, [692](#)
- IsCompressed
 - Spinnaker::Image, [887](#)
- IsDeprecated
 - INode Interface, [294](#)
 - Spinnaker::GenApi::Node, [980](#)
- IsDone
 - ICommand Interface, [268](#)
 - Spinnaker::GenApi::CommandNode, [704](#)
- IsEmpty
 - Spinnaker::GenApi::CNodeMapFactory, [692](#)
 - Spinnaker::GenApi::CSelectorSet, [724](#)
- IsFeature
 - INode Interface, [294](#)
 - Spinnaker::GenApi::Node, [980](#)
- IsImageInUse
 - Spinnaker::IDataStream, [828](#)
- IsImplemented
 - INode Interface, [294](#), [295](#)
 - Pointer Class, [336](#)
- IsInUse
 - Spinnaker::Image, [845](#)
 - Spinnaker::IInterface, [856](#)
 - Spinnaker::ISystem, [950](#)
 - Spinnaker::Image, [887](#)
 - Spinnaker::Interface, [932](#)
 - Spinnaker::System, [1035](#)
- IsIncomplete
 - Spinnaker::Image, [845](#)
 - Spinnaker::Image, [887](#)
- IsInitialized
 - Spinnaker::CameraBase, [601](#)
 - Spinnaker::ICameraBase, [810](#)
- IsLoaded
 - Spinnaker::GenApi::CNodeMapFactory, [692](#)
- IsPreprocessed
 - Spinnaker::GenApi::CNodeMapFactory, [693](#)
- IsReadable
 - INode Interface, [295](#)
 - Pointer Class, [336](#)
- IsSelector
 - Spinnaker::GenApi::Node, [980](#)
- IsSelfClearing
 - IEnumEntry Interface, [274](#)
 - Spinnaker::GenApi::EnumEntryNode, [745](#)
- IsStreamable
 - INode Interface, [295](#)
 - Spinnaker::GenApi::Node, [980](#)
- IsStreaming
 - Spinnaker::CameraBase, [601](#)
 - Spinnaker::ICameraBase, [810](#)
 - Spinnaker::IDataStream, [828](#)
- IsValid
 - Spinnaker::BasePtr, [433](#)
 - Spinnaker::CameraBase, [601](#)
 - Spinnaker::GenApi::CPointer, [709](#)
 - Spinnaker::GenICam::CGlobalLock, [661](#)
 - Spinnaker::ICameraBase, [811](#)
- IsValueCacheValid
 - IValue Class, [321](#)
 - Spinnaker::GenApi::ValueNode, [1079](#)
- IsVisible
 - INode Interface, [295](#)
- IsWritable
 - INode Interface, [296](#)
 - Pointer Class, [336](#)
- IsZero
 - Spinnaker::GenApi::Counter, [706](#)
- IspEnable
 - Spinnaker::Camera, [559](#)
- istream_type
 - Spinnaker::GenApi::IDevFileStreamBase, [831](#)
- Items
 - GVCP_EVENT_REQUEST_EXTENDED_ID, [797](#)
 - GVCP_EVENT_REQUEST, [796](#)
- JPEGOOption, [953](#)
 - Spinnaker::JPEGOOption, [954](#)
- JPG2Option, [955](#)
 - Spinnaker::JPG2Option, [955](#)
- KillBufferEvent
 - Spinnaker::IDataStream, [828](#)
- LUTEnable
 - Spinnaker::Camera, [563](#)
- LUTIndex
 - Spinnaker::Camera, [563](#)
- LUTSelector
 - Spinnaker::Camera, [563](#)
- LUTSelectorEnums
 - CameraDefs Class, [120](#)
- LUTValue
 - Spinnaker::Camera, [563](#)
- LUTValueAll
 - Spinnaker::Camera, [564](#)
- Length
 - GVCP_REQUEST_HEADER, [800](#)
 - IPort Interface, [309](#)
 - U3V_COMMAND_HEADER, [1073](#)
- length
 - Spinnaker::GenICam::gcstring, [787](#)
- LibraryVersion, [956](#)
- LineFilterWidth
 - Spinnaker::Camera, [559](#)
- LineFormat
 - Spinnaker::Camera, [560](#)
- LineFormatEnums
 - CameraDefs Class, [116](#)
- LineInputFilterSelector
 - Spinnaker::Camera, [560](#)
- LineInputFilterSelectorEnums

- CameraDefs Class, [117](#)
- LineInverter
 - Spinnaker::Camera, [560](#)
- LineMode
 - Spinnaker::Camera, [560](#)
- LineModeEnums
 - CameraDefs Class, [117](#)
- LinePitch
 - Spinnaker::Camera, [560](#)
- LineSelector
 - Spinnaker::Camera, [560](#)
- LineSelectorEnums
 - CameraDefs Class, [117](#)
- LineSource
 - Spinnaker::Camera, [561](#)
- LineSourceEnums
 - CameraDefs Class, [118](#)
- LineStatus
 - Spinnaker::Camera, [561](#)
- LineStatusAll
 - Spinnaker::Camera, [561](#)
- Linear
 - Types Enums, [359](#)
- LinkErrorCount
 - Spinnaker::Camera, [561](#)
- LinkUptime
 - Spinnaker::Camera, [561](#)
- LoadAndInject
 - Spinnaker::GenApi::CNodeMapFactory, [693](#)
- LoadFromBag
 - Spinnaker::GenApi::CFeatureBag, [655](#)
- LoadXMLFromFile
 - INodeMapDyn Interface, [302](#)
 - Spinnaker::GenApi::NodeMap, [989](#)
- LoadXMLFromFileInject
 - INodeMapDyn Interface, [302](#)
 - Spinnaker::GenApi::NodeMap, [989](#)
- LoadXMLFromString
 - INodeMapDyn Interface, [303](#)
 - Spinnaker::GenApi::NodeMap, [989](#)
- LoadXMLFromStringInject
 - INodeMapDyn Interface, [303](#)
 - Spinnaker::GenApi::NodeMap, [989](#)
- LoadXMLFromZIPData
 - INodeMapDyn Interface, [303](#)
 - Spinnaker::GenApi::NodeMap, [990](#)
- LoadXMLFromZIPFile
 - INodeMapDyn Interface, [303](#)
 - Spinnaker::GenApi::NodeMap, [990](#)
- Lock
 - Spinnaker::GenApi::CLock, [677](#)
 - Spinnaker::GenICam::CGlobalLock, [661](#)
 - Spinnaker::GenICam::CLock, [680](#)
 - Spinnaker::GenICam::LockableObject, [959](#)
 - Spinnaker::GenICam::LockableObject::Lock, [958](#)
- LockableObject< Object >, [958](#)
- LockableObject< Object >::Lock, [957](#)
- Logarithmic
 - Types Enums, [359](#)
- Logging Event Class, [173](#)
- LoggingEvent, [960](#)
 - Spinnaker::LoggingEvent, [961](#)
- LoggingEvent Class, [172](#)
- LoggingEventData, [962](#)
 - Spinnaker::LoggingEventData, [963](#)
- LoggingEventDataPtr, [966](#)
 - Spinnaker::LoggingEventDataPtr, [967](#)
- LoggingEventDataPtr Class, [174](#)
- LogicBlockLUTInputActivation
 - Spinnaker::Camera, [561](#)
- LogicBlockLUTInputActivationEnums
 - CameraDefs Class, [118](#)
- LogicBlockLUTInputSelector
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTInputSelectorEnums
 - CameraDefs Class, [119](#)
- LogicBlockLUTInputSource
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTInputSourceEnums
 - CameraDefs Class, [119](#)
- LogicBlockLUTOutputValue
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTOutputValueAll
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTRowIndex
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTSelector
 - Spinnaker::Camera, [562](#)
- LogicBlockLUTSelectorEnums
 - CameraDefs Class, [120](#)
- LogicBlockSelector
 - Spinnaker::Camera, [563](#)
- LogicBlockSelectorEnums
 - CameraDefs Class, [120](#)
- m_BaseAddress
 - Spinnaker::GenApi::CTestPortStruct, [728](#)
- m_CallbackType
 - Spinnaker::GenApi::CNodeCallback, [685](#)
- m_Callbacks
 - Spinnaker::GenApi::Node, [982](#)
- m_DebugCount
 - Spinnaker::GenICam::CGlobalLock, [662](#)
- m_Lock
 - Spinnaker::GenICam::CGlobalLockUnlocker, [664](#)
 - Spinnaker::GenICam::LockableObject, [959](#)
- m_NumReads
 - Spinnaker::GenApi::CTestPortStruct, [728](#)
- m_NumWrites
 - Spinnaker::GenApi::CTestPortStruct, [729](#)
- m_bOwnLock
 - Spinnaker::GenApi::CLock, [678](#)
- m_enabled
 - Spinnaker::GenICam::CGlobalLockUnlocker, [664](#)
- m_lock
 - Spinnaker::GenApi::CLock, [678](#)
- m_lockEx

- Spinnaker::GenApi::CLOCKEx, 682
- m_pCameraBaseData
 - Spinnaker::ICameraBase, 813
- m_pCameraListData
 - Spinnaker::ICameraList, 817
- m_pChunkAdapter
 - Spinnaker::GenApi::CChunkAdapter, 618
- m_pChunkPort
 - Spinnaker::GenApi::CChunkPort, 633
- m_pEnumeration
 - Spinnaker::GenApi::EnumNode, 751
- m_pEventAdapter
 - Spinnaker::GenApi::CEventAdapter, 640
- m_pEventData
 - Spinnaker::Event, 759
- m_pEventPort
 - Spinnaker::GenApi::CEventPort, 653
- m_plImageData
 - Spinnaker::Image, 893
- m_plInterfaceData
 - Spinnaker::IInterface, 858
- m_plInterfaceListData
 - Spinnaker::IInterfaceList, 863
- m_pNode
 - Spinnaker::GenApi::CEventPort, 654
 - Spinnaker::GenApi::CNodeCallback, 685
- m_pNodeData
 - Spinnaker::GenApi::Node, 982
- m_pNodeMap
 - Spinnaker::GenApi::Node, 982
- m_pPort
 - Spinnaker::GenApi::CChunkPort, 633
- m_pPortAdapter
 - Spinnaker::GenApi::CChunkPort, 633
 - Spinnaker::GenApi::CEventPort, 654
- m_pWriteList
 - Spinnaker::GenApi::CPortWriteList, 718
- m_pT
 - Spinnaker::BasePtr, 435
 - Spinnaker::GenApi::CPointer, 712
- m_ptrPort
 - Spinnaker::GenApi::CPortImpl, 715
- MJPGOption, 970
 - Spinnaker::Video::MJPGOption, 970
- Magic
 - GVCP_REQUEST_HEADER, 800
- Major
 - Spinnaker::GenICam::Version_t, 1080
- major
 - Spinnaker::LibraryVersion, 956
- make_NodeCallback
 - NodeCallback Class, 324
- max_size
 - Spinnaker::GenICam::gcstring, 787
- MaxDeviceResetTime
 - Spinnaker::Camera, 564
- MemSet
 - Spinnaker::GenApi::CTestPortStruct, 727
- Member_NodeCallback
 - Spinnaker::GenApi::Member_NodeCallback, 969
- Member_NodeCallback< Client, Member >, 968
- MergeXMLFiles
 - INodeMapDyn Interface, 303
- Minor
 - Spinnaker::GenICam::Version_t, 1080
- minor
 - Spinnaker::LibraryVersion, 957
- NA
 - Types Enums, 355
- NI
 - Types Enums, 355
- No
 - Types Enums, 361
- Node, 971
 - Spinnaker::GenApi::Node, 974
- Node Class, 322
- NodeCallback Class, 323
 - Deregister, 324
 - ECallbackType, 324
 - make_NodeCallback, 324
 - Register, 325
- NodeList_t
 - Spinnaker GenApi Interfaces, 227
- NodeMap, 982
 - Spinnaker::GenApi::CLOCK, 678
 - Spinnaker::GenApi::NodeMap, 984
- NodeMap Class, 326
- NodeMapFactory Class, 327
 - ECacheUsage_t, 327
 - EContentType_t, 328
- NodeMapRef Class, 329
- None
 - Types Enums, 360
- npos
 - Spinnaker::GenICam::gcstring, 791
- NumAttachedChunks
 - AttachStatistics_t, 428
- NumChunkPorts
 - AttachStatistics_t, 428
- NumChunks
 - AttachStatistics_t, 428
- NumLinks
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, 991
- NumNodes
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, 991
- NumProperties
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, 991
- NumStrings
 - Spinnaker::GenApi::CNodeMapFactory::Node↔Statistics_t, 991
- ODevFileStream
 - Spinnaker::GenApi, 418

- ODevFileStreamBase< CharType, Traits >, 992
- ODevFileStreamBuf
 - Spinnaker::GenApi::ODevFileStreamBuf, 995
- ODevFileStreamBuf< CharType, Traits >, 994
- OffsetX
 - Spinnaker::Camera, 564
- OffsetY
 - Spinnaker::Camera, 564
- OnDeviceArrival
 - Spinnaker::ArrivalEvent, 427
 - Spinnaker::IArrivalEvent, 804
 - Spinnaker::IInterfaceEvent, 860
 - Spinnaker::InterfaceEvent, 936
- OnDeviceEvent
 - Spinnaker::DeviceEvent, 732
 - Spinnaker::IDeviceEvent, 836
- OnDeviceRemoval
 - Spinnaker::IInterfaceEvent, 860
 - Spinnaker::IRemovalEvent, 947
 - Spinnaker::InterfaceEvent, 937
 - Spinnaker::RemovalEvent, 1016
- OnImageEvent
 - Spinnaker::IImageEvent, 849
 - Spinnaker::ImageEvent, 895
- OnLogEvent
 - Spinnaker::ILoggingEvent, 865
 - Spinnaker::LoggingEvent, 961
- Open
 - Spinnaker::Video::SpinVideo, 1021, 1023
- open
 - Spinnaker::GenApi::IDevFileStreamBase, 831
 - Spinnaker::GenApi::IDevFileStreamBuf, 833
 - Spinnaker::GenApi::ODevFileStreamBase, 993
 - Spinnaker::GenApi::ODevFileStreamBuf, 995
- openFile
 - Spinnaker::GenApi::FileProtocolAdapter, 769
- operator bool
 - Spinnaker::BasePtr, 433
 - Spinnaker::GenApi::CPointer, 709
- operator const char *
 - Spinnaker::GenICam::gcstring, 787
- operator delete
 - Spinnaker::GenApi::double_autovector_t, 734
 - Spinnaker::GenApi::int64_autovector_t, 922
 - Spinnaker::GenICam::gcstring, 787
- operator new
 - Spinnaker::GenApi::double_autovector_t, 734
 - Spinnaker::GenApi::int64_autovector_t, 922
 - Spinnaker::GenICam::gcstring, 787
- operator T*
 - Spinnaker::BasePtr, 433
 - Spinnaker::GenApi::CPointer, 709
- operator unsigned int
 - Spinnaker::GenApi::Counter, 706
- operator!=
 - INode Interface, 296
 - Spinnaker::Exception, 764
 - Spinnaker::GenApi::CPointer, 709, 710
 - Spinnaker::GenApi::Node, 980
 - Spinnaker::GenICam::gcstring, 788
- operator<
 - Spinnaker::GenICam::gcstring, 789
- operator<<
 - GCString.h, 1232
 - Spinnaker GenApi Classes, 223
- operator>
 - Spinnaker::GenICam::gcstring, 789
- operator>>
 - GCString.h, 1233
 - Spinnaker GenApi Classes, 224
- operator*
 - IEnumeration Interface, 277
 - Spinnaker::GenApi::CPointer, 710
 - Spinnaker::GenApi::EnumNode, 750
 - Spinnaker::GenApi::FloatNode, 776
 - Spinnaker::GenApi::IntegerNode, 929
 - Spinnaker::GenApi::StringNode, 1027
- operator()
 - IBoolean Interface, 264
 - Spinnaker::GenApi::CEnumerationTRef, 636
 - Spinnaker::GenApi::CNodeCallback, 685
 - Spinnaker::GenApi::CPointer, 710
 - Spinnaker::GenApi::CommandNode, 705
 - Spinnaker::GenApi::FloatNode, 776
 - Spinnaker::GenApi::Function_NodeCallback, 781
 - Spinnaker::GenApi::IntegerNode, 928
 - Spinnaker::GenApi::Member_NodeCallback, 969
 - Spinnaker::GenApi::StringNode, 1027
- operator+
 - Spinnaker::GenICam::gcstring, 790
- operator++
 - Spinnaker::GenApi::Counter, 706
- operator+=
 - Spinnaker::GenICam::gcstring, 788
- operator->
 - Spinnaker::BasePtr, 433
 - Spinnaker::GenApi::CPointer, 711
- operator--
 - Spinnaker::GenApi::Counter, 706, 707
- operator=
 - IBoolean Interface, 264
 - IEnumerationT Interface, 279, 280
 - IFloat Interface, 284
 - IInteger Interface, 285
 - Spinnaker GenApi Classes, 224
 - Spinnaker::ArrivalEvent, 427
 - Spinnaker::BasePtr, 434
 - Spinnaker::CameraBase, 602
 - Spinnaker::CameraList, 610
 - Spinnaker::DeviceEvent, 733
 - Spinnaker::Event, 758
 - Spinnaker::Exception, 764
 - Spinnaker::GenApi::BooleanNode, 439
 - Spinnaker::GenApi::CEnumerationTRef, 637
 - Spinnaker::GenApi::CFloatPtr, 658
 - Spinnaker::GenApi::CNodeMapFactory, 693

