
PySpin API Reference

Release 4.2

Teledyne

Feb 26, 2025

CONTENTS:

1	Introduction	1
2	Software Licensing Information	3
3	Event Classes	5
3.1	PySpin.DeviceArrivalEventHandler	5
3.2	PySpin.DeviceEventHandler	5
3.3	PySpin.DeviceRemovalEventHandler	6
3.4	PySpin.EventHandler	6
3.5	PySpin.ImageEventHandler	6
3.6	PySpin.ImageListEventHandler	7
3.7	PySpin.InterfaceArrivalEventHandler	7
3.8	PySpin.InterfaceEventHandler	7
3.9	PySpin.InterfaceRemovalEventHandler	7
3.10	PySpin.LoggingEventHandler	8
3.11	PySpin.LoggingEventDataPtr	8
3.12	PySpin.SystemEventHandler	8
4	PySpin Classes	9
4.1	PySpin.CBasePtr	10
4.2	PySpin.Camera	10
4.3	PySpin.CameraBase	35
4.4	PySpin.CameraList	39
4.5	PySpin.CameraPtr	41
4.6	PySpin.ChannelStatistics	42
4.7	PySpin.ChunkData	42
4.8	PySpin.Image	46
4.9	PySpin.ImageList	55
4.10	PySpin.ImageProcessor	56
4.11	PySpin.ImagePtr	58
4.12	PySpin.ImageUtility	58
4.13	PySpin.ImageUtilityCCM	59
4.14	PySpin.ImageUtilityHeatmap	60
4.15	PySpin.ImageUtilityPolarization	62
4.16	PySpin.ImageUtilityStereo	64
4.17	PySpin.IInterface	66
4.18	PySpin.InterfaceList	67
4.19	PySpin.InterfacePtr	68
4.20	PySpin.PointCloud	68
4.21	PySpin.SpinnakerException	69

4.22	PySpin.SpinVideo	69
4.23	PySpin.System	70
4.24	PySpin.SystemPtr	74
5	QuickSpin classes	75
5.1	PySpin.TransportLayerDevice	75
5.2	PySpin.TransportLayerInterface	77
5.3	PySpin.TransportLayerStream	79
6	PySpin Module	81
6.1	Parameters:	107
6.2	Parameters:	109
6.3	Parameters:	336
6.4	Parameters:	336
6.5	Parameters:	375
6.6	Parameters:	378
6.7	Parameters:	378
	Python Module Index	413
	Index	415

INTRODUCTION

PySpin is a wrapper for Teledyne Spinnaker library.

Teledyne Machine Vision website is located at <https://www.flir.com/iis/machine-vision>.

The PySpin Python extension provides a common software interface to control and acquire images from Teledyne USB 3.0, GigE, and USB 2.0 cameras using the same API.

SOFTWARE LICENSING INFORMATION

Component	License
PySpin	Copyright (c) 2025 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 http://www.emva.org/wp-content/uploads/GenICam_License_20140921.pdf
AdapterList	The Code Project Open License (CPOL) http://www.codeproject.com/info/cpol10.aspx
Boost	Boost Software License http://www.boost.org/users/license.html
FFMPEG	LGPLv2.1 License https://www.ffmpeg.org/legal.html
FreeImage	FreeImage public license http://freeimage.sourceforge.net/freeimage-license.txt
Libusb	LGPLv2. License http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt
Libraw394	LGPLv2.0 License http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt
log4Net	Apache license 2.0 https://logging.apache.org/log4net/license.html
log4Cpp	LGPL License http://log4cpp.sourceforge.net/#license
Work with Bitmaps Faster in C#	The Code Project Open License (CPOL) 1.02 http://www.codeproject.com/info/cpol10.aspx
GUI ListView Improvements	WP:CC_BY-SA License https://goo.gl/a9I9yA

EVENT CLASSES

- *PySpin.DeviceArrivalEventHandler*
- *PySpin.DeviceEventHandler*
- *PySpin.DeviceRemovalEventHandler*
- *PySpin.EventHandler*
- *PySpin.ImageEventHandler*
- *PySpin.ImageListEventHandler*
- *PySpin.InterfaceArrivalEventHandler*
- *PySpin.InterfaceEventHandler*
- *PySpin.InterfaceRemovalEventHandler*
- *PySpin.LoggingEventHandler*
- *PySpin.LoggingEventDataPtr*
- *PySpin.SystemEventHandler*

3.1 PySpin.DeviceArrivalEventHandler

class `PySpin.DeviceArrivalEventHandler`

Proxy of C++ Spinnaker::DeviceArrivalEventHandler class.