- Spinnaker::GenApi::CNodeMapRef, 696
- Spinnaker::GenApi::CPointer, 711
- Spinnaker::GenApi::EnumNode, 750
- Spinnaker::GenApi::FloatNode, 776
- Spinnaker::GenApi::IntegerNode, 929
- Spinnaker::GenApi::StringNode, 1027
- Spinnaker::GenApi::double_autovector_t, 734
- Spinnaker::GenApi::int64_autovector_t, 922
- Spinnaker::GenICam::gcstring, 789
- Spinnaker::IArrivalEvent, 804
- Spinnaker::ICameraBase, 811
- Spinnaker::ICameraList, 816
- Spinnaker::IDeviceEvent, 836
- Spinnaker::IImageEvent, 849
- Spinnaker::IInterface, 856
- Spinnaker::IInterfaceEvent, 860
- Spinnaker::IInterfaceList, 862
- Spinnaker::ILoggingEvent, 865
- Spinnaker::IRemovalEvent, 947
- Spinnaker::ISystem, 950
- Spinnaker::ImageEvent, 895
- Spinnaker::ImagePtr, 898
- Spinnaker::ImageStatistics, 904
- Spinnaker::InterfaceEvent, 937
- Spinnaker::InterfaceList, 940
- Spinnaker::LoggingEvent, 962
- Spinnaker::RemovalEvent, 1017
- operator==
 - INode Interface, 296
 - Spinnaker::BasePtr, 434, 435
 - Spinnaker::Exception, 764
 - Spinnaker::GenApi::CFeatureBag, 656
 - Spinnaker::GenApi::CPointer, 711
 - Spinnaker::GenApi::Node, 981
 - Spinnaker::GenICam::gcstring, 789
- operator[]
 - Spinnaker::CameraList, 610
 - Spinnaker::GenApi::double_autovector_t, 735
 - Spinnaker::GenApi::int64_autovector_t, 923
 - Spinnaker::ICameraList, 816
 - Spinnaker::IInterfaceList, 863
 - Spinnaker::InterfaceList, 940
- ostream_type
 - Spinnaker::GenApi::ODevFileStreamBase, 993
- overflow
 - Spinnaker::GenApi::ODevFileStreamBuf, 996
- PGMOption, 996
 - Spinnaker::PGMOption, 997
- PMEMBERFUNC
 - Spinnaker::GenApi::Member_NodeCallback, 969
- PNGOption, 997
 - Spinnaker::PNGOption, 998
- POEStatus
 - Spinnaker::TransportLayerInterface, 1063
- POEStatusEnum
 - TransportLayerDefs Class, 200
- PPMOption, 1009
 - Spinnaker::PPMOption, 1010
- PacketResendRequestCount
 - Spinnaker::Camera, 564
- PayloadSize
 - Spinnaker::Camera, 565
- PayloadTypeInfoIds
 - Spinnaker Definitions, 186
- pbackfail
 - Spinnaker::GenApi::IDevFileStreamBuf, 834
- PersistFeature
 - Spinnaker::GenApi, 419
 - Spinnaker::GenApi::CFeatureBag, 656
- Persistence Class, 330
- PixelColorFilter
 - Spinnaker::Camera, 565
- PixelColorFilterEnums
 - CameraDefs Class, 121
- PixelDynamicRangeMax
 - Spinnaker::Camera, 565
- PixelDynamicRangeMin
 - Spinnaker::Camera, 565
- PixelFormat
 - Spinnaker::Camera, 565
- PixelFormatEnums
 - CameraDefs Class, 121
- PixelFormatInfoID
 - Spinnaker::Camera, 566
- PixelFormatInfoSelector
 - Spinnaker::Camera, 566
- PixelFormatInfoSelectorEnums
 - CameraDefs Class, 127
- PixelFormatIntType
 - Spinnaker Definitions, 187
- PixelFormatNamespaceID
 - Spinnaker Definitions, 187
- PixelSize
 - Spinnaker::Camera, 566
- PixelSizeEnums
 - CameraDefs Class, 132
- Pointer Class, 331
 - CBasePtr, 332
 - CBooleanPtr, 332
 - CCategoryPtr, 333
 - CChunkPortPtr, 333
 - CCommandPtr, 333
 - CDeviceInfoPtr, 333
 - CEnumEntryPtr, 333
 - CEnumerationPtr, 333
 - CIntegerPtr, 334
 - CNodeMapDynPtr, 334
 - CNodeMapPtr, 334
 - CNodePtr, 334
 - CPortConstructPtr, 334
 - CPortPtr, 334
 - CPortRecorderPtr, 335
 - CPortReplayPtr, 335
 - CPortWriteListPtr, 335
 - CRegisterPtr, 335
 - CSelectorPtr, 335

- CStringPtr, 335
- CValuePtr, 336
- GetInterfaceName, 336
- IsAvailable, 336
- IsImplemented, 336
- IsReadable, 336
- IsWritable, 336
- PolarizationQuadrant
 - Spinnaker::ImageUtilityPolarization, 914
- Poll
 - INodeMap Interface, 300
 - Spinnaker::GenApi::NodeMap, 990
- PortImpl Class, 337
- PortNode, 999
 - Spinnaker::GenApi::PortNode, 1001
- PortNode Class, 338
 - CPortRef, 338
- PortRecorder, 1004
 - Spinnaker::GenApi::PortRecorder, 1005
- PortRecorder Class, 339
 - CPortRecorderRef, 339
- PortReplay, 1007
 - Spinnaker::GenApi::PortReplay, 1008
- PortReplay Class, 340
- PortWriteList Class, 341
- PowerSupplyCurrent
 - Spinnaker::Camera, 566
- PowerSupplyVoltage
 - Spinnaker::Camera, 566
- Prefix
 - U3V_COMMAND_HEADER, 1073
- Preprocess
 - Spinnaker::GenApi::CNodeMapFactory, 693
- PreprocessXMLFromFile
 - INodeMapDyn Interface, 304
- PreprocessXMLFromZIPFile
 - INodeMapDyn Interface, 304
- progressive
 - Spinnaker::JPEGOption, 954
- quality
 - Spinnaker::JPEGOption, 954
 - Spinnaker::JPG2Option, 955
 - Spinnaker::Video::MJPGOption, 971
- rdbuf
 - Spinnaker::GenApi::IDevFileStreamBase, 832
 - Spinnaker::GenApi::ODevFileStreamBase, 994
- Read
 - Spinnaker::GenApi::CChunkPort, 632
 - Spinnaker::GenApi::CEventPort, 653
 - Spinnaker::GenApi::CPortImpl, 714
 - Spinnaker::GenApi::CRegisterPortImpl, 721
 - Spinnaker::GenApi::CTestPortStruct, 728
 - Spinnaker::GenApi::PortNode, 1002
- read
 - Spinnaker::GenApi::FileProtocolAdapter, 769
- ReadPort
 - Spinnaker::CameraBase, 602
 - Spinnaker::ICameraBase, 811
- ReadRegister
 - Spinnaker::GenApi::CRegisterPortImpl, 721
- Reference Interfaces, 342
 - SetNumEnums, 342
- RegionDestination
 - Spinnaker::Camera, 566
- RegionDestinationEnums
 - CameraDefs Class, 133
- RegionMode
 - Spinnaker::Camera, 567
- RegionModeEnums
 - CameraDefs Class, 133
- RegionSelector
 - Spinnaker::Camera, 567
- RegionSelectorEnums
 - CameraDefs Class, 134
- Register
 - NodeCallback Class, 325
- RegisterCallback
 - INode Interface, 296
 - Spinnaker::GenApi::Node, 981
- RegisterEvent
 - Spinnaker::CameraBase, 602, 603
 - Spinnaker::ICameraBase, 811
 - Spinnaker::IInterface, 856
 - Spinnaker::Interface, 932
- RegisterImageEvent
 - Spinnaker::IDataStream, 828
- RegisterInterfaceEvent
 - Spinnaker::ISystem, 951
 - Spinnaker::System, 1036
- RegisterLoggingEvent
 - Spinnaker::ISystem, 951
 - Spinnaker::System, 1036
- RegisterNode, 1011
 - Spinnaker::GenApi::RegisterNode, 1012, 1013
- RegisterNode Class, 343
 - CRegisterRef, 343
- RegisterPortImpl Class, 344
- Release
 - Spinnaker::IImage, 845
 - Spinnaker::Image, 888
- ReleaseCameraDescriptionFileData
 - Spinnaker::GenApi::CNodeMapFactory, 693
- ReleaseImage
 - Spinnaker::IDataStream, 828
- ReleaseInstance
 - Spinnaker::ISystem, 951
 - Spinnaker::System, 1036
- RemovalEvent, 1015
 - Spinnaker::RemovalEvent, 1016
- RemovalEvent Class, 175
- RemoveByIndex
 - Spinnaker::CameraList, 611
 - Spinnaker::ICameraList, 816
- RemoveBySerial
 - Spinnaker::CameraList, 611

- Spinnaker::ICameraList, [816](#)
- ReplaceEnvironmentVariables
 - GCUtilities Utility, [260](#)
- Replay
 - IPortRecorder Interface, [311](#)
 - Spinnaker::GenApi::CPortImpl, [714](#)
 - Spinnaker::GenApi::CPortWriteList, [717](#)
 - Spinnaker::GenApi::PortNode, [1002](#)
 - Spinnaker::GenApi::PortReplay, [1009](#)
- ReqId
 - GVCP_REQUEST_HEADER, [800](#)
 - U3V_COMMAND_HEADER, [1073](#)
- Reserved
 - U3V_EVENT_DATA, [1074](#)
- reserved
 - Spinnaker::BMPOption, [436](#)
 - Spinnaker::JPEGOption, [954](#)
 - Spinnaker::JPG2Option, [955](#)
 - Spinnaker::PGMOption, [997](#)
 - Spinnaker::PNGOption, [998](#)
 - Spinnaker::PPMOption, [1010](#)
 - Spinnaker::TIFFOption, [1043](#)
 - Spinnaker::Video::AVIOption, [431](#)
 - Spinnaker::Video::H264Option, [802](#)
 - Spinnaker::Video::MJPGOption, [971](#)
- ReservedOrEventSize
 - GVCP_EVENT_ITEM_BASIC, [793](#)
 - GVCP_EVENT_ITEM_EXTENDED_ID, [794](#)
 - GVCP_EVENT_ITEM, [792](#)
- ResetImage
 - Spinnaker::Image, [845](#)
 - Spinnaker::Image, [888](#)
- ResetStatistics
 - Spinnaker::GenApi::CTestPortStruct, [728](#)
- resize
 - Spinnaker::GenICam::gcstring, [789](#)
- Restore
 - ISelectorDigit Interface, [317](#)
 - Spinnaker::GenApi::CSelectorSet, [724](#)
- ReverseX
 - Spinnaker::Camera, [567](#)
- ReverseY
 - Spinnaker::Camera, [567](#)
- RevokeImages
 - Spinnaker::IDataStream, [829](#)
- RgbTransformLightSource
 - Spinnaker::Camera, [567](#)
- RgbTransformLightSourceEnums
 - CameraDefs Class, [134](#)
- RO
 - Types Enums, [355](#)
- RW
 - Types Enums, [355](#)
- SET_GUID
 - Spinnaker::GenApi, [419](#)
- SPINNAKER_API_ABSTRACT
 - Spinnaker Platform, [190](#)
- SPINNAKER_API
 - Spinnaker Platform, [190](#)
- SPINNAKER_LOCAL
 - Spinnaker Platform, [190](#)
- SPINUPDATE_API
 - SpinUpdate.h, [1330](#)
- Saturation
 - Spinnaker::Camera, [568](#)
- SaturationEnable
 - Spinnaker::Camera, [568](#)
- Save
 - Spinnaker::Image, [846](#), [847](#)
 - Spinnaker::Image, [889–891](#)
- Scan3dAxisMax
 - Spinnaker::Camera, [568](#)
- Scan3dAxisMin
 - Spinnaker::Camera, [568](#)
- Scan3dCoordinateOffset
 - Spinnaker::Camera, [568](#)
- Scan3dCoordinateReferenceSelector
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateReferenceSelectorEnums
 - CameraDefs Class, [134](#)
- Scan3dCoordinateReferenceValue
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateScale
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateSelector
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateSelectorEnums
 - CameraDefs Class, [135](#)
- Scan3dCoordinateSystem
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateSystemEnums
 - CameraDefs Class, [135](#)
- Scan3dCoordinateSystemReference
 - Spinnaker::Camera, [569](#)
- Scan3dCoordinateSystemReferenceEnums
 - CameraDefs Class, [135](#)
- Scan3dCoordinateTransformSelector
 - Spinnaker::Camera, [570](#)
- Scan3dCoordinateTransformSelectorEnums
 - CameraDefs Class, [136](#)
- Scan3dDistanceUnit
 - Spinnaker::Camera, [570](#)
- Scan3dDistanceUnitEnums
 - CameraDefs Class, [136](#)
- Scan3dInvalidDataFlag
 - Spinnaker::Camera, [570](#)
- Scan3dInvalidDataValue
 - Spinnaker::Camera, [570](#)
- Scan3dOutputMode
 - Spinnaker::Camera, [570](#)
- Scan3dOutputModeEnums
 - CameraDefs Class, [137](#)
- Scan3dTransformValue
 - Spinnaker::Camera, [570](#)
- SelectorSet Class, [345](#)
- SendActionCommand

- Spinnaker::IInterface, [857](#)
- Spinnaker::ISystem, [951](#)
- Spinnaker::Interface, [933](#)
- Spinnaker::System, [1037](#)
- SensorDescription
 - Spinnaker::Camera, [571](#)
- SensorDigitizationTaps
 - Spinnaker::Camera, [571](#)
- SensorDigitizationTapsEnums
 - CameraDefs Class, [137](#)
- SensorHeight
 - Spinnaker::Camera, [571](#)
- SensorShutterMode
 - Spinnaker::Camera, [571](#)
- SensorShutterModeEnums
 - CameraDefs Class, [138](#)
- SensorTaps
 - Spinnaker::Camera, [571](#)
- SensorTapsEnums
 - CameraDefs Class, [138](#)
- SensorWidth
 - Spinnaker::Camera, [571](#)
- SequencerConfigurationMode
 - Spinnaker::Camera, [572](#)
- SequencerConfigurationModeEnums
 - CameraDefs Class, [139](#)
- SequencerConfigurationValid
 - Spinnaker::Camera, [572](#)
- SequencerConfigurationValidEnums
 - CameraDefs Class, [139](#)
- SequencerFeatureEnable
 - Spinnaker::Camera, [572](#)
- SequencerMode
 - Spinnaker::Camera, [572](#)
- SequencerModeEnums
 - CameraDefs Class, [139](#)
- SequencerPathSelector
 - Spinnaker::Camera, [572](#)
- SequencerSetActive
 - Spinnaker::Camera, [573](#)
- SequencerSetLoad
 - Spinnaker::Camera, [573](#)
- SequencerSetNext
 - Spinnaker::Camera, [573](#)
- SequencerSetSave
 - Spinnaker::Camera, [573](#)
- SequencerSetSelector
 - Spinnaker::Camera, [573](#)
- SequencerSetStart
 - Spinnaker::Camera, [574](#)
- SequencerSetValid
 - Spinnaker::Camera, [574](#)
- SequencerSetValidEnums
 - CameraDefs Class, [139](#)
- SequencerTriggerActivation
 - Spinnaker::Camera, [574](#)
- SequencerTriggerActivationEnums
 - CameraDefs Class, [140](#)
- SequencerTriggerSource
 - Spinnaker::Camera, [574](#)
- SequencerTriggerSourceEnums
 - CameraDefs Class, [140](#)
- SerialPortBaudRate
 - Spinnaker::Camera, [574](#)
- SerialPortBaudRateEnums
 - CameraDefs Class, [140](#)
- SerialPortDataBits
 - Spinnaker::Camera, [575](#)
- SerialPortParity
 - Spinnaker::Camera, [575](#)
- SerialPortParityEnums
 - CameraDefs Class, [141](#)
- SerialPortSelector
 - Spinnaker::Camera, [575](#)
- SerialPortSelectorEnums
 - CameraDefs Class, [141](#)
- SerialPortSource
 - Spinnaker::Camera, [575](#)
- SerialPortSourceEnums
 - CameraDefs Class, [142](#)
- SerialPortStopBits
 - Spinnaker::Camera, [575](#)
- SerialPortStopBitsEnums
 - CameraDefs Class, [142](#)
- SerialReceiveFramingErrorCount
 - Spinnaker::Camera, [575](#)
- SerialReceiveParityErrorCount
 - Spinnaker::Camera, [576](#)
- SerialReceiveQueueClear
 - Spinnaker::Camera, [576](#)
- SerialReceiveQueueCurrentCharacterCount
 - Spinnaker::Camera, [576](#)
- SerialReceiveQueueMaxCharacterCount
 - Spinnaker::Camera, [576](#)
- SerialTransmitQueueCurrentCharacterCount
 - Spinnaker::Camera, [576](#)
- SerialTransmitQueueMaxCharacterCount
 - Spinnaker::Camera, [576](#)
- Set
 - Spinnaker::GenApi::RegisterNode, [1014](#)
- SetBufferOwnership
 - Spinnaker::CameraBase, [603](#)
 - Spinnaker::ICameraBase, [811](#)
- SetChannelStatus
 - Spinnaker::IImageStatistics, [853](#)
 - Spinnaker::ImageStatistics, [904](#)
- SetChunks
 - Spinnaker::ChunkData, [675](#)
 - Spinnaker::IChunkData, [825](#)
- SetCookie
 - IPortRecorder Interface, [312](#)
 - Spinnaker::GenApi::CPortWriteList, [718](#)
- SetDefaultColorProcessing
 - Spinnaker::Image, [892](#)
- SetEnumReference
 - Spinnaker::GenApi::CEnumerationTRef, [637](#)