OnDeviceArrival (*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.2 PySpin.DeviceEventHandler

class `PySpin.DeviceEventHandler`

Proxy of C++ Spinnaker::DeviceEventHandler class.

GetDeviceEventId(*self*) → uint64_t

GetDeviceEventName(*self*) → gcstring

OnDeviceEvent(*self*, *eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

3.3 PySpin.DeviceRemovalEventHandler

class **PySpin.DeviceRemovalEventHandler**

Proxy of C++ Spinnaker::DeviceRemovalEventHandler class.

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.4 PySpin.EventHandler

class **PySpin.EventHandler**(*args, **kwargs)

Proxy of C++ Spinnaker::EventHandler class.

GetEventPayloadData(*self*) → PyObject *

GetEventPayloadDataSize(*self*) → size_t const

GetEventType(*self*) → Spinnaker::EventType

SetEventType(*self*, *eventType*)

Parameters

eventType (*enum Spinnaker::EventType*)

property thisown

The membership flag

3.5 PySpin.ImageEventHandler

class **PySpin.ImageEventHandler**

Proxy of C++ Spinnaker::ImageEventHandler class.

OnImageEvent(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

property thisown

The membership flag

3.6 PySpin.ImageListEventHandler

class PySpin.ImageListEventHandler

Proxy of C++ Spinnaker::ImageListEventHandler class.

OnImageListEvent(*self*, *imageList*)

Parameters

imageList (*Spinnaker::ImageList*)

property thisown

The membership flag

3.7 PySpin.InterfaceArrivalEventHandler

class PySpin.InterfaceArrivalEventHandler

Proxy of C++ Spinnaker::InterfaceArrivalEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

3.8 PySpin.InterfaceEventHandler

class PySpin.InterfaceEventHandler

Proxy of C++ Spinnaker::InterfaceEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

3.9 PySpin.InterfaceRemovalEventHandler

class PySpin.InterfaceRemovalEventHandler

Proxy of C++ Spinnaker::InterfaceRemovalEventHandler class.

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

3.10 PySpin.LoggingEventHandler

class PySpin.LoggingEventHandler

Proxy of C++ Spinnaker::LoggingEventHandler class.

OnLogEvent(*self*, *eventPtr*)

Parameters

eventPtr (*Spinnaker::LoggingEventDataPtr*)

property thisown

The membership flag

3.11 PySpin.LoggingEventDataPtr

class PySpin.LoggingEventDataPtr(*args)

A reference tracked pointer to the LoggingEvent object.

C++ includes: LoggingEventDataPtr.h

property thisown

The membership flag

3.12 PySpin.SystemEventHandler

class PySpin.SystemEventHandler

Proxy of C++ Spinnaker::SystemEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

PYSPIN CLASSES

- *PySpin.CBasePtr*
- *PySpin.Camera*
- *PySpin.CameraBase*
- *PySpin.CameraList*
- *PySpin.CameraPtr*
- *PySpin.ChannelStatistics*
- *PySpin.ChunkData*
- *PySpin.Image*
- *PySpin.ImageList*
- *PySpin.ImageProcessor*
- *PySpin.ImagePtr*
- *PySpin.ImageUtility*
- *PySpin.ImageUtilityCCM*
- *PySpin.ImageUtilityHeatmap*
- *PySpin.ImageUtilityPolarization*
- *PySpin.ImageUtilityStereo*
- *PySpin.IInterface*
- *PySpin.InterfaceList*
- *PySpin.InterfacePtr*
- *PySpin.PointCloud*
- *PySpin.SpinnakerException*
- *PySpin.SpinVideo*
- *PySpin.System*
- *PySpin.SystemPtr*

4.1 PySpin.CBasePtr

class PySpin.CBasePtr(*args)

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

IsValid(self) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

4.2 PySpin.Camera

class PySpin.Camera(*args, **kwargs)

The camera object class.