- SetEventPayload
 - Spinnaker::Event, [758](#)
- SetEventType
 - Spinnaker::Event, [758](#)
- SetFirst
 - Spinnaker::GenApi::CSelectorSet, [724](#)
- SetGenICamCLProtocolFolder
 - GCUtilities Utility, [261](#)
- SetGenICamCacheFolder
 - GCUtilities Utility, [260](#)
- SetGenICamLogConfig
 - GCUtilities Utility, [261](#)
- SetHeatmapColorGradient
 - Spinnaker::ImageUtilityHeatmap, [912](#)
- SetHeatmapRange
 - Spinnaker::ImageUtilityHeatmap, [913](#)
- SetInfo
 - Spinnaker::GenApi::CFeatureBag, [656](#)
- SetIntValue
 - IEnumeration Interface, [278](#)
 - Spinnaker::GenApi::EnumNode, [750](#)
- SetLoggingEventPriorityLevel
 - Spinnaker::ISystem, [951](#)
 - Spinnaker::System, [1037](#)
- SetMaximumFileSize
 - Spinnaker::Video::SpinVideo, [1023](#)
- SetMessageCallback
 - SpinUpdate.h, [1330](#)
- SetNext
 - ISelectorDigit Interface, [317](#)
 - Spinnaker::GenApi::CSelectorSet, [724](#)
- SetNodeHandle
 - Spinnaker::GenApi::Node, [981](#)
- SetNodeMap
 - Spinnaker::GenApi::Node, [981](#)
- SetNumEnums
 - Reference Interfaces, [342](#)
 - Spinnaker::GenApi::CEnumerationTRef, [637](#)
- SetPortImpl
 - Spinnaker::GenApi::CChunkPort, [632](#)
 - Spinnaker::GenApi::CEventPort, [653](#)
 - Spinnaker::GenApi::CPortImpl, [714](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [721](#)
 - Spinnaker::GenApi::PortNode, [1002](#)
- SetProgressCallback
 - SpinUpdate.h, [1330](#)
- SetReference
 - Spinnaker::GenApi::BooleanNode, [439](#)
 - Spinnaker::GenApi::CEnumerationTRef, [637](#)
 - Spinnaker::GenApi::CategoryNode, [615](#)
 - Spinnaker::GenApi::CommandNode, [705](#)
 - Spinnaker::GenApi::EnumEntryNode, [745](#)
 - Spinnaker::GenApi::EnumNode, [750](#)
 - Spinnaker::GenApi::FloatNode, [776](#)
 - Spinnaker::GenApi::FloatRegNode, [779](#)
 - Spinnaker::GenApi::IntRegNode, [945](#)
 - Spinnaker::GenApi::IntegerNode, [929](#)
 - Spinnaker::GenApi::Node, [981](#)
 - Spinnaker::GenApi::PortNode, [1003](#)
 - Spinnaker::GenApi::PortRecorder, [1006](#)
 - Spinnaker::GenApi::PortReplay, [1009](#)
 - Spinnaker::GenApi::RegisterNode, [1014](#)
 - Spinnaker::GenApi::StringNode, [1027](#)
 - Spinnaker::GenApi::StringRegNode, [1030](#)
 - Spinnaker::GenApi::ValueNode, [1079](#)
- SetUserBuffers
 - Spinnaker::CameraBase, [603](#), [604](#)
 - Spinnaker::ICameraBase, [812](#)
- SetValue
 - Spinnaker::GenApi::BooleanNode, [439](#)
 - Spinnaker::GenApi::CEnumerationTRef, [638](#)
 - Spinnaker::GenApi::FloatNode, [776](#)
 - Spinnaker::GenApi::IntegerNode, [929](#)
 - Spinnaker::GenApi::StringNode, [1027](#)
- Sharpening
 - Spinnaker::Camera, [577](#)
- SharpeningAuto
 - Spinnaker::Camera, [577](#)
- SharpeningEnable
 - Spinnaker::Camera, [577](#)
- SharpeningThreshold
 - Spinnaker::Camera, [577](#)
- Signed
 - Types Enums, [360](#)
- SingleChunkData_t, [1017](#)
 - ChunkID, [1017](#)
 - ChunkLength, [1017](#)
 - ChunkOffset, [1017](#)
- SingleChunkDataStr_t, [1018](#)
 - ChunkID, [1018](#)
 - ChunkLength, [1018](#)
 - ChunkOffset, [1018](#)
- size
 - Spinnaker::GenApi::double_autovector_t, [735](#)
 - Spinnaker::GenApi::int64_autovector_t, [923](#)
 - Spinnaker::GenICam::gcstring, [789](#)
- SoftwareSignalPulse
 - Spinnaker::Camera, [578](#)
- SoftwareSignalSelector
 - Spinnaker::Camera, [578](#)
- SoftwareSignalSelectorEnums
 - CameraDefs Class, [142](#)
- SourceCount
 - Spinnaker::Camera, [578](#)
- SourceSelector
 - Spinnaker::Camera, [578](#)
- SourceSelectorEnums
 - CameraDefs Class, [143](#)
- SpinTestCamera, [1019](#)
- SpinTestCamera Class, [346](#)
- SpinUpdate.h
 - GetErrorMessage, [1330](#)
 - SPINUPDATE_API, [1330](#)
 - SetMessageCallback, [1330](#)
 - SetProgressCallback, [1330](#)
 - UpdateFirmware, [1330](#)

- UpdateFirmwareConsole, [1331](#)
- UpdatorMessageCallback, [1331](#)
- UpdatorProgressCallback, [1331](#)
- SpinVideo, [1019](#)
 - Spinnaker::Video::SpinVideo, [1020](#)
- Spinnaker, [365](#)
- Spinnaker Classes, [30](#)
- Spinnaker Definitions, [179](#)
 - ActionCommandStatus, [182](#)
 - BufferOwnership, [183](#)
 - ColorProcessingAlgorithm, [183](#)
 - DEPRECATED_CLASS, [189](#)
 - Error, [184](#)
 - EventType, [185](#)
 - ImageFileFormat, [185](#)
 - ImageStatus, [186](#)
 - PayloadTypeInfoIDs, [186](#)
 - PixelFormatIntType, [187](#)
 - PixelFormatNamespaceID, [187](#)
 - SpinnakerLogLevel, [188](#)
 - StatisticsChannel, [188](#)
- Spinnaker Event Classes, [27](#)
- Spinnaker GenApi Classes, [214](#)
 - _ClearXMLCache, [220](#)
 - _Connect, [220](#), [221](#)
 - _Destroy, [221](#)
 - _GetDeviceName, [221](#)
 - _GetNode, [221](#)
 - _GetNodes, [221](#)
 - _GetSupportedSchemaVersions, [221](#)
 - _InvalidateNodes, [221](#)
 - _LoadXMLFromFile, [222](#)
 - _LoadXMLFromFileInject, [222](#)
 - _LoadXMLFromString, [222](#)
 - _LoadXMLFromStringInject, [222](#)
 - _LoadXMLFromZIPData, [222](#)
 - _LoadXMLFromZIPFile, [222](#)
 - _Poll, [222](#)
 - ~CNodeMapRefT, [224](#)
 - CNodeMapRef, [220](#)
 - CNodeMapRefT, [223](#)
 - CNodeRef, [220](#)
 - CSelectorRef, [220](#)
 - CastToIDestroy, [223](#)
 - EatComments, [223](#)
 - operator<<, [223](#)
 - operator>>, [224](#)
 - operator=, [224](#)
- Spinnaker GenApi Enums, [351](#)
- Spinnaker GenApi Interfaces, [226](#)
 - CallbackHandleType, [227](#)
 - NodeList_t, [227](#)
- Spinnaker GenApi Utilities, [256](#)
- Spinnaker Headers, [176](#)
 - EVENT_TIMEOUT_INFINITE, [177](#)
 - EVENT_TIMEOUT_NONE, [177](#)
- Spinnaker Platform, [190](#)
 - SPINNAKER_API_ABSTRACT, [190](#)
 - SPINNAKER_API, [190](#)
 - SPINNAKER_LOCAL, [190](#)
- Spinnaker QuickSpin Classes, [195](#)
- Spinnaker Video Class, [191](#)
- Spinnaker Video Definitions, [192](#)
- Spinnaker.h, [178](#)
- Spinnaker::ActionCommandResult
 - DeviceAddress, [425](#)
 - Status, [425](#)
- Spinnaker::ArrivalEvent
 - ~ArrivalEvent, [427](#)
 - ArrivalEvent, [427](#)
 - OnDeviceArrival, [427](#)
 - operator=, [427](#)
- Spinnaker::BMPOption
 - BMPOption, [436](#)
 - indexedColor_8bit, [436](#)
 - reserved, [436](#)
- Spinnaker::BasePtr
 - ~BasePtr, [432](#)
 - BasePtr, [432](#)
 - get, [433](#)
 - IsValid, [433](#)
 - m_pT, [435](#)
 - operator bool, [433](#)
 - operator T*, [433](#)
 - operator->, [433](#)
 - operator=, [434](#)
 - operator==, [434](#), [435](#)
- Spinnaker::Camera
 - ~Camera, [470](#)
 - aPAUSEMACCtrlFramesReceived, [475](#)
 - aPAUSEMACCtrlFramesTransmitted, [475](#)
 - AasRoiEnable, [470](#)
 - AasRoiHeight, [470](#)
 - AasRoiOffsetX, [471](#)
 - AasRoiOffsetY, [471](#)
 - AasRoiWidth, [471](#)
 - AcquisitionAbort, [471](#)
 - AcquisitionArm, [472](#)
 - AcquisitionBurstFrameCount, [472](#)
 - AcquisitionFrameCount, [472](#)
 - AcquisitionFrameRate, [472](#)
 - AcquisitionFrameRateEnable, [472](#)
 - AcquisitionLineRate, [473](#)
 - AcquisitionMode, [473](#)
 - AcquisitionResultingFrameRate, [473](#)
 - AcquisitionStart, [473](#)
 - AcquisitionStatus, [473](#)
 - AcquisitionStatusSelector, [473](#)
 - AcquisitionStop, [474](#)
 - ActionDeviceKey, [474](#)
 - ActionGroupKey, [474](#)
 - ActionGroupMask, [474](#)
 - ActionQueueSize, [474](#)
 - ActionSelector, [474](#)
 - ActionUnconditionalMode, [475](#)
 - AdaptiveCompressionEnable, [475](#)

- AdcBitDepth, [475](#)
- AutoAlgorithmSelector, [476](#)
- AutoExposureControlLoopDamping, [476](#)
- AutoExposureControlPriority, [476](#)
- AutoExposureEVCompensation, [476](#)
- AutoExposureExposureTimeLowerLimit, [477](#)
- AutoExposureExposureTimeUpperLimit, [477](#)
- AutoExposureGainLowerLimit, [477](#)
- AutoExposureGainUpperLimit, [477](#)
- AutoExposureGreyValueLowerLimit, [478](#)
- AutoExposureGreyValueUpperLimit, [478](#)
- AutoExposureLightingMode, [478](#)
- AutoExposureMeteringMode, [478](#)
- AutoExposureTargetGreyValue, [479](#)
- AutoExposureTargetGreyValueAuto, [479](#)
- BalanceRatio, [479](#)
- BalanceRatioSelector, [480](#)
- BalanceWhiteAuto, [480](#)
- BalanceWhiteAutoDamping, [480](#)
- BalanceWhiteAutoLowerLimit, [480](#)
- BalanceWhiteAutoProfile, [481](#)
- BalanceWhiteAutoUpperLimit, [481](#)
- BinningHorizontal, [481](#)
- BinningHorizontalMode, [481](#)
- BinningSelector, [482](#)
- BinningVertical, [482](#)
- BinningVerticalMode, [482](#)
- BlackLevel, [482](#)
- BlackLevelAuto, [482](#)
- BlackLevelAutoBalance, [483](#)
- BlackLevelClampingEnable, [483](#)
- BlackLevelRaw, [483](#)
- BlackLevelSelector, [483](#)
- Camera, [470](#)
- ChunkBlackLevel, [483](#)
- ChunkBlackLevelSelector, [484](#)
- ChunkCRC, [484](#)
- ChunkCounterSelector, [484](#)
- ChunkCounterValue, [484](#)
- ChunkEnable, [484](#)
- ChunkEncoderSelector, [484](#)
- ChunkEncoderStatus, [485](#)
- ChunkEncoderValue, [485](#)
- ChunkExposureEndLineStatusAll, [485](#)
- ChunkExposureTime, [485](#)
- ChunkExposureTimeSelector, [485](#)
- ChunkFrameID, [485](#)
- ChunkGain, [486](#)
- ChunkGainSelector, [486](#)
- ChunkHeight, [486](#)
- ChunkImage, [486](#)
- ChunkImageComponent, [486](#)
- ChunkInferenceConfidence, [486](#)
- ChunkInferenceResult, [487](#)
- ChunkLinePitch, [487](#)
- ChunkLineStatusAll, [487](#)
- ChunkModeActive, [487](#)
- ChunkOffsetX, [487](#)
- ChunkOffsetY, [487](#)
- ChunkPartSelector, [488](#)
- ChunkPixelDynamicRangeMax, [488](#)
- ChunkPixelDynamicRangeMin, [488](#)
- ChunkPixelFormat, [488](#)
- ChunkRegionID, [488](#)
- ChunkScan3dAxisMax, [488](#)
- ChunkScan3dAxisMin, [489](#)
- ChunkScan3dCoordinateOffset, [489](#)
- ChunkScan3dCoordinateReferenceSelector, [489](#)
- ChunkScan3dCoordinateReferenceValue, [489](#)
- ChunkScan3dCoordinateScale, [489](#)
- ChunkScan3dCoordinateSelector, [489](#)
- ChunkScan3dCoordinateSystem, [490](#)
- ChunkScan3dCoordinateSystemReference, [490](#)
- ChunkScan3dCoordinateTransformSelector, [490](#)
- ChunkScan3dDistanceUnit, [490](#)
- ChunkScan3dInvalidDataFlag, [490](#)
- ChunkScan3dInvalidDataValue, [490](#)
- ChunkScan3dOutputMode, [491](#)
- ChunkScan3dTransformValue, [491](#)
- ChunkScanLineSelector, [491](#)
- ChunkSelector, [491](#)
- ChunkSequencerSetActive, [491](#)
- ChunkSerialData, [491](#)
- ChunkSerialDataLength, [492](#)
- ChunkSerialReceiveOverflow, [492](#)
- ChunkSourceID, [492](#)
- ChunkStreamChannelID, [492](#)
- ChunkTimerSelector, [492](#)
- ChunkTimerValue, [492](#)
- ChunkTimestamp, [493](#)
- ChunkTimestampLatchValue, [493](#)
- ChunkTransferBlockID, [493](#)
- ChunkTransferQueueCurrentBlockCount, [493](#)
- ChunkTransferStreamID, [493](#)
- ChunkWidth, [493](#)
- CIConfiguration, [494](#)
- CITimeSlotsCount, [494](#)
- ColorTransformationEnable, [494](#)
- ColorTransformationSelector, [494](#)
- ColorTransformationValue, [494](#)
- ColorTransformationValueSelector, [495](#)
- CompressionRatio, [495](#)
- CounterDelay, [495](#)
- CounterDuration, [495](#)
- CounterEventActivation, [495](#)
- CounterEventSource, [496](#)
- CounterReset, [496](#)
- CounterResetActivation, [496](#)
- CounterResetSource, [496](#)
- CounterSelector, [496](#)
- CounterStatus, [496](#)
- CounterTriggerActivation, [497](#)
- CounterTriggerSource, [497](#)
- CounterValue, [497](#)
- CounterValueAtReset, [497](#)
- CxpConnectionSelector, [497](#)

- CxpConnectionTestErrorCount, [497](#)
- CxpConnectionTestMode, [498](#)
- CxpConnectionTestPacketCount, [498](#)
- CxpLinkConfiguration, [498](#)
- CxpLinkConfigurationPreferred, [498](#)
- CxpLinkConfigurationStatus, [498](#)
- CxpPoCxpAuto, [498](#)
- CxpPoCxpStatus, [499](#)
- CxpPoCxpTripReset, [499](#)
- CxpPoCxpTurnOff, [499](#)
- DecimationHorizontal, [499](#)
- DecimationHorizontalMode, [499](#)
- DecimationSelector, [500](#)
- DecimationVertical, [500](#)
- DecimationVerticalMode, [500](#)
- DefectCorrectStaticEnable, [501](#)
- DefectCorrectionMode, [500](#)
- DefectTableApply, [501](#)
- DefectTableCoordinateX, [501](#)
- DefectTableCoordinateY, [501](#)
- DefectTableFactoryRestore, [501](#)
- DefectTableIndex, [502](#)
- DefectTablePixelCount, [502](#)
- DefectTableSave, [502](#)
- Deinterlacing, [502](#)
- DeviceCharacterSet, [502](#)
- DeviceClockFrequency, [503](#)
- DeviceClockSelector, [503](#)
- DeviceConnectionSelector, [503](#)
- DeviceConnectionSpeed, [503](#)
- DeviceConnectionStatus, [503](#)
- DeviceEventChannelCount, [503](#)
- DeviceFamilyName, [504](#)
- DeviceFeaturePersistenceEnd, [504](#)
- DeviceFeaturePersistenceStart, [504](#)
- DeviceFirmwareVersion, [504](#)
- DeviceGenCPVersionMajor, [504](#)
- DeviceGenCPVersionMinor, [504](#)
- DeviceID, [505](#)
- DeviceIndicatorMode, [505](#)
- DeviceLinkBandwidthReserve, [505](#)
- DeviceLinkCommandTimeout, [505](#)
- DeviceLinkConnectionCount, [505](#)
- DeviceLinkCurrentThroughput, [505](#)
- DeviceLinkHeartbeatMode, [506](#)
- DeviceLinkHeartbeatTimeout, [506](#)
- DeviceLinkSelector, [506](#)
- DeviceLinkSpeed, [506](#)
- DeviceLinkThroughputLimit, [506](#)
- DeviceLinkThroughputLimitMode, [507](#)
- DeviceManifestEntrySelector, [507](#)
- DeviceManifestPrimaryURL, [507](#)
- DeviceManifestSchemaMajorVersion, [507](#)
- DeviceManifestSchemaMinorVersion, [507](#)
- DeviceManifestSecondaryURL, [508](#)
- DeviceManifestXMLMajorVersion, [508](#)
- DeviceManifestXMLMinorVersion, [508](#)
- DeviceManifestXMLSubMinorVersion, [508](#)
- DeviceManufacturerInfo, [508](#)
- DeviceMaxThroughput, [508](#)
- DeviceModelName, [509](#)
- DevicePowerSupplySelector, [509](#)
- DeviceRegistersCheck, [509](#)
- DeviceRegistersEndianness, [509](#)
- DeviceRegistersStreamingEnd, [509](#)
- DeviceRegistersStreamingStart, [510](#)
- DeviceRegistersValid, [510](#)
- DeviceReset, [510](#)
- DeviceSFNCVersionMajor, [511](#)
- DeviceSFNCVersionMinor, [511](#)
- DeviceSFNCVersionSubMinor, [511](#)
- DeviceScanType, [510](#)
- DeviceSerialNumber, [510](#)
- DeviceSerialPortBaudRate, [510](#)
- DeviceSerialPortSelector, [511](#)
- DeviceStreamChannelCount, [511](#)
- DeviceStreamChannelEndianness, [511](#)
- DeviceStreamChannelLink, [512](#)
- DeviceStreamChannelPacketSize, [512](#)
- DeviceStreamChannelSelector, [512](#)
- DeviceStreamChannelType, [512](#)
- DeviceTLType, [513](#)
- DeviceTLVersionMajor, [513](#)
- DeviceTLVersionMinor, [513](#)
- DeviceTLVersionSubMinor, [513](#)
- DeviceTapGeometry, [512](#)
- DeviceTemperature, [512](#)
- DeviceTemperatureSelector, [513](#)
- DeviceType, [514](#)
- DeviceUptime, [514](#)
- DeviceUserID, [514](#)
- DeviceVendorName, [514](#)
- DeviceVersion, [514](#)
- EncoderDivider, [514](#)
- EncoderMode, [515](#)
- EncoderOutputMode, [515](#)
- EncoderReset, [515](#)
- EncoderResetActivation, [515](#)
- EncoderResetSource, [515](#)
- EncoderSelector, [515](#)
- EncoderSourceA, [516](#)
- EncoderSourceB, [516](#)
- EncoderStatus, [516](#)
- EncoderTimeout, [516](#)
- EncoderValue, [516](#)
- EncoderValueAtReset, [516](#)
- EnumerationCount, [517](#)
- EventAcquisitionEnd, [517](#)
- EventAcquisitionEndFrameID, [517](#)
- EventAcquisitionEndTimestamp, [517](#)
- EventAcquisitionError, [517](#)
- EventAcquisitionErrorFrameID, [517](#)
- EventAcquisitionErrorTimestamp, [518](#)
- EventAcquisitionStart, [518](#)
- EventAcquisitionStartFrameID, [518](#)
- EventAcquisitionStartTimestamp, [518](#)

- EventAcquisitionTransferEnd, [518](#)
- EventAcquisitionTransferEndFrameID, [518](#)
- EventAcquisitionTransferEndTimestamp, [519](#)
- EventAcquisitionTransferStart, [519](#)
- EventAcquisitionTransferStartFrameID, [519](#)
- EventAcquisitionTransferStartTimestamp, [519](#)
- EventAcquisitionTrigger, [519](#)
- EventAcquisitionTriggerFrameID, [519](#)
- EventAcquisitionTriggerTimestamp, [520](#)
- EventActionLate, [520](#)
- EventActionLateFrameID, [520](#)
- EventActionLateTimestamp, [520](#)
- EventCounter0End, [520](#)
- EventCounter0EndFrameID, [520](#)
- EventCounter0EndTimestamp, [521](#)
- EventCounter0Start, [521](#)
- EventCounter0StartFrameID, [521](#)
- EventCounter0StartTimestamp, [521](#)
- EventCounter1End, [521](#)
- EventCounter1EndFrameID, [521](#)
- EventCounter1EndTimestamp, [522](#)
- EventCounter1Start, [522](#)
- EventCounter1StartFrameID, [522](#)
- EventCounter1StartTimestamp, [522](#)
- EventEncoder0Restarted, [522](#)
- EventEncoder0RestartedFrameID, [522](#)
- EventEncoder0RestartedTimestamp, [523](#)
- EventEncoder0Stopped, [523](#)
- EventEncoder0StoppedFrameID, [523](#)
- EventEncoder0StoppedTimestamp, [523](#)
- EventEncoder1Restarted, [523](#)
- EventEncoder1RestartedFrameID, [523](#)
- EventEncoder1RestartedTimestamp, [524](#)
- EventEncoder1Stopped, [524](#)
- EventEncoder1StoppedFrameID, [524](#)
- EventEncoder1StoppedTimestamp, [524](#)
- EventError, [524](#)
- EventErrorCode, [524](#)
- EventErrorFrameID, [525](#)
- EventErrorTimestamp, [525](#)
- EventExposureEnd, [525](#)
- EventExposureEndFrameID, [525](#)
- EventExposureEndTimestamp, [525](#)
- EventExposureStart, [525](#)
- EventExposureStartFrameID, [526](#)
- EventExposureStartTimestamp, [526](#)
- EventFrameBurstEnd, [526](#)
- EventFrameBurstEndFrameID, [526](#)
- EventFrameBurstEndTimestamp, [526](#)
- EventFrameBurstStart, [526](#)
- EventFrameBurstStartFrameID, [527](#)
- EventFrameBurstStartTimestamp, [527](#)
- EventFrameEnd, [527](#)
- EventFrameEndFrameID, [527](#)
- EventFrameEndTimestamp, [527](#)
- EventFrameStart, [527](#)
- EventFrameStartFrameID, [528](#)
- EventFrameStartTimestamp, [528](#)
- EventFrameTransferEnd, [528](#)
- EventFrameTransferEndFrameID, [528](#)
- EventFrameTransferEndTimestamp, [528](#)
- EventFrameTransferStart, [528](#)
- EventFrameTransferStartFrameID, [529](#)
- EventFrameTransferStartTimestamp, [529](#)
- EventFrameTrigger, [529](#)
- EventFrameTriggerFrameID, [529](#)
- EventFrameTriggerTimestamp, [529](#)
- EventLine0AnyEdge, [529](#)
- EventLine0AnyEdgeFrameID, [530](#)
- EventLine0AnyEdgeTimestamp, [530](#)
- EventLine0FallingEdge, [530](#)
- EventLine0FallingEdgeFrameID, [530](#)
- EventLine0FallingEdgeTimestamp, [530](#)
- EventLine0RisingEdge, [530](#)
- EventLine0RisingEdgeFrameID, [531](#)
- EventLine0RisingEdgeTimestamp, [531](#)
- EventLine1AnyEdge, [531](#)
- EventLine1AnyEdgeFrameID, [531](#)
- EventLine1AnyEdgeTimestamp, [531](#)
- EventLine1FallingEdge, [531](#)
- EventLine1FallingEdgeFrameID, [532](#)
- EventLine1FallingEdgeTimestamp, [532](#)
- EventLine1RisingEdge, [532](#)
- EventLine1RisingEdgeFrameID, [532](#)
- EventLine1RisingEdgeTimestamp, [532](#)
- EventLinkSpeedChange, [532](#)
- EventLinkSpeedChangeFrameID, [533](#)
- EventLinkSpeedChangeTimestamp, [533](#)
- EventLinkTrigger0, [533](#)
- EventLinkTrigger0FrameID, [533](#)
- EventLinkTrigger0Timestamp, [533](#)
- EventLinkTrigger1, [533](#)
- EventLinkTrigger1FrameID, [534](#)
- EventLinkTrigger1Timestamp, [534](#)
- EventNotification, [534](#)
- EventSelector, [534](#)
- EventSequencerSetChange, [534](#)
- EventSequencerSetChangeFrameID, [534](#)
- EventSequencerSetChangeTimestamp, [535](#)
- EventSerialData, [535](#)
- EventSerialDataLength, [535](#)
- EventSerialPortReceive, [535](#)
- EventSerialPortReceiveTimestamp, [535](#)
- EventSerialReceiveOverflow, [535](#)
- EventStream0TransferBlockEnd, [536](#)
- EventStream0TransferBlockEndFrameID, [536](#)
- EventStream0TransferBlockEndTimestamp, [536](#)
- EventStream0TransferBlockStart, [536](#)
- EventStream0TransferBlockStartFrameID, [536](#)
- EventStream0TransferBlockStartTimestamp, [536](#)
- EventStream0TransferBlockTrigger, [537](#)
- EventStream0TransferBlockTriggerFrameID, [537](#)
- EventStream0TransferBlockTriggerTimestamp, [537](#)
- EventStream0TransferBurstEnd, [537](#)
- EventStream0TransferBurstEndFrameID, [537](#)
- EventStream0TransferBurstEndTimestamp, [537](#)

- EventStream0TransferBurstStart, [538](#)
- EventStream0TransferBurstStartFrameID, [538](#)
- EventStream0TransferBurstStartTimestamp, [538](#)
- EventStream0TransferEnd, [538](#)
- EventStream0TransferEndFrameID, [538](#)
- EventStream0TransferEndTimestamp, [538](#)
- EventStream0TransferOverflow, [539](#)
- EventStream0TransferOverflowFrameID, [539](#)
- EventStream0TransferOverflowTimestamp, [539](#)
- EventStream0TransferPause, [539](#)
- EventStream0TransferPauseFrameID, [539](#)
- EventStream0TransferPauseTimestamp, [539](#)
- EventStream0TransferResume, [540](#)
- EventStream0TransferResumeFrameID, [540](#)
- EventStream0TransferResumeTimestamp, [540](#)
- EventStream0TransferStart, [540](#)
- EventStream0TransferStartFrameID, [540](#)
- EventStream0TransferStartTimestamp, [540](#)
- EventTest, [541](#)
- EventTestTimestamp, [541](#)
- EventTimer0End, [541](#)
- EventTimer0EndFrameID, [541](#)
- EventTimer0EndTimestamp, [541](#)
- EventTimer0Start, [541](#)
- EventTimer0StartFrameID, [542](#)
- EventTimer0StartTimestamp, [542](#)
- EventTimer1End, [542](#)
- EventTimer1EndFrameID, [542](#)
- EventTimer1EndTimestamp, [542](#)
- EventTimer1Start, [542](#)
- EventTimer1StartFrameID, [543](#)
- EventTimer1StartTimestamp, [543](#)
- ExposureActiveMode, [543](#)
- ExposureAuto, [543](#)
- ExposureMode, [543](#)
- ExposureTime, [543](#)
- ExposureTimeMode, [544](#)
- ExposureTimeSelector, [544](#)
- FactoryReset, [544](#)
- FileAccessBuffer, [544](#)
- FileAccessLength, [544](#)
- FileAccessOffset, [544](#)
- FileOpenMode, [545](#)
- FileOperationExecute, [545](#)
- FileOperationResult, [545](#)
- FileOperationSelector, [545](#)
- FileOperationStatus, [545](#)
- FileSelector, [546](#)
- FileSize, [546](#)
- Gain, [546](#)
- GainAuto, [546](#)
- GainAutoBalance, [546](#)
- GainSelector, [547](#)
- Gamma, [547](#)
- GammaEnable, [547](#)
- GevActiveLinkCount, [547](#)
- GevCCP, [547](#)
- GevCurrentDefaultGateway, [547](#)
- GevCurrentIPAddress, [548](#)
- GevCurrentIPConfigurationDHCP, [548](#)
- GevCurrentIPConfigurationLLA, [548](#)
- GevCurrentIPConfigurationPersistentIP, [548](#)
- GevCurrentPhysicalLinkConfiguration, [548](#)
- GevCurrentSubnetMask, [548](#)
- GevDiscoveryAckDelay, [549](#)
- GevFirstURL, [549](#)
- GevGVCPExtendedStatusCodes, [549](#)
- GevGVCPExtendedStatusCodesSelector, [549](#)
- GevGVCPHeartbeatDisable, [549](#)
- GevGVCPPendingAck, [549](#)
- GevGVCPPendingTimeout, [550](#)
- GevGVSPExtendedIDMode, [550](#)
- GevHeartbeatTimeout, [550](#)
- GevIEEE1588, [550](#)
- GevIEEE1588ClockAccuracy, [550](#)
- GevIEEE1588Mode, [550](#)
- GevIEEE1588Status, [551](#)
- GevIPConfigurationStatus, [551](#)
- GevInterfaceSelector, [551](#)
- GevMACAddress, [551](#)
- GevMCDA, [551](#)
- GevMCPHostPort, [551](#)
- GevMCRC, [552](#)
- GevMCSP, [552](#)
- GevMCTT, [552](#)
- GevNumberOfInterfaces, [552](#)
- GevPAUSEFrameReception, [552](#)
- GevPAUSEFrameTransmission, [552](#)
- GevPersistentDefaultGateway, [553](#)
- GevPersistentIPAddress, [553](#)
- GevPersistentSubnetMask, [553](#)
- GevPhysicalLinkConfiguration, [553](#)
- GevPrimaryApplicationIPAddress, [553](#)
- GevPrimaryApplicationSocket, [553](#)
- GevPrimaryApplicationSwitchoverKey, [554](#)
- GevSCCFGAllInTransmission, [554](#)
- GevSCCFGExtendedChunkData, [554](#)
- GevSCCFGPacketResendDestination, [554](#)
- GevSCCFGUnconditionalStreaming, [554](#)
- GevSCDA, [554](#)
- GevSCPDDirection, [555](#)
- GevSCPHostPort, [555](#)
- GevSCPInterfaceIndex, [555](#)
- GevSCPSBigEndian, [555](#)
- GevSCPSDoNotFragment, [555](#)
- GevSCPSFireTestPacket, [556](#)
- GevSCPSPacketSize, [556](#)
- GevSCPD, [556](#)
- GevSCSP, [556](#)
- GevSCZoneConfigurationLock, [556](#)
- GevSCZoneCount, [556](#)
- GevSCZoneDirectionAll, [556](#)
- GevSecondURL, [557](#)
- GevStreamChannelSelector, [557](#)
- GevSupportedOption, [557](#)
- GevSupportedOptionSelector, [557](#)

- GevTimestampTickFrequency, [557](#)
- GuiXmlManifestAddress, [557](#)
- Height, [558](#)
- HeightMax, [558](#)
- ImageComponentEnable, [558](#)
- ImageComponentSelector, [558](#)
- ImageCompressionBitrate, [558](#)
- ImageCompressionJPEGFormatOption, [558](#)
- ImageCompressionMode, [559](#)
- ImageCompressionQuality, [559](#)
- ImageCompressionRateOption, [559](#)
- Init, [470](#)
- IspEnable, [559](#)
- LUTEnable, [563](#)
- LUTIndex, [563](#)
- LUTSelector, [563](#)
- LUTValue, [563](#)
- LUTValueAll, [564](#)
- LineFilterWidth, [559](#)
- LineFormat, [560](#)
- LineInputFilterSelector, [560](#)
- LineInverter, [560](#)
- LineMode, [560](#)
- LinePitch, [560](#)
- LineSelector, [560](#)
- LineSource, [561](#)
- LineStatus, [561](#)
- LineStatusAll, [561](#)
- LinkErrorCount, [561](#)
- LinkUptime, [561](#)
- LogicBlockLUTInputActivation, [561](#)
- LogicBlockLUTInputSelector, [562](#)
- LogicBlockLUTInputSource, [562](#)
- LogicBlockLUTOutputValue, [562](#)
- LogicBlockLUTOutputValueAll, [562](#)
- LogicBlockLUTRowIndex, [562](#)
- LogicBlockLUTSelector, [562](#)
- LogicBlockSelector, [563](#)
- MaxDeviceResetTime, [564](#)
- OffsetX, [564](#)
- OffsetY, [564](#)
- PacketResendRequestCount, [564](#)
- PayloadSize, [565](#)
- PixelColorFilter, [565](#)
- PixelDynamicRangeMax, [565](#)
- PixelDynamicRangeMin, [565](#)
- PixelFormat, [565](#)
- PixelFormatInfoID, [566](#)
- PixelFormatInfoSelector, [566](#)
- PixelSize, [566](#)
- PowerSupplyCurrent, [566](#)
- PowerSupplyVoltage, [566](#)
- RegionDestination, [566](#)
- RegionMode, [567](#)
- RegionSelector, [567](#)
- ReverseX, [567](#)
- ReverseY, [567](#)
- RgbTransformLightSource, [567](#)
- Saturation, [568](#)
- SaturationEnable, [568](#)
- Scan3dAxisMax, [568](#)
- Scan3dAxisMin, [568](#)
- Scan3dCoordinateOffset, [568](#)
- Scan3dCoordinateReferenceSelector, [569](#)
- Scan3dCoordinateReferenceValue, [569](#)
- Scan3dCoordinateScale, [569](#)
- Scan3dCoordinateSelector, [569](#)
- Scan3dCoordinateSystem, [569](#)
- Scan3dCoordinateSystemReference, [569](#)
- Scan3dCoordinateTransformSelector, [570](#)
- Scan3dDistanceUnit, [570](#)
- Scan3dInvalidDataFlag, [570](#)
- Scan3dInvalidDataValue, [570](#)
- Scan3dOutputMode, [570](#)
- Scan3dTransformValue, [570](#)
- SensorDescription, [571](#)
- SensorDigitizationTaps, [571](#)
- SensorHeight, [571](#)
- SensorShutterMode, [571](#)
- SensorTaps, [571](#)
- SensorWidth, [571](#)
- SequencerConfigurationMode, [572](#)
- SequencerConfigurationValid, [572](#)
- SequencerFeatureEnable, [572](#)
- SequencerMode, [572](#)
- SequencerPathSelector, [572](#)
- SequencerSetActive, [573](#)
- SequencerSetLoad, [573](#)
- SequencerSetNext, [573](#)
- SequencerSetSave, [573](#)
- SequencerSetSelector, [573](#)
- SequencerSetStart, [574](#)
- SequencerSetValid, [574](#)
- SequencerTriggerActivation, [574](#)
- SequencerTriggerSource, [574](#)
- SerialPortBaudRate, [574](#)
- SerialPortDataBits, [575](#)
- SerialPortParity, [575](#)
- SerialPortSelector, [575](#)
- SerialPortSource, [575](#)
- SerialPortStopBits, [575](#)
- SerialReceiveFramingErrorCount, [575](#)
- SerialReceiveParityErrorCount, [576](#)
- SerialReceiveQueueClear, [576](#)
- SerialReceiveQueueCurrentCharacterCount, [576](#)
- SerialReceiveQueueMaxCharacterCount, [576](#)
- SerialTransmitQueueCurrentCharacterCount, [576](#)
- SerialTransmitQueueMaxCharacterCount, [576](#)
- Sharpening, [577](#)
- SharpeningAuto, [577](#)
- SharpeningEnable, [577](#)
- SharpeningThreshold, [577](#)
- SoftwareSignalPulse, [578](#)
- SoftwareSignalSelector, [578](#)
- SourceCount, [578](#)
- SourceSelector, [578](#)