C++ includes: Camera.h

property AasRoiEnable

property AasRoiHeight

property AasRoiOffsetX

property AasRoiOffsetY

property AasRoiWidth

property AcquisitionAbort

property AcquisitionArm

property AcquisitionBurstFrameCount

property AcquisitionFrameCount

property AcquisitionFrameRate

property AcquisitionFrameRateEnable

property AcquisitionFrameRatePersistence

property AcquisitionLineRate

property AcquisitionMode

property AcquisitionResultingFrameRate

property AcquisitionStart

property AcquisitionStatus

property AcquisitionStatusSelector

property AcquisitionStop

property AcquisitionTransferFrameRate

property ActionDeviceKey

property ActionGroupKey

property ActionGroupMask

property ActionQueueEmpty

property ActionQueueSize

property ActionSelector

property ActionSignalSize

property ActionUnconditionalMode

property AdaptiveCompressionEnable

property AdcBitDepth

property AutoAlgorithmSelector

property AutoExposureControlLoopDamping

property AutoExposureControlPriority

property AutoExposureEVCompensation

property AutoExposureExposureTimeLowerLimit

property AutoExposureExposureTimeUpperLimit

property AutoExposureGainLowerLimit

property AutoExposureGainUpperLimit

property AutoExposureGreyValueLowerLimit

property AutoExposureGreyValueUpperLimit

property AutoExposureLightingMode

property AutoExposureMeteringMode

property AutoExposureTargetGreyValue

property AutoExposureTargetGreyValueAuto

property BalanceRatio

property BalanceRatioSelector

property BalanceWhiteAuto

property BalanceWhiteAutoDamping

property BalanceWhiteAutoLowerLimit

property BalanceWhiteAutoProfile
property BalanceWhiteAutoUpperLimit
property BinningHorizontal
property BinningHorizontalMode
property BinningSelector
property BinningVertical
property BinningVerticalMode
property BlackLevel
property BlackLevelAuto
property BlackLevelAutoBalance
property BlackLevelClampingEnable
property BlackLevelRaw
property BlackLevelSelector
property BsiFlatFieldCorrectionAuto
property BsiFlatFieldCorrectionAutoDamping
property BsiFlatFieldCorrectionEnable
property BsiFlatFieldCorrectionGain
property BsiFlatFieldCorrectionGainSelector
property BufferedBurstFrameCountMax
property BufferedBurstMode
property ChunkBlackLevel
property ChunkBlackLevelSelector
property ChunkCRC
property ChunkCompressionMode
property ChunkCompressionRatio
property ChunkCounterSelector
property ChunkCounterValue
property ChunkCurrentDataRate
property ChunkEnable
property ChunkEncoderSelector
property ChunkEncoderStatus

property ChunkEncoderValue

property ChunkExposureEndLineStatusAll

property ChunkExposureTime

property ChunkExposureTimeSelector

property ChunkFrameID

property ChunkGain

property ChunkGainSelector

property ChunkHeight

property ChunkImage

property ChunkImageComponent

property ChunkInferenceBoundingBoxResult

property ChunkInferenceConfidence

property ChunkInferenceFrameId

property ChunkInferenceResult

property ChunkLinePitch

property ChunkLineStatusAll

property ChunkModeActive

property ChunkOffsetX

property ChunkOffsetY

property ChunkPartSelector

property ChunkPixelDynamicRangeMax

property ChunkPixelDynamicRangeMin

property ChunkPixelFormat

property ChunkRegionID

property ChunkScan3dAxisMax

property ChunkScan3dAxisMin

property ChunkScan3dCoordinateOffset

property ChunkScan3dCoordinateReferenceSelector

property ChunkScan3dCoordinateReferenceValue

property ChunkScan3dCoordinateScale

property ChunkScan3dCoordinateSelector

property ChunkScan3dCoordinateSystem
property ChunkScan3dCoordinateSystemReference
property ChunkScan3dCoordinateTransformSelector
property ChunkScan3dDistanceUnit
property ChunkScan3dInvalidDataFlag
property ChunkScan3dInvalidDataValue
property ChunkScan3dOutputMode
property ChunkScan3dTransformValue
property ChunkScanLineSelector
property ChunkSelector
property ChunkSequencerSetActive
property ChunkSerialData
property ChunkSerialDataLength
property ChunkSerialReceiveOverflow
property ChunkSourceID
property ChunkStreamChannelID
property ChunkTimerSelector
property ChunkTimerValue
property ChunkTimestamp
property ChunkTimestampLatchValue
property ChunkTransferBlockID