- TLParamsLocked, [581](#)
- Test0001, [578](#)
- TestEventGenerate, [579](#)
- TestPattern, [579](#)
- TestPatternGeneratorSelector, [579](#)
- TestPendingAck, [579](#)
- TimerDelay, [579](#)
- TimerDuration, [580](#)
- TimerReset, [580](#)
- TimerSelector, [580](#)
- TimerStatus, [580](#)
- TimerTriggerActivation, [580](#)
- TimerTriggerSource, [580](#)
- TimerValue, [581](#)
- Timestamp, [581](#)
- TimestampLatch, [581](#)
- TimestampLatchValue, [581](#)
- TimestampReset, [581](#)
- TransferAbort, [582](#)
- TransferBlockCount, [582](#)
- TransferBurstCount, [582](#)
- TransferComponentSelector, [582](#)
- TransferControlMode, [582](#)
- TransferOperationMode, [582](#)
- TransferPause, [583](#)
- TransferQueueCurrentBlockCount, [583](#)
- TransferQueueMaxBlockCount, [583](#)
- TransferQueueMode, [583](#)
- TransferQueueOverflowCount, [583](#)
- TransferResume, [583](#)
- TransferSelector, [584](#)
- TransferStart, [584](#)
- TransferStatus, [584](#)
- TransferStatusSelector, [584](#)
- TransferStop, [584](#)
- TransferStreamChannel, [584](#)
- TransferTriggerActivation, [585](#)
- TransferTriggerMode, [585](#)
- TransferTriggerSelector, [585](#)
- TransferTriggerSource, [585](#)
- TriggerActivation, [585](#)
- TriggerDelay, [585](#)
- TriggerDivider, [586](#)
- TriggerEventTest, [586](#)
- TriggerMode, [586](#)
- TriggerMultiplier, [586](#)
- TriggerOverlap, [586](#)
- TriggerSelector, [587](#)
- TriggerSoftware, [587](#)
- TriggerSource, [587](#)
- UserOutputSelector, [587](#)
- UserOutputValue, [587](#)
- UserOutputValueAll, [588](#)
- UserOutputValueAllMask, [588](#)
- UserSetDefault, [588](#)
- UserSetFeatureEnable, [588](#)
- UserSetLoad, [588](#)
- UserSetSave, [589](#)
- UserSetSelector, [589](#)
- V3_3Enable, [589](#)
- WhiteClip, [589](#)
- WhiteClipSelector, [589](#)
- Width, [590](#)
- WidthMax, [590](#)
- Spinnaker::CameraBase
 - ~CameraBase, [593](#)
 - BeginAcquisition, [594](#)
 - CameraBase, [593](#), [594](#)
 - DeInit, [594](#)
 - DiscoverMaxPacketSize, [594](#)
 - EndAcquisition, [595](#)
 - ForceIP, [595](#)
 - GetAccessMode, [595](#)
 - GetBufferOwnership, [596](#)
 - GetGuiXml, [596](#)
 - GetNextImage, [596](#)
 - GetNodeMap, [597](#)
 - GetNumDataStreams, [597](#)
 - GetNumImagesInUse, [598](#)
 - GetTLDeviceNodeMap, [598](#)
 - GetTLStreamNodeMap, [598](#)
 - GetUniqueID, [599](#)
 - GetUserBufferCount, [599](#)
 - GetUserBufferSize, [599](#)
 - GetUserBufferTotalSize, [600](#)
 - Init, [600](#)
 - InterfaceImpl, [605](#)
 - IsInitialized, [601](#)
 - IsStreaming, [601](#)
 - IsValid, [601](#)
 - operator=, [602](#)
 - ReadPort, [602](#)
 - RegisterEvent, [602](#), [603](#)
 - SetBufferOwnership, [603](#)
 - SetUserBuffers, [603](#), [604](#)
 - UnregisterEvent, [605](#)
 - WritePort, [605](#)
- Spinnaker::CameraList
 - ~CameraList, [607](#)
 - Append, [608](#)
 - CameraList, [607](#)
 - Clear, [609](#)
 - GetByIndex, [609](#)
 - GetBySerial, [609](#)
 - GetSize, [610](#)
 - operator=, [610](#)
 - operator[], [610](#)
 - RemoveByIndex, [611](#)
 - RemoveBySerial, [611](#)
- Spinnaker::ChunkData
 - ~ChunkData, [667](#)
 - ChunkData, [666](#)
 - GetBlackLevel, [667](#)
 - GetCRC, [667](#)
 - GetCounterValue, [667](#)
 - GetEncoderValue, [667](#)

- GetExposureEndLineStatusAll, 668
- GetExposureTime, 668
- GetFrameID, 668
- GetGain, 668
- GetHeight, 669
- GetImage, 669
- GetInferenceConfidence, 669
- GetInferenceResult, 669
- GetLinePitch, 669
- GetLineStatusAll, 670
- GetOffsetX, 670
- GetOffsetY, 670
- GetPartSelector, 670
- GetPixelDynamicRangeMax, 671
- GetPixelDynamicRangeMin, 671
- GetScan3dAxisMax, 671
- GetScan3dAxisMin, 671
- GetScan3dCoordinateOffset, 672
- GetScan3dCoordinateReferenceValue, 672
- GetScan3dCoordinateScale, 672
- GetScan3dInvalidDataValue, 672
- GetScan3dTransformValue, 673
- GetScanLineSelector, 673
- GetSequencerSetActive, 673
- GetSerialDataLength, 673
- GetStreamChannelID, 674
- GetTimerValue, 674
- GetTimestamp, 674
- GetTimestampLatchValue, 674
- GetTransferBlockID, 675
- GetTransferQueueCurrentBlockCount, 675
- GetWidth, 675
- SetChunks, 675
- Spinnaker::DeviceEvent
 - ~DeviceEvent, 732
 - DeviceEvent, 731
 - GetDeviceEventId, 732
 - GetDeviceEventName, 732
 - OnDeviceEvent, 732
 - operator=, 733
- Spinnaker::Event
 - ~Event, 757
 - Event, 757
 - EventProcessor, 758
 - GetEventPayloadData, 757
 - GetEventPayloadDataSize, 757
 - GetEventType, 757
 - IDataStream, 758
 - m_pEventData, 759
 - operator=, 758
 - SetEventPayload, 758
 - SetEventType, 758
 - Stream, 759
- Spinnaker::Exception
 - ~Exception, 762
 - Exception, 762
 - GetBuildDate, 763
 - GetBuildTime, 763
 - GetError, 763
 - GetErrorMessage, 763
 - GetFileName, 763
 - GetFullErrorMessage, 763
 - GetFunctionName, 763
 - GetLineNumber, 764
 - operator!=, 764
 - operator=, 764
 - operator==, 764
 - what, 764
- Spinnaker::GenApi, 404
 - COMMAND_MAGIC, 420
 - GENCP_COMMAND_HEADER_SIZE, 420
 - GENCP_EVENT_BASIC_SIZE, 420
 - GENCP_EVENT_CMD_ID, 420
 - GVCP_MESSAGE_TAGS, 419
 - IDevFileStream, 418
 - IPersistScript, 420
 - ODevFileStream, 418
 - PersistFeature, 419
 - SET_GUID, 419
 - U3V_EVENT_PREFIX, 420
- Spinnaker::GenApi::AutoLock
 - ~AutoLock, 429
 - AutoLock, 429
- Spinnaker::GenApi::BooleanNode
 - ~BooleanNode, 438
 - BooleanNode, 438
 - GetValue, 438
 - operator=, 439
 - SetReference, 439
 - SetValue, 439
- Spinnaker::GenApi::CChunkAdapter
 - ~CChunkAdapter, 616
 - AttachBuffer, 617
 - AttachNodeMap, 617
 - CChunkAdapter, 616
 - CheckBufferLayout, 617
 - ClearCaches, 617
 - DetachBuffer, 618
 - DetachNodeMap, 618
 - m_pChunkAdapter, 618
 - UpdateBuffer, 618
- Spinnaker::GenApi::CChunkAdapterDcam
 - ~CChunkAdapterDcam, 620
 - AttachBuffer, 620
 - CChunkAdapterDcam, 620
 - CheckBufferLayout, 620
 - CheckCRC, 621
 - HasCRC, 621
- Spinnaker::GenApi::CChunkAdapterGEV
 - ~CChunkAdapterGEV, 625
 - AttachBuffer, 625
 - CChunkAdapterGEV, 625
 - CheckBufferLayout, 625
- Spinnaker::GenApi::CChunkAdapterGeneric
 - ~CChunkAdapterGeneric, 622
 - AttachBuffer, 623

- CChunkAdapterGeneric, [622](#)
- CheckBufferLayout, [623](#)
- Spinnaker::GenApi::CChunkAdapterU3V
 - ~CChunkAdapterU3V, [627](#)
 - AttachBuffer, [627](#)
 - CChunkAdapterU3V, [627](#)
 - CheckBufferLayout, [627](#)
- Spinnaker::GenApi::CChunkPort
 - ~CChunkPort, [629](#)
 - AttachChunk, [630](#)
 - AttachPort, [630](#)
 - CChunkPort, [629](#)
 - CheckChunkID, [630](#)
 - ClearCache, [630](#)
 - DetachChunk, [631](#)
 - DetachPort, [631](#)
 - GetAccessMode, [631](#)
 - GetChunkIDLength, [631](#)
 - GetPrincipalInterfaceType, [631](#)
 - GetSwapEndianness, [631](#)
 - InvalidateNode, [632](#)
 - m_pChunkPort, [633](#)
 - m_pPort, [633](#)
 - m_pPortAdapter, [633](#)
 - Read, [632](#)
 - SetPortImpl, [632](#)
 - UpdateBuffer, [632](#)
 - Write, [632](#)
- Spinnaker::GenApi::CEnumerationTRef
 - ~CEnumerationTRef, [635](#)
 - CEnumerationTRef, [635](#)
 - GetCurrentEntry, [635](#)
 - GetEntry, [636](#)
 - GetValue, [636](#)
 - operator(), [636](#)
 - operator=, [637](#)
 - SetEnumReference, [637](#)
 - SetNumEnums, [637](#)
 - SetReference, [637](#)
 - SetValue, [638](#)
- Spinnaker::GenApi::CEventAdapter
 - ~CEventAdapter, [639](#)
 - AttachNodeMap, [639](#)
 - CEventAdapter, [639](#)
 - DeliverMessage, [639](#)
 - DetachNodeMap, [640](#)
 - m_pEventAdapter, [640](#)
- Spinnaker::GenApi::CEventAdapter1394
 - ~CEventAdapter1394, [641](#)
 - CEventAdapter1394, [641](#)
 - DeliverEventMessage, [642](#)
 - DeliverMessage, [642](#)
- Spinnaker::GenApi::CEventAdapterGEV
 - ~CEventAdapterGEV, [646](#)
 - CEventAdapterGEV, [646](#)
 - DeliverEventMessage, [646](#)
 - DeliverMessage, [646](#)
- Spinnaker::GenApi::CEventAdapterGeneric
 - ~CEventAdapterGeneric, [643](#)
 - CEventAdapterGeneric, [643](#)
 - DeliverMessage, [644](#)
- Spinnaker::GenApi::CEventAdapterU3V
 - ~CEventAdapterU3V, [648](#)
 - CEventAdapterU3V, [648](#)
 - DeliverEventMessage, [648](#)
 - DeliverMessage, [648](#)
- Spinnaker::GenApi::CEventPort
 - ~CEventPort, [651](#)
 - AttachEvent, [651](#)
 - AttachNode, [651](#)
 - CEventPort, [650](#)
 - CheckEventID, [651](#)
 - DetachEvent, [652](#)
 - DetachNode, [652](#)
 - GetAccessMode, [652](#)
 - GetEventIDLength, [652](#)
 - GetPrincipalInterfaceType, [652](#)
 - GetSwapEndianness, [652](#)
 - InvalidateNode, [653](#)
 - m_pEventPort, [653](#)
 - m_pNode, [654](#)
 - m_pPortAdapter, [654](#)
 - Read, [653](#)
 - SetPortImpl, [653](#)
 - Write, [653](#)
- Spinnaker::GenApi::CFeatureBag
 - ~CFeatureBag, [655](#)
 - CFeatureBag, [655](#)
 - GetFeatureBagHandle, [655](#)
 - LoadFromBag, [655](#)
 - operator==, [656](#)
 - PersistFeature, [656](#)
 - SetInfo, [656](#)
 - StoreToBag, [656](#)
- Spinnaker::GenApi::CFloatPtr
 - CFloatPtr, [658](#)
 - GetEnumAlias, [658](#)
 - GetIntAlias, [658](#)
 - operator=, [658](#)
- Spinnaker::GenApi::CGeneric_XMLLoaderParams
 - _Initialize, [659](#)
- Spinnaker::GenApi::CLock
 - ~CLock, [677](#)
 - CLock, [677](#)
 - Lock, [677](#)
 - m_bOwnLock, [678](#)
 - m_lock, [678](#)
 - NodeMap, [678](#)
 - TryLock, [677](#)
 - Unlock, [678](#)
- Spinnaker::GenApi::CLockEx
 - m_lockEx, [682](#)
- Spinnaker::GenApi::CNodeCallback
 - ~CNodeCallback, [684](#)
 - CNodeCallback, [684](#)
 - Destroy, [684](#)

- GetCallbackType, 684
- GetNode, 685
- m_CallbackType, 685
- m_pNode, 685
- operator(), 685
- Spinnaker::GenApi::CNodeMapFactory
 - ~CNodeMapFactory, 688
 - AddInjectionData, 690
 - ApplyStyleSheet, 690
 - CNodeMapFactory, 688, 689
 - ClearCache, 690
 - CreateEmptyNodeMap, 691
 - CreateNodeDataFromNodeMap, 691
 - CreateNodeMap, 691
 - ExtractSubtree, 691
 - GetNodeStatistics, 692
 - GetSupportedSchemaVersions, 692
 - IsCameraDescriptionFileDataReleased, 692
 - IsEmpty, 692
 - IsLoaded, 692
 - IsPreprocessed, 693
 - LoadAndInject, 693
 - operator=, 693
 - Preprocess, 693
 - ReleaseCameraDescriptionFileData, 693
 - ToString, 694
 - ToXml, 694
- Spinnaker::GenApi::CNodeMapFactory::NodeStatistics↔
 - _t
 - NumLinks, 991
 - NumNodes, 991
 - NumProperties, 991
 - NumStrings, 991
- Spinnaker::GenApi::CNodeMapRef
 - CNodeMapRef, 695, 696
 - operator=, 696
- Spinnaker::GenApi::CNodeMapRefT
 - _ClearXMLCache, 699
 - _Connect, 699
 - _GetDeviceName, 699
 - _GetNode, 699
 - _GetNodes, 700
 - _GetSupportedSchemaVersions, 700
 - _InvalidateNodes, 700
 - _LoadXMLFromFile, 700
 - _LoadXMLFromFileInject, 700
 - _LoadXMLFromString, 701
 - _LoadXMLFromStringInject, 701
 - _LoadXMLFromZIPData, 701
 - _LoadXMLFromZIPFile, 701
 - _Poll, 701
 - _Ptr, 702
- Spinnaker::GenApi::CPointer
 - ~CPointer, 709
 - CPointer, 708
 - IsValid, 709
 - m_pT, 712
 - operator bool, 709
 - operator T*, 709
 - operator!=, 709, 710
 - operator*, 710
 - operator(), 710
 - operator->, 711
 - operator=, 711
 - operator==, 711
- Spinnaker::GenApi::CPortImpl
 - ~CPortImpl, 713
 - CPortImpl, 713
 - GetAccessMode, 714
 - GetSwapEndianness, 714
 - InvalidateNode, 714
 - m_ptrPort, 715
 - Read, 714
 - Replay, 714
 - SetPortImpl, 714
 - Write, 715
- Spinnaker::GenApi::CPortWriteList
 - ~CPortWriteList, 717
 - CPortWriteList, 717
 - GetCookie, 717
 - GetPortWriteListHandle, 717
 - m_pWriteList, 718
 - Replay, 717
 - SetCookie, 718
 - Write, 718
- Spinnaker::GenApi::CRegisterPortImpl
 - ~CRegisterPortImpl, 720
 - CRegisterPortImpl, 720
 - GetAccessMode, 720
 - Read, 721
 - ReadRegister, 721
 - SetPortImpl, 721
 - Write, 721
 - WriteRegister, 722
- Spinnaker::GenApi::CSelectorSet
 - ~CSelectorSet, 724
 - CSelectorSet, 723
 - GetSelectorList, 724
 - IsEmpty, 724
 - Restore, 724
 - SetFirst, 724
 - SetNext, 724
 - ToString, 725
- Spinnaker::GenApi::CTestPortStruct
 - CTestPortStruct, 727
 - GetAccessMode, 727
 - GetNumReads, 727
 - GetNumWrites, 727
 - GetPrincipalInterfaceType, 727
 - m_BaseAddress, 728
 - m_NumReads, 728
 - m_NumWrites, 729
 - MemSet, 727
 - Read, 728
 - ResetStatistics, 728
 - Write, 728

- Spinnaker::GenApi::CategoryNode
 - ~CategoryNode, [615](#)
 - CategoryNode, [614](#)
 - GetFeatures, [615](#)
 - SetReference, [615](#)
- Spinnaker::GenApi::CommandNode
 - ~CommandNode, [704](#)
 - CommandNode, [703](#), [704](#)
 - Execute, [704](#)
 - IsDone, [704](#)
 - operator(), [705](#)
 - SetReference, [705](#)
- Spinnaker::GenApi::Counter
 - Counter, [706](#)
 - GetValue, [706](#)
 - IsZero, [706](#)
 - operator unsigned int, [706](#)
 - operator++, [706](#)
 - operator--, [706](#), [707](#)
- Spinnaker::GenApi::EAccessModeClass
 - FromString, [736](#)
 - ToString, [736](#)
- Spinnaker::GenApi::ECachingModeClass
 - FromString, [737](#)
 - ToString, [737](#)
- Spinnaker::GenApi::EDisplayNotationClass
 - FromString, [738](#)
 - ToString, [738](#)
- Spinnaker::GenApi::EEndianessClass
 - FromString, [739](#)
 - ToString, [739](#)
- Spinnaker::GenApi::EGenApiSchemaVersionClass
 - FromString, [740](#)
 - ToString, [740](#)
- Spinnaker::GenApi::EInputDirectionClass
 - FromString, [741](#)
 - ToString, [741](#)
- Spinnaker::GenApi::ENameSpaceClass
 - FromString, [742](#)
 - ToString, [742](#)
- Spinnaker::GenApi::ERepresentationClass
 - FromString, [751](#)
 - ToString, [752](#)
- Spinnaker::GenApi::ESignClass
 - FromString, [753](#)
 - ToString, [753](#)
- Spinnaker::GenApi::ESlopeClass
 - FromString, [754](#)
 - ToString, [754](#)
- Spinnaker::GenApi::EStandardNameSpaceClass
 - FromString, [755](#)
 - ToString, [755](#)
- Spinnaker::GenApi::EVisibilityClass
 - FromString, [759](#)
 - ToString, [760](#)
- Spinnaker::GenApi::EYesNoClass
 - FromString, [765](#)
 - ToString, [765](#)
- Spinnaker::GenApi::EnumEntryNode
 - ~EnumEntryNode, [744](#)
 - EnumEntryNode, [744](#)
 - GetNumericValue, [744](#)
 - GetSymbolic, [745](#)
 - GetValue, [745](#)
 - IsSelfClearing, [745](#)
 - SetReference, [745](#)
- Spinnaker::GenApi::EnumNode
 - ~EnumNode, [748](#)
 - EnumNode, [748](#)
 - GetCurrentEntry, [748](#)
 - GetEntries, [748](#)
 - GetEntry, [749](#)
 - GetEntryByName, [749](#)
 - GetIntValue, [749](#)
 - GetSymbolics, [749](#)
 - m_pEnumeration, [751](#)
 - operator*, [750](#)
 - operator=, [750](#)
 - SetIntValue, [750](#)
 - SetReference, [750](#)
- Spinnaker::GenApi::FileProtocolAdapter
 - ~FileProtocolAdapter, [766](#)
 - attach, [767](#)
 - closeFile, [768](#)
 - deleteFile, [768](#)
 - FileProtocolAdapter, [766](#)
 - getBufSize, [768](#)
 - openFile, [769](#)
 - read, [769](#)
 - write, [770](#)
- Spinnaker::GenApi::FloatNode
 - ~FloatNode, [773](#)
 - FloatNode, [773](#)
 - GetDisplayNotation, [773](#)
 - GetDisplayPrecision, [773](#)
 - GetEnumAlias, [773](#)
 - GetInc, [773](#)
 - GetIncMode, [774](#)
 - GetIntAlias, [774](#)
 - GetListOfValidValues, [774](#)
 - GetMax, [774](#)
 - GetMin, [774](#)
 - GetRepresentation, [774](#)
 - GetUnit, [775](#)
 - GetValue, [775](#)
 - HasInc, [775](#)
 - ImposeMax, [775](#)
 - ImposeMin, [775](#)
 - operator*, [776](#)
 - operator(), [776](#)
 - operator=, [776](#)
 - SetReference, [776](#)
 - SetValue, [776](#)
- Spinnaker::GenApi::FloatRegNode
 - ~FloatRegNode, [779](#)
 - FloatRegNode, [778](#), [779](#)

- SetReference, [779](#)
- Spinnaker::GenApi::Function_NodeCallback
 - Destroy, [781](#)
 - Function_NodeCallback, [781](#)
 - operator(), [781](#)
- Spinnaker::GenApi::IDevFileStreamBase
 - close, [831](#)
 - filebuf_type, [831](#)
 - ios_type, [831](#)
 - is_open, [831](#)
 - istream_type, [831](#)
 - open, [831](#)
 - rdbuf, [832](#)
- Spinnaker::GenApi::IDevFileStreamBuf
 - ~IDevFileStreamBuf, [833](#)
 - close, [833](#)
 - IDevFileStreamBuf, [833](#)
 - is_open, [833](#)
 - open, [833](#)
 - pbackfail, [834](#)
 - underflow, [834](#)
- Spinnaker::GenApi::IntRegNode
 - ~IntRegNode, [945](#)
 - IntRegNode, [944](#), [945](#)
 - SetReference, [945](#)
- Spinnaker::GenApi::IntegerNode
 - ~IntegerNode, [926](#)
 - GetFloatAlias, [926](#)
 - GetInc, [927](#)
 - GetIncMode, [927](#)
 - GetListOfValidValues, [927](#)
 - GetMax, [927](#)
 - GetMin, [927](#)
 - GetRepresentation, [927](#)
 - GetUnit, [928](#)
 - GetValue, [928](#)
 - ImposeMax, [928](#)
 - ImposeMin, [928](#)
 - IntegerNode, [926](#)
 - operator*, [929](#)
 - operator(), [928](#)
 - operator=, [929](#)
 - SetReference, [929](#)
 - SetValue, [929](#)
- Spinnaker::GenApi::Member_NodeCallback
 - Destroy, [969](#)
 - Member_NodeCallback, [969](#)
 - operator(), [969](#)
 - PMEMBERFUNC, [969](#)
- Spinnaker::GenApi::Node
 - ~Node, [974](#)
 - DeregisterCallback, [974](#)
 - GetAccessMode, [975](#)
 - GetAlias, [975](#)
 - GetCachingMode, [975](#)
 - GetCastAlias, [975](#)
 - GetChildren, [975](#)
 - GetDescription, [976](#)
 - GetDeviceName, [976](#)
 - GetDisplayName, [976](#)
 - GetDocuURL, [976](#)
 - GetEventID, [976](#)
 - GetName, [976](#)
 - GetNameSpace, [977](#)
 - GetNodeHandle, [977](#)
 - GetNodeMap, [977](#)
 - GetParents, [977](#)
 - GetPollingTime, [977](#)
 - GetPrincipalInterfaceType, [978](#)
 - GetProperty, [978](#)
 - GetPropertyNames, [978](#)
 - GetSelectedFeatures, [978](#)
 - GetSelectingFeatures, [978](#)
 - GetToolTip, [979](#)
 - GetVisibility, [979](#)
 - ImposeAccessMode, [979](#)
 - ImposeVisibility, [979](#)
 - InvalidateNode, [979](#)
 - IsAccessModeCacheable, [979](#)
 - IsCachable, [980](#)
 - IsDeprecated, [980](#)
 - IsFeature, [980](#)
 - IsSelector, [980](#)
 - IsStreamable, [980](#)
 - m_Callbacks, [982](#)
 - m_pNodeData, [982](#)
 - m_pNodeMap, [982](#)
 - Node, [974](#)
 - operator!=, [980](#)
 - operator==, [981](#)
 - RegisterCallback, [981](#)
 - SetNodeHandle, [981](#)
 - SetNodeMap, [981](#)
 - SetReference, [981](#)
- Spinnaker::GenApi::NodeMap
 - _Ptr, [990](#)
 - ~NodeMap, [985](#)
 - ClearXMLCache, [985](#)
 - Connect, [985](#)
 - Destroy, [985](#)
 - GetDeviceName, [986](#)
 - GetDeviceVersion, [986](#)
 - GetGenApiVersion, [986](#)
 - GetLock, [986](#)
 - GetModelName, [986](#)
 - GetNode, [986](#)
 - GetNodeMapHandle, [987](#)
 - GetNodes, [987](#)
 - GetNumNodes, [987](#)
 - GetProductGuid, [987](#)
 - GetSchemaVersion, [987](#)
 - GetStandardNameSpace, [987](#)
 - GetSupportedSchemaVersions, [988](#)
 - GetToolTip, [988](#)
 - GetVendorName, [988](#)
 - GetVersionGuid, [988](#)

- InvalidateNodes, [989](#)
- LoadXMLFromFile, [989](#)
- LoadXMLFromFileInject, [989](#)
- LoadXMLFromString, [989](#)
- LoadXMLFromStringInject, [989](#)
- LoadXMLFromZIPData, [990](#)
- LoadXMLFromZIPFile, [990](#)
- NodeMap, [984](#)
- Poll, [990](#)
- Spinnaker::GenApi::ODevFileStreamBase
 - close, [993](#)
 - filebuf_type, [993](#)
 - ios_type, [993](#)
 - is_open, [993](#)
 - open, [993](#)
 - ostream_type, [993](#)
 - rdbuf, [994](#)
- Spinnaker::GenApi::ODevFileStreamBuf
 - ~ODevFileStreamBuf, [995](#)
 - close, [995](#)
 - is_open, [995](#)
 - ODevFileStreamBuf, [995](#)
 - open, [995](#)
 - overflow, [996](#)
 - sync, [996](#)
 - xspn, [996](#)
- Spinnaker::GenApi::PortNode
 - ~PortNode, [1001](#)
 - CacheChunkData, [1001](#)
 - GetChunkID, [1001](#)
 - GetPortHandle, [1002](#)
 - GetSwapEndianness, [1002](#)
 - PortNode, [1001](#)
 - Read, [1002](#)
 - Replay, [1002](#)
 - SetPortImpl, [1002](#)
 - SetReference, [1003](#)
 - StartRecording, [1003](#)
 - StopRecording, [1003](#)
 - Write, [1004](#)
- Spinnaker::GenApi::PortRecorder
 - ~PortRecorder, [1006](#)
 - GetAccessMode, [1006](#)
 - PortRecorder, [1005](#)
 - SetReference, [1006](#)
 - StartRecording, [1006](#)
 - StopRecording, [1006](#)
- Spinnaker::GenApi::PortReplay
 - ~PortReplay, [1008](#)
 - GetPortReplayHandle, [1009](#)
 - PortReplay, [1008](#)
 - Replay, [1009](#)
 - SetReference, [1009](#)
- Spinnaker::GenApi::RegisterNode
 - ~RegisterNode, [1013](#)
 - Get, [1013](#)
 - GetAddress, [1013](#)
 - GetLength, [1014](#)
 - RegisterNode, [1012](#), [1013](#)
 - Set, [1014](#)
 - SetReference, [1014](#)
- Spinnaker::GenApi::StringNode
 - ~StringNode, [1026](#)
 - GetMaxLength, [1026](#)
 - GetValue, [1026](#)
 - operator*, [1027](#)
 - operator(), [1027](#)
 - operator=, [1027](#)
 - SetReference, [1027](#)
 - SetValue, [1027](#)
 - StringNode, [1026](#)
- Spinnaker::GenApi::StringRegNode
 - ~StringRegNode, [1030](#)
 - SetReference, [1030](#)
 - StringRegNode, [1029](#), [1030](#)
- Spinnaker::GenApi::ValueNode
 - ~ValueNode, [1077](#)
 - FromString, [1078](#)
 - GetNode, [1079](#)
 - IsValueCacheValid, [1079](#)
 - SetReference, [1079](#)
 - ToString, [1079](#)
 - ValueNode, [1077](#)
- Spinnaker::GenApi::double_autovector_t
 - _pCount, [735](#)
 - _pv, [735](#)
 - ~double_autovector_t, [734](#)
 - double_autovector_t, [734](#)
 - operator delete, [734](#)
 - operator new, [734](#)
 - operator=, [734](#)
 - operator[], [735](#)
 - size, [735](#)
- Spinnaker::GenApi::int64_autovector_t
 - _pCount, [923](#)
 - _pv, [923](#)
 - ~int64_autovector_t, [922](#)
 - int64_autovector_t, [922](#)
 - operator delete, [922](#)
 - operator new, [922](#)
 - operator=, [922](#)
 - operator[], [923](#)
 - size, [923](#)
- Spinnaker::GenICam, [421](#)
 - getline, [422](#)
 - ThrowBadAlloc, [422](#)
- Spinnaker::GenICam::AutoLock
 - ~AutoLock, [430](#)
 - AutoLock, [430](#)
- Spinnaker::GenICam::CGlobalLock
 - ~CGlobalLock, [661](#)
 - CGlobalLock, [660](#), [661](#)
 - IsValid, [661](#)
 - Lock, [661](#)
 - m_DebugCount, [662](#)
 - TryLock, [661](#)

- Unlock, [662](#)
- Spinnaker::GenICam::CGlobalLockUnlocker
 - ~CGlobalLockUnlocker, [663](#)
 - CGlobalLockUnlocker, [663](#)
 - m_Lock, [664](#)
 - m_enabled, [664](#)
 - UnlockEarly, [663](#)
- Spinnaker::GenICam::CLock
 - ~CLock, [679](#)
 - CLock, [679](#)
 - Lock, [680](#)
 - TryLock, [680](#)
 - Unlock, [680](#)
- Spinnaker::GenICam::LockableObject
 - GetLock, [959](#)
 - Lock, [959](#)
 - m_Lock, [959](#)
- Spinnaker::GenICam::LockableObject::Lock
 - ~Lock, [958](#)
 - Lock, [958](#)
- Spinnaker::GenICam::Version_t
 - Major, [1080](#)
 - Minor, [1080](#)
 - SubMinor, [1080](#)
- Spinnaker::GenICam::gcstring
 - _npos, [784](#)
 - ~gcstring, [784](#)
 - append, [784](#)
 - assign, [784](#), [785](#)
 - c_str, [785](#)
 - compare, [785](#)
 - empty, [785](#)
 - find, [785](#), [786](#)
 - find_first_not_of, [786](#)
 - find_first_of, [786](#)
 - gcstring, [783](#)
 - length, [787](#)
 - max_size, [787](#)
 - npos, [791](#)
 - operator const char *, [787](#)
 - operator delete, [787](#)
 - operator new, [787](#)
 - operator!=, [788](#)
 - operator<, [789](#)
 - operator>, [789](#)
 - operator+, [790](#)
 - operator+=, [788](#)
 - operator=, [789](#)
 - operator==, [789](#)
 - resize, [789](#)
 - size, [789](#)
 - substr, [790](#)
 - swap, [790](#)
- Spinnaker::IArrivalEvent
 - ~IArrivalEvent, [804](#)
 - IArrivalEvent, [804](#)
 - OnDeviceArrival, [804](#)
 - operator=, [804](#)
- Spinnaker::ICameraBase
 - ~ICameraBase, [807](#)
 - BeginAcquisition, [807](#)
 - CameraInternal, [813](#)
 - DeInit, [807](#)
 - DiscoverMaxPacketSize, [807](#)
 - EndAcquisition, [808](#)
 - ForceIP, [808](#)
 - GetAccessMode, [808](#)
 - GetBufferOwnership, [808](#)
 - GetGuiXml, [808](#)
 - GetNextImage, [808](#)
 - GetNodeMap, [809](#)
 - GetNumDataStreams, [809](#)
 - GetNumImagesInUse, [809](#)
 - GetTLDeviceNodeMap, [809](#)
 - GetTLStreamNodeMap, [809](#)
 - GetUniqueID, [809](#)
 - GetUserBufferCount, [810](#)
 - GetUserBufferSize, [810](#)
 - GetUserBufferTotalSize, [810](#)
 - ICameraBase, [807](#)
 - Init, [810](#)
 - InterfacImpl, [813](#)
 - IsInitialized, [810](#)
 - IsStreaming, [810](#)
 - IsValid, [811](#)
 - m_pCameraBaseData, [813](#)
 - operator=, [811](#)
 - ReadPort, [811](#)
 - RegisterEvent, [811](#)
 - SetBufferOwnership, [811](#)
 - SetUserBuffers, [812](#)
 - TLDevice, [813](#)
 - TLStream, [813](#)
 - UnregisterEvent, [812](#)
 - WritePort, [812](#)
- Spinnaker::ICameraList
 - ~ICameraList, [815](#)
 - Append, [815](#)
 - CameraListImpl, [817](#)
 - Clear, [815](#)
 - GetByIndex, [815](#)
 - GetBySerial, [816](#)
 - GetSize, [816](#)
 - ICameraList, [815](#)
 - InterfacImpl, [817](#)
 - m_pCameraListData, [817](#)
 - operator=, [816](#)
 - operator[], [816](#)
 - RemoveByIndex, [816](#)
 - RemoveBySerial, [816](#)
- Spinnaker::IChunkData
 - ~IChunkData, [818](#)
 - GetBlackLevel, [819](#)
 - GetCRC, [819](#)
 - GetCounterValue, [819](#)
 - GetEncoderValue, [819](#)

- GetExposureEndLineStatusAll, 819
- GetExposureTime, 820
- GetFrameID, 820
- GetGain, 820
- GetHeight, 820
- GetImage, 820
- GetInferenceConfidence, 820
- GetInferenceResult, 821
- GetLinePitch, 821
- GetLineStatusAll, 821
- GetOffsetX, 821
- GetOffsetY, 821
- GetPartSelector, 821
- GetPixelDynamicRangeMax, 822
- GetPixelDynamicRangeMin, 822
- GetScan3dAxisMax, 822
- GetScan3dAxisMin, 822
- GetScan3dCoordinateOffset, 822
- GetScan3dCoordinateReferenceValue, 822
- GetScan3dCoordinateScale, 823
- GetScan3dInvalidDataValue, 823
- GetScan3dTransformValue, 823
- GetScanLineSelector, 823
- GetSequencerSetActive, 823
- GetSerialDataLength, 823
- GetStreamChannelID, 824
- GetTimerValue, 824
- GetTimestamp, 824
- GetTimestampLatchValue, 824
- GetTransferBlockID, 824
- GetTransferQueueCurrentBlockCount, 824
- GetWidth, 825
- IChunkData, 819
- SetChunks, 825
- Spinnaker::IDataStream
 - ~IDataStream, 826
 - AddChunks, 826
 - AnnounceImage, 826
 - CleanupChunkAdapter, 827
 - FillCRCInfo, 827
 - FlushQueueAllDiscard, 827
 - GetNextImage, 827
 - GetNextImageInternal, 827
 - GetNodeMap, 827
 - GetNumImagesInUse, 827
 - GetPort, 828
 - IDataStream, 826
 - InitChunkAdapter, 828
 - IsImageInUse, 828
 - IsStreaming, 828
 - KillBufferEvent, 828
 - RegisterImageEvent, 828
 - ReleaseImage, 828
 - RevokeImages, 829
 - StartStream, 829
 - StopStream, 829
 - TransportLayerStreamInfo, 829
 - UnregisterImageEvent, 829
 - WaitOnImageEvent, 829
- Spinnaker::IDeviceEvent
 - ~IDeviceEvent, 835
 - GetDeviceEventId, 836
 - GetDeviceEventName, 836
 - IDeviceEvent, 835
 - OnDeviceEvent, 836
 - operator=, 836
- Spinnaker::IImage
 - ~IImage, 838
 - CalculateStatistics, 839
 - CheckCRC, 839
 - Convert, 839
 - DEPRECATED_FUNC, 839, 840
 - DeepCopy, 839
 - GetBitsPerPixel, 840
 - GetBufferSize, 840
 - GetChunkData, 840
 - GetChunkLayoutId, 840
 - GetColorProcessing, 841
 - GetData, 841
 - GetFrameID, 841
 - GetHeight, 841
 - GetID, 841
 - GetImageSize, 841
 - GetImageStatus, 842
 - GetNumChannels, 842
 - GetPayloadType, 842
 - GetPixelFormat, 842
 - GetPixelFormatIntType, 842
 - GetPixelFormatName, 842
 - GetPrivateData, 843
 - GetStride, 843
 - GetTLPayloadType, 843
 - GetTLPixelFormat, 843
 - GetTLPixelFormatNamespace, 843
 - GetTimeStamp, 843
 - GetValidPayloadSize, 844
 - GetWidth, 844
 - GetXOffset, 844
 - GetXPadding, 844
 - GetYOffset, 844
 - GetYPadding, 844
 - HasCRC, 845
 - IImage, 838
 - IsInUse, 845
 - IsIncomplete, 845
 - Release, 845
 - ResetImage, 845
 - Save, 846, 847
- Spinnaker::IImageEvent
 - ~IImageEvent, 849
 - IImageEvent, 849
 - OnImageEvent, 849
 - operator=, 849
- Spinnaker::IImageStatistics
 - ~IImageStatistics, 851
 - DisableAll, 851

- EnableAll, [851](#)
- EnableGreyOnly, [851](#)
- EnableHSLOnly, [851](#)
- EnableRGBOnly, [852](#)
- GetChannelStatus, [852](#)
- GetHistogram, [852](#)
- GetMean, [852](#)
- GetNumPixelValues, [852](#)
- GetPixelValueRange, [853](#)
- GetRange, [853](#)
- GetStatistics, [853](#)
- ImageStatistics, [851](#)
- SetChannelStatus, [853](#)
- Spinnaker::IInterface
 - ~IInterface, [855](#)
 - GetCameras, [856](#)
 - GetTLNodeMap, [856](#)
 - IInterface, [855](#), [856](#)
 - InterfaceInternal, [857](#)
 - IsInUse, [856](#)
 - m_pInterfaceData, [858](#)
 - operator=, [856](#)
 - RegisterEvent, [856](#)
 - SendActionCommand, [857](#)
 - TLInterface, [858](#)
 - UnregisterEvent, [857](#)
 - UpdateCameras, [857](#)
- Spinnaker::IInterfaceEvent
 - ~IInterfaceEvent, [859](#)
 - IInterfaceEvent, [859](#)
 - OnDeviceArrival, [860](#)
 - OnDeviceRemoval, [860](#)
 - operator=, [860](#)
- Spinnaker::IInterfaceList
 - ~IInterfaceList, [861](#)
 - Clear, [862](#)
 - GetByIndex, [862](#)
 - GetSize, [862](#)
 - IInterfaceList, [862](#)
 - m_pInterfaceListData, [863](#)
 - operator=, [862](#)
 - operator[], [863](#)
- Spinnaker::ILoggingEvent
 - ~ILoggingEvent, [864](#)
 - ILoggingEvent, [864](#)
 - OnLogEvent, [865](#)
 - operator=, [865](#)
- Spinnaker::IRemovalEvent
 - ~IRemovalEvent, [947](#)
 - IRemovalEvent, [947](#)
 - OnDeviceRemoval, [947](#)
 - operator=, [947](#)
- Spinnaker::ISystem
 - ~ISystem, [949](#)
 - GetCameras, [949](#)
 - GetInterfaces, [950](#)
 - GetLibraryVersion, [950](#)
 - GetLoggingEventPriorityLevel, [950](#)
 - GetTLNodeMap, [950](#)
 - ISystem, [949](#)
 - IsInUse, [950](#)
 - operator=, [950](#)
 - RegisterInterfaceEvent, [951](#)
 - RegisterLoggingEvent, [951](#)
 - ReleaseInstance, [951](#)
 - SendActionCommand, [951](#)
 - SetLoggingEventPriorityLevel, [951](#)
 - SystemPtrInternal, [953](#)
 - TLSystem, [953](#)
 - UnregisterAllLoggingEvent, [952](#)
 - UnregisterInterfaceEvent, [952](#)
 - UnregisterLoggingEvent, [952](#)
 - UpdateCameras, [952](#)
 - UpdateInterfaceList, [952](#)
- Spinnaker::Image
 - ~Image, [869](#)
 - CalculateStatistics, [870](#)
 - CheckCRC, [871](#)
 - Convert, [871](#)
 - Create, [871](#), [872](#)
 - CreateShared, [872](#)
 - DEPRECATED_FUNC, [873–876](#)
 - DeepCopy, [872](#), [873](#)
 - GetBitsPerPixel, [877](#)
 - GetBufferSize, [877](#)
 - GetChunkData, [877](#)
 - GetChunkLayoutId, [878](#)
 - GetColorProcessing, [878](#)
 - GetData, [878](#)
 - GetDefaultColorProcessing, [879](#)
 - GetFrameID, [879](#)
 - GetHeight, [879](#)
 - GetID, [880](#)
 - GetImageSize, [880](#)
 - GetImageStatus, [880](#)
 - GetImageStatusDescription, [881](#)
 - GetNumChannels, [881](#)
 - GetPayloadType, [881](#)
 - GetPixelFormat, [882](#)
 - GetPixelFormatIntType, [882](#)
 - GetPixelFormatName, [882](#)
 - GetPrivateData, [883](#)
 - GetStride, [883](#)
 - GetTLPayloadType, [884](#)
 - GetTLPixelFormat, [884](#)
 - GetTLPixelFormatNamespace, [884](#)
 - GetTimeStamp, [883](#)
 - GetValidPayloadSize, [885](#)
 - GetWidth, [885](#)
 - GetXOffset, [885](#)
 - GetXPadding, [886](#)
 - GetYOffset, [886](#)
 - GetYPadding, [886](#)
 - HasCRC, [887](#)
 - IDataStream, [892](#)
 - Image, [870](#)

- ImageConverter, 892
- ImageFiler, 892
- ImageStatsCalculator, 892
- ImageUtilityImpl, 893
- IsCompressed, 887
- IsInUse, 887
- IsIncomplete, 887
- m_pImageData, 893
- Release, 888
- ResetImage, 888
- Save, 889–891
- SetDefaultColorProcessing, 892
- Stream, 893
- Spinnaker::ImageEvent
 - ~ImageEvent, 895
 - ImageEvent, 894
 - OnImageEvent, 895
 - operator=, 895
- Spinnaker::ImagePtr
 - ~ImagePtr, 897
 - ImagePtr, 897
 - operator=, 898
- Spinnaker::ImageStatistics
 - ~ImageStatistics, 900
 - DisableAll, 900
 - EnableAll, 900
 - EnableGreyOnly, 900
 - EnableHSLOnly, 901
 - EnableRGBOnly, 901
 - GetChannelStatus, 901
 - GetHistogram, 901
 - GetMean, 902
 - GetNumPixelValues, 902
 - GetPixelValueRange, 902
 - GetRange, 903
 - GetStatistics, 903
 - ImageStatistics, 900
 - ImageStatsCalculator, 905
 - operator=, 904
 - SetChannelStatus, 904
- Spinnaker::ImageUtility
 - CreateNormalized, 906–908
 - CreateScaled, 908, 909
 - ImageScalingAlgorithm, 906
- Spinnaker::ImageUtilityHeatmap
 - CreateHeatmap, 910, 911
 - GetHeatmapColorGradient, 911
 - GetHeatmapRange, 912
 - HeatmapColor, 910
 - SetHeatmapColorGradient, 912
 - SetHeatmapRange, 913
- Spinnaker::ImageUtilityPolarization
 - CreateAolp, 915
 - CreateDolp, 916
 - CreateStokesS0, 916, 917
 - CreateStokesS1, 917, 919
 - CreateStokesS2, 919, 920
 - ExtractPolarQuadrant, 920
 - PolarizationQuadrant, 914
- Spinnaker::Interface
 - ~Interface, 931
 - GetCameras, 931
 - GetTLNodeMap, 932
 - InterfaceInternal, 934
 - IsInUse, 932
 - RegisterEvent, 932
 - SendActionCommand, 933
 - UnregisterEvent, 933
 - UpdateCameras, 934
- Spinnaker::InterfaceEvent
 - ~InterfaceEvent, 936
 - InterfaceEvent, 936
 - OnDeviceArrival, 936
 - OnDeviceRemoval, 937
 - operator=, 937
- Spinnaker::InterfaceList
 - ~InterfaceList, 939
 - Clear, 939
 - GetByIndex, 939
 - GetSize, 940
 - InterfaceList, 939
 - operator=, 940
 - operator[], 940
 - SystemImpl, 940
- Spinnaker::InterfacePtr
 - InterfacePtr, 942
- Spinnaker::JPEGOption
 - JPEGOption, 954
 - progressive, 954
 - quality, 954
 - reserved, 954
- Spinnaker::JPG2Option
 - JPG2Option, 955
 - quality, 955
 - reserved, 955
- Spinnaker::LibraryVersion
 - build, 956
 - major, 956
 - minor, 957
 - type, 957
- Spinnaker::LoggingEvent
 - ~LoggingEvent, 961
 - LoggingEvent, 961
 - OnLogEvent, 961
 - operator=, 962
- Spinnaker::LoggingEventData
 - ~LoggingEventData, 963
 - GetCategoryName, 963
 - GetLogMessage, 963
 - GetNDC, 964
 - GetPriority, 964
 - GetPriorityName, 964
 - GetThreadName, 964
 - GetTimestamp, 965
 - LoggingEventData, 963
 - SystemImpl, 965

- Spinnaker::LoggingEventDataPtr
 - LoggingEventDataPtr, [967](#)
- Spinnaker::PGMOption
 - binaryFile, [997](#)
 - PGMOption, [997](#)
 - reserved, [997](#)
- Spinnaker::PNGOption
 - compressionLevel, [998](#)
 - interlaced, [998](#)
 - PNGOption, [998](#)
 - reserved, [998](#)
- Spinnaker::PPMOption
 - binaryFile, [1010](#)
 - PPMOption, [1010](#)
 - reserved, [1010](#)
- Spinnaker::RemovalEvent
 - ~RemovalEvent, [1016](#)
 - OnDeviceRemoval, [1016](#)
 - operator=, [1017](#)
 - RemovalEvent, [1016](#)
- Spinnaker::System
 - ~System, [1032](#)
 - GetCameras, [1033](#)
 - GetInstance, [1033](#)
 - GetInterfaces, [1034](#)
 - GetLibraryVersion, [1034](#)
 - GetLoggingEventPriorityLevel, [1035](#)
 - GetTLNodeMap, [1035](#)
 - IsInUse, [1035](#)
 - RegisterInterfaceEvent, [1036](#)
 - RegisterLoggingEvent, [1036](#)
 - ReleaseInstance, [1036](#)
 - SendActionCommand, [1037](#)
 - SetLoggingEventPriorityLevel, [1037](#)
 - System, [1033](#)
 - UnregisterAllLoggingEvent, [1038](#)
 - UnregisterInterfaceEvent, [1038](#)
 - UnregisterLoggingEvent, [1039](#)
 - UpdateCameras, [1039](#)
 - UpdateInterfaceList, [1039](#)
- Spinnaker::SystemPtr
 - ~SystemPtr, [1042](#)
 - SystemPtr, [1041](#)
- Spinnaker::TIFFOption
 - compression, [1043](#)
 - CompressionMethod, [1043](#)
 - reserved, [1043](#)
 - TIFFOption, [1043](#)
- Spinnaker::TransportLayerDevice
 - ~TransportLayerDevice, [1046](#)
 - CameraBase, [1046](#)
 - CameraInternal, [1046](#)
 - DeviceAccessStatus, [1047](#)
 - DeviceCurrentSpeed, [1047](#)
 - DeviceDisplayName, [1047](#)
 - DeviceDriverVersion, [1047](#)
 - DeviceEndianessMechanism, [1047](#)
 - DeviceId, [1048](#)
 - DeviceInstanceId, [1048](#)
 - DevicesUpdater, [1048](#)
 - DeviceLinkSpeed, [1048](#)
 - DeviceLocation, [1048](#)
 - DeviceModelName, [1048](#)
 - DeviceMulticastMonitorMode, [1049](#)
 - DeviceSerialNumber, [1049](#)
 - DeviceType, [1049](#)
 - DeviceU3VProtocol, [1049](#)
 - DeviceUserID, [1049](#)
 - DeviceVendorName, [1049](#)
 - DeviceVersion, [1050](#)
 - GUIXMLLocation, [1053](#)
 - GUIXMLPath, [1053](#)
 - GenICamXMLLocation, [1050](#)
 - GenICamXMLPath, [1050](#)
 - GevCCP, [1050](#)
 - GevDeviceDiscoverMaximumPacketSize, [1050](#)
 - GevDeviceForceIP, [1050](#)
 - GevDeviceGateway, [1051](#)
 - GevDeviceIPAddress, [1051](#)
 - GevDevicesWrongSubnet, [1051](#)
 - GevDeviceMACAddress, [1051](#)
 - GevDeviceMaximumPacketSize, [1051](#)
 - GevDeviceMaximumRetryCount, [1051](#)
 - GevDeviceModelsBigEndian, [1052](#)
 - GevDevicePort, [1052](#)
 - GevDeviceReadAndWriteTimeout, [1052](#)
 - GevDeviceSubnetMask, [1052](#)
 - GevVersionMajor, [1052](#)
 - GevVersionMinor, [1052](#)
 - ICameraBase, [1047](#)
 - TransportLayerDevice, [1046](#)
- Spinnaker::TransportLayerInterface
 - ~TransportLayerInterface, [1056](#)
 - ActionCommand, [1056](#)
 - AutoForceIP, [1057](#)
 - DeviceAccessStatus, [1057](#)
 - DeviceCount, [1057](#)
 - DeviceID, [1057](#)
 - DeviceModelName, [1057](#)
 - DeviceSelector, [1057](#)
 - DeviceUnlock, [1058](#)
 - DeviceUpdateList, [1058](#)
 - DeviceVendorName, [1058](#)
 - FilterDriverStatus, [1058](#)
 - GevActionDeviceKey, [1058](#)
 - GevActionGroupKey, [1058](#)
 - GevActionGroupMask, [1059](#)
 - GevActionTime, [1059](#)
 - GevDeviceIPAddress, [1059](#)
 - GevDeviceMACAddress, [1059](#)
 - GevDeviceSubnetMask, [1059](#)
 - GevInterfaceGateway, [1059](#)
 - GevInterfaceIPAddress, [1060](#)
 - GevInterfaceMACAddress, [1060](#)
 - GevInterfaceMTU, [1060](#)
 - GevInterfaceReceiveLinkSpeed, [1060](#)

- GevInterfaceSubnetMask, [1060](#)
- GevInterfaceTransmitLinkSpeed, [1060](#)
- HostAdapterDriverVersion, [1061](#)
- HostAdapterName, [1061](#)
- HostAdapterVendor, [1061](#)
- IInterface, [1056](#)
- IncompatibleDeviceCount, [1061](#)
- IncompatibleDeviceID, [1061](#)
- IncompatibleDeviceModelName, [1061](#)
- IncompatibleDeviceSelector, [1062](#)
- IncompatibleDeviceVendorName, [1062](#)
- IncompatibleGevDeviceIPAddress, [1062](#)
- IncompatibleGevDeviceMACAddress, [1062](#)
- IncompatibleGevDeviceSubnetMask, [1062](#)
- Interface, [1056](#)
- InterfaceDisplayName, [1062](#)
- InterfaceID, [1063](#)
- InterfaceInternal, [1056](#)
- InterfaceType, [1063](#)
- POEStatus, [1063](#)
- TransportLayerInterface, [1056](#)
- Spinnaker::TransportLayerStream
 - ~TransportLayerStream, [1065](#)
 - CameraBase, [1065](#)
 - CameraInternal, [1065](#)
 - GevFailedPacketCount, [1066](#)
 - GevMaximumNumberResendBuffers, [1066](#)
 - GevMaximumNumberResendRequests, [1066](#)
 - GevPacketResendMode, [1066](#)
 - GevPacketResendTimeout, [1066](#)
 - GevResendPacketCount, [1067](#)
 - GevResendRequestCount, [1067](#)
 - GevTotalPacketCount, [1067](#)
 - ICameraBase, [1066](#)
 - StreamBlockTransferSize, [1067](#)
 - StreamBufferCountManual, [1067](#)
 - StreamBufferCountMax, [1067](#)
 - StreamBufferCountMode, [1068](#)
 - StreamBufferCountResult, [1068](#)
 - StreamBufferHandlingMode, [1068](#)
 - StreamBufferUnderrunCount, [1068](#)
 - StreamCRCCheckEnable, [1068](#)
 - StreamDefaultBufferCount, [1068](#)
 - StreamDefaultBufferCountMax, [1069](#)
 - StreamDefaultBufferCountMode, [1069](#)
 - StreamFailedBufferCount, [1069](#)
 - StreamID, [1069](#)
 - StreamTotalBufferCount, [1069](#)
 - StreamType, [1069](#)
 - TransportLayerStream, [1065](#)
- Spinnaker::TransportLayerSystem
 - ~TransportLayerSystem, [1071](#)
 - EnumerateGEVInterfaces, [1071](#)
 - ISystem, [1071](#)
 - System, [1071](#)
 - SystemPtrInternal, [1071](#)
 - TransportLayerSystem, [1070](#), [1071](#)
- Spinnaker::Video, [423](#)
- Spinnaker::Video::AVIOption
 - AVIOption, [430](#)
 - frameRate, [431](#)
 - reserved, [431](#)
- Spinnaker::Video::H264Option
 - bitrate, [801](#)
 - frameRate, [801](#)
 - H264Option, [801](#)
 - height, [802](#)
 - reserved, [802](#)
 - width, [802](#)
- Spinnaker::Video::MJPGOption
 - frameRate, [970](#)
 - MJPGOption, [970](#)
 - quality, [971](#)
 - reserved, [971](#)
- Spinnaker::Video::SpinVideo
 - ~SpinVideo, [1020](#)
 - Append, [1020](#)
 - Close, [1021](#)
 - Open, [1021](#), [1023](#)
 - SetMaximumFileSize, [1023](#)
 - SpinVideo, [1020](#)
- SpinnakerLogLevel
 - Spinnaker Definitions, [188](#)
- Standard
 - Types Enums, [359](#)
- StartRecording
 - Spinnaker::GenApi::PortNode, [1003](#)
 - Spinnaker::GenApi::PortRecorder, [1006](#)
- StartStream
 - Spinnaker::IDataStream, [829](#)
- StatisticsChannel
 - Spinnaker Definitions, [188](#)
- Status
 - Spinnaker::ActionCommandResult, [425](#)
- StopRecording
 - IPortRecorder Interface, [312](#)
 - Spinnaker::GenApi::PortNode, [1003](#)
 - Spinnaker::GenApi::PortRecorder, [1006](#)
- StopStream
 - Spinnaker::IDataStream, [829](#)
- StoreToBag
 - Spinnaker::GenApi::CFeatureBag, [656](#)
- Stream
 - Spinnaker::Event, [759](#)
 - Spinnaker::Image, [893](#)
- StreamBlockTransferSize
 - Spinnaker::TransportLayerStream, [1067](#)
- StreamBufferCountManual
 - Spinnaker::TransportLayerStream, [1067](#)
- StreamBufferCountMax
 - Spinnaker::TransportLayerStream, [1067](#)
- StreamBufferCountMode
 - Spinnaker::TransportLayerStream, [1068](#)
- StreamBufferCountModeEnum
 - TransportLayerDefs Class, [200](#)
- StreamBufferCountResult

- Spinnaker::TransportLayerStream, 1068
- StreamBufferHandlingMode
 - Spinnaker::TransportLayerStream, 1068
- StreamBufferHandlingModeEnum
 - TransportLayerDefs Class, 201
- StreamBufferUnderrunCount
 - Spinnaker::TransportLayerStream, 1068
- StreamCRCCheckEnable
 - Spinnaker::TransportLayerStream, 1068
- StreamChannelId
 - GVCP_EVENT_ITEM_EXTENDED_ID, 795
 - GVCP_EVENT_ITEM, 792
- StreamDefaultBufferCount
 - Spinnaker::TransportLayerStream, 1068
- StreamDefaultBufferCountMax
 - Spinnaker::TransportLayerStream, 1069
- StreamDefaultBufferCountMode
 - Spinnaker::TransportLayerStream, 1069
- StreamDefaultBufferCountModeEnum
 - TransportLayerDefs Class, 201
- StreamFailedBufferCount
 - Spinnaker::TransportLayerStream, 1069
- StreamID
 - Spinnaker::TransportLayerStream, 1069
- StreamTotalBufferCount
 - Spinnaker::TransportLayerStream, 1069
- StreamType
 - Spinnaker::TransportLayerStream, 1069
- StreamTypeEnum
 - TransportLayerDefs Class, 202
- StringList_t
 - Types Enums, 354
- StringNode, 1024
 - Spinnaker::GenApi::StringNode, 1026
- StringNode Class, 347
 - CStringRef, 347
- StringRegNode, 1028
 - Spinnaker::GenApi::StringRegNode, 1029, 1030
- StringRegNode Class, 348
- StructPort Class, 349
- SubMinor
 - Spinnaker::GenICam::Version_t, 1080
- substr
 - Spinnaker::GenICam::gcstring, 790
- swap
 - Spinnaker::GenICam::gcstring, 790
- sync
 - Spinnaker::GenApi::ODevFileStreamBuf, 996
- Synch Class, 350
- System, 1031
 - Spinnaker::System, 1033
 - Spinnaker::TransportLayerSystem, 1071
- System Class, 193
- System.h
 - FLIR_SPINNAKER_VERSION_BUILD, 1334
 - FLIR_SPINNAKER_VERSION_MAJOR, 1335
 - FLIR_SPINNAKER_VERSION_MINOR, 1335
 - FLIR_SPINNAKER_VERSION_TYPE, 1335
- SystemImpl
 - Spinnaker::InterfaceList, 940
 - Spinnaker::LoggingEventData, 965
- SystemPtr, 1040
 - Spinnaker::SystemPtr, 1041
- SystemPtr Class, 194
- SystemPtrInternal
 - Spinnaker::ISystem, 953
 - Spinnaker::TransportLayerSystem, 1071
- TIFFOption, 1042
 - Spinnaker::TIFFOption, 1043
- TLDevice
 - Spinnaker::ICameraBase, 813
- TLInterface
 - Spinnaker::IInterface, 858
- TLParamsLocked
 - Spinnaker::Camera, 581
- TLStream
 - Spinnaker::ICameraBase, 813
- TLSystem
 - Spinnaker::ISystem, 953
- Test0001
 - Spinnaker::Camera, 578
- TestEventGenerate
 - Spinnaker::Camera, 579
- TestPattern
 - Spinnaker::Camera, 579
- TestPatternEnums
 - CameraDefs Class, 143
- TestPatternGeneratorSelector
 - Spinnaker::Camera, 579
- TestPatternGeneratorSelectorEnums
 - CameraDefs Class, 143
- TestPendingAck
 - Spinnaker::Camera, 579
- ThrowBadAlloc
 - Spinnaker::GenICam, 422
- TimerDelay
 - Spinnaker::Camera, 579
- TimerDuration
 - Spinnaker::Camera, 580
- TimerReset
 - Spinnaker::Camera, 580
- TimerSelector
 - Spinnaker::Camera, 580
- TimerSelectorEnums
 - CameraDefs Class, 144
- TimerStatus
 - Spinnaker::Camera, 580
- TimerStatusEnums
 - CameraDefs Class, 144
- TimerTriggerActivation
 - Spinnaker::Camera, 580
- TimerTriggerActivationEnums
 - CameraDefs Class, 144
- TimerTriggerSource
 - Spinnaker::Camera, 580
- TimerTriggerSourceEnums

- CameraDefs Class, [145](#)
- TimerValue
 - Spinnaker::Camera, [581](#)
- Timestamp
 - Spinnaker::Camera, [581](#)
 - U3V_EVENT_DATA, [1074](#)
- TimestampHigh
 - GVCP_EVENT_ITEM_EXTENDED_ID, [795](#)
 - GVCP_EVENT_ITEM, [792](#)
- TimestampLatch
 - Spinnaker::Camera, [581](#)
- TimestampLatchValue
 - Spinnaker::Camera, [581](#)
- TimestampLow
 - GVCP_EVENT_ITEM_EXTENDED_ID, [795](#)
 - GVCP_EVENT_ITEM, [793](#)
- TimestampReset
 - Spinnaker::Camera, [581](#)
- ToString
 - ISectorDigit Interface, [317](#)
 - IValue Class, [321](#)
 - Spinnaker::GenApi::CNodeMapFactory, [694](#)
 - Spinnaker::GenApi::CSelectorSet, [725](#)
 - Spinnaker::GenApi::EAccessModeClass, [736](#)
 - Spinnaker::GenApi::ECachingModeClass, [737](#)
 - Spinnaker::GenApi::EDisplayNotationClass, [738](#)
 - Spinnaker::GenApi::EEndianessClass, [739](#)
 - Spinnaker::GenApi::EGenApiSchemaVersion↔
Class, [740](#)
 - Spinnaker::GenApi::EInputDirectionClass, [741](#)
 - Spinnaker::GenApi::ENameSpaceClass, [742](#)
 - Spinnaker::GenApi::ERepresentationClass, [752](#)
 - Spinnaker::GenApi::ESignClass, [753](#)
 - Spinnaker::GenApi::ESlopeClass, [754](#)
 - Spinnaker::GenApi::EStandardNameSpaceClass,
[755](#)
 - Spinnaker::GenApi::EVisibilityClass, [760](#)
 - Spinnaker::GenApi::EYesNoClass, [765](#)
 - Spinnaker::GenApi::ValueNode, [1079](#)
- ToXml
 - Spinnaker::GenApi::CNodeMapFactory, [694](#)
- Tokenize
 - GCUilities Utility, [261](#)
- TransferAbort
 - Spinnaker::Camera, [582](#)
- TransferBlockCount
 - Spinnaker::Camera, [582](#)
- TransferBurstCount
 - Spinnaker::Camera, [582](#)
- TransferComponentSelector
 - Spinnaker::Camera, [582](#)
- TransferComponentSelectorEnums
 - CameraDefs Class, [146](#)
- TransferControlMode
 - Spinnaker::Camera, [582](#)
- TransferControlModeEnums
 - CameraDefs Class, [146](#)
- TransferOperationMode
 - Spinnaker::Camera, [582](#)
- TransferOperationModeEnums
 - CameraDefs Class, [147](#)
- TransferPause
 - Spinnaker::Camera, [583](#)
- TransferQueueCurrentBlockCount
 - Spinnaker::Camera, [583](#)
- TransferQueueMaxBlockCount
 - Spinnaker::Camera, [583](#)
- TransferQueueMode
 - Spinnaker::Camera, [583](#)
- TransferQueueModeEnums
 - CameraDefs Class, [147](#)
- TransferQueueOverflowCount
 - Spinnaker::Camera, [583](#)
- TransferResume
 - Spinnaker::Camera, [583](#)
- TransferSelector
 - Spinnaker::Camera, [584](#)
- TransferSelectorEnums
 - CameraDefs Class, [147](#)
- TransferStart
 - Spinnaker::Camera, [584](#)
- TransferStatus
 - Spinnaker::Camera, [584](#)
- TransferStatusSelector
 - Spinnaker::Camera, [584](#)
- TransferStatusSelectorEnums
 - CameraDefs Class, [148](#)
- TransferStop
 - Spinnaker::Camera, [584](#)
- TransferStreamChannel
 - Spinnaker::Camera, [584](#)
- TransferTriggerActivation
 - Spinnaker::Camera, [585](#)
- TransferTriggerActivationEnums
 - CameraDefs Class, [148](#)
- TransferTriggerMode
 - Spinnaker::Camera, [585](#)
- TransferTriggerModeEnums
 - CameraDefs Class, [148](#)
- TransferTriggerSelector
 - Spinnaker::Camera, [585](#)
- TransferTriggerSelectorEnums
 - CameraDefs Class, [149](#)
- TransferTriggerSource
 - Spinnaker::Camera, [585](#)
- TransferTriggerSourceEnums
 - CameraDefs Class, [149](#)
- TransportLayerDefs Class, [196](#)
 - DeviceAccessStatusEnum, [197](#)
 - DeviceCurrentSpeedEnum, [198](#)
 - DeviceEndianessMechanismEnum, [198](#)
 - DeviceTypeEnum, [198](#)
 - FilterDriverStatusEnum, [199](#)
 - GUXMLLocationEnum, [200](#)
 - GenICamXMLLocationEnum, [199](#)
 - GevCCPEnum, [199](#)

- POEStatusEnum, 200
- StreamBufferCountModeEnum, 200
- StreamBufferHandlingModeEnum, 201
- StreamDefaultBufferCountModeEnum, 201
- StreamTypeEnum, 202
- TransportLayerDevice, 1044
 - Spinnaker::TransportLayerDevice, 1046
- TransportLayerDevice Class, 203
- TransportLayerInterface, 1053
 - Spinnaker::TransportLayerInterface, 1056
- TransportLayerInterface Class, 204
- TransportLayerStream, 1063
 - Spinnaker::TransportLayerStream, 1065
- TransportLayerStream Class, 205
- TransportLayerStreamInfo
 - Spinnaker::IDataStream, 829
- TransportLayerSystem, 1070
 - Spinnaker::TransportLayerSystem, 1070, 1071
- TransportLayerSystem Class, 206
- TriggerActivation
 - Spinnaker::Camera, 585
- TriggerActivationEnums
 - CameraDefs Class, 150
- TriggerDelay
 - Spinnaker::Camera, 585
- TriggerDivider
 - Spinnaker::Camera, 586
- TriggerEventTest
 - Spinnaker::Camera, 586
- TriggerMode
 - Spinnaker::Camera, 586
- TriggerModeEnums
 - CameraDefs Class, 151
- TriggerMultiplier
 - Spinnaker::Camera, 586
- TriggerOverlap
 - Spinnaker::Camera, 586
- TriggerOverlapEnums
 - CameraDefs Class, 151
- TriggerSelector
 - Spinnaker::Camera, 587
- TriggerSelectorEnums
 - CameraDefs Class, 151
- TriggerSoftware
 - Spinnaker::Camera, 587
- TriggerSource
 - Spinnaker::Camera, 587
- TriggerSourceEnums
 - CameraDefs Class, 151
- TryLock
 - Spinnaker::GenApi::CLock, 677
 - Spinnaker::GenICam::CGlobalLock, 661
 - Spinnaker::GenICam::CLock, 680
- type
 - Spinnaker::LibraryVersion, 957
- Types Enums, 352
 - _UndefinedRepresentation, 354
 - Automatic, 360
 - Beginner, 361
 - Boolean, 359
 - CL, 360
 - Custom, 359
 - Decreasing, 360
 - EAccessMode, 355
 - ECachingMode, 355
 - EDisplayNotation, 355
 - EEndianess, 357
 - EGenApiSchemaVersion, 357
 - ElncMode, 357
 - EInputDirection, 358
 - EInterfaceType, 358
 - ELinkType, 358
 - ENamespace, 359
 - ERepresentation, 359
 - ESign, 359
 - ESlope, 360
 - EStandardNameSpace, 360
 - EVisibility, 360
 - EXMLValidation, 361
 - EYesNo, 361
 - Expert, 361
 - Guru, 361
 - Increasing, 360
 - Invisible, 361
 - Linear, 359
 - Logarithmic, 359
 - NA, 355
 - NI, 355
 - No, 361
 - None, 360
 - RO, 355
 - RW, 355
 - Signed, 360
 - Standard, 359
 - StringList_t, 354
 - Unsigned, 360
 - Varying, 360
 - WO, 355
 - Yes, 361
- Types.h
 - interface, 1321
- U3V_CHUNK_TRAILER, 1072
 - ChunkID, 1072
 - ChunkLength, 1072
- U3V_COMMAND_HEADER, 1073
 - CommandId, 1073
 - Flags, 1073
 - Length, 1073
 - Prefix, 1073
 - ReqId, 1073
- U3V_EVENT_DATA, 1074
 - EventId, 1074
 - Reserved, 1074
 - Timestamp, 1074
- U3V_EVENT_MESSAGE, 1075
 - CommandHeader, 1075

- EventData, [1075](#)
- U3V_EVENT_PREFIX
 - Spinnaker::GenApi, [420](#)
- USE_TEMP_CACHE_FILE
 - GCUtilities.h, [1242](#)
- underflow
 - Spinnaker::GenApi::IDevFileStreamBuf, [834](#)
- Unlock
 - Spinnaker::GenApi::CLock, [678](#)
 - Spinnaker::GenICam::CGlobalLock, [662](#)
 - Spinnaker::GenICam::CLock, [680](#)
- UnlockEarly
 - Spinnaker::GenICam::CGlobalLockUnlocker, [663](#)
- UnregisterAllLoggingEvent
 - Spinnaker::ISystem, [952](#)
 - Spinnaker::System, [1038](#)
- UnregisterEvent
 - Spinnaker::CameraBase, [605](#)
 - Spinnaker::ICameraBase, [812](#)
 - Spinnaker::IInterface, [857](#)
 - Spinnaker::Interface, [933](#)
- UnregisterImageEvent
 - Spinnaker::IDataStream, [829](#)
- UnregisterInterfaceEvent
 - Spinnaker::ISystem, [952](#)
 - Spinnaker::System, [1038](#)
- UnregisterLoggingEvent
 - Spinnaker::ISystem, [952](#)
 - Spinnaker::System, [1039](#)
- Unsigned
 - Types Enums, [360](#)
- UpdateBuffer
 - Spinnaker::GenApi::CChunkAdapter, [618](#)
 - Spinnaker::GenApi::CChunkPort, [632](#)
- UpdateCameras
 - Spinnaker::IInterface, [857](#)
 - Spinnaker::ISystem, [952](#)
 - Spinnaker::Interface, [934](#)
 - Spinnaker::System, [1039](#)
- UpdateFirmware
 - SpinUpdate.h, [1330](#)
- UpdateFirmwareConsole
 - SpinUpdate.h, [1331](#)
- UpdateInterfaceList
 - Spinnaker::ISystem, [952](#)
 - Spinnaker::System, [1039](#)
- UpdaterMessageCallback
 - SpinUpdate.h, [1331](#)
- UpdaterProgressCallback
 - SpinUpdate.h, [1331](#)
- UrlDecode
 - GCUtilities Utility, [261](#)
- UrlEncode
 - GCUtilities Utility, [262](#)
- UserOutputSelector
 - Spinnaker::Camera, [587](#)
- UserOutputSelectorEnums
 - CameraDefs Class, [152](#)
- UserOutputValue
 - Spinnaker::Camera, [587](#)
- UserOutputValueAll
 - Spinnaker::Camera, [588](#)
- UserOutputValueAllMask
 - Spinnaker::Camera, [588](#)
- UserSetDefault
 - Spinnaker::Camera, [588](#)
- UserSetDefaultEnums
 - CameraDefs Class, [152](#)
- UserSetFeatureEnable
 - Spinnaker::Camera, [588](#)
- UserSetLoad
 - Spinnaker::Camera, [588](#)
- UserSetSave
 - Spinnaker::Camera, [589](#)
- UserSetSelector
 - Spinnaker::Camera, [589](#)
- UserSetSelectorEnums
 - CameraDefs Class, [153](#)
- V3_3Enable
 - Spinnaker::Camera, [589](#)
- ValueNode, [1076](#)
 - Spinnaker::GenApi::ValueNode, [1077](#)
- ValueNode Class, [362](#)
 - CValueRef, [362](#)
- Varying
 - Types Enums, [360](#)
- Verify
 - IBoolean Interface, [264](#)
- Version_t, [1080](#)
- WaitOnImageEvent
 - Spinnaker::IDataStream, [829](#)
- what
 - Spinnaker::Exception, [764](#)
- WhiteClip
 - Spinnaker::Camera, [589](#)
- WhiteClipSelector
 - Spinnaker::Camera, [589](#)
- WhiteClipSelectorEnums
 - CameraDefs Class, [153](#)
- Width
 - Spinnaker::Camera, [590](#)
- width
 - Spinnaker::Video::H264Option, [802](#)
- WidthMax
 - Spinnaker::Camera, [590](#)
- WO
 - Types Enums, [355](#)
- Write
 - IPort Interface, [308](#)
 - Spinnaker::GenApi::CChunkPort, [632](#)
 - Spinnaker::GenApi::CEventPort, [653](#)
 - Spinnaker::GenApi::CPortImpl, [715](#)
 - Spinnaker::GenApi::CPortWriteList, [718](#)
 - Spinnaker::GenApi::CRegisterPortImpl, [721](#)
 - Spinnaker::GenApi::CTestPortStruct, [728](#)

- Spinnaker::GenApi::PortNode, [1004](#)
- write
 - Spinnaker::GenApi::FileProtocolAdapter, [770](#)
- WritePort
 - Spinnaker::CameraBase, [605](#)
 - Spinnaker::ICameraBase, [812](#)
- WriteRegister
 - Spinnaker::GenApi::CRegisterPortImpl, [722](#)
- xspuIn
 - Spinnaker::GenApi::ODevFileStreamBuf, [996](#)
- Yes
 - Types Enums, [361](#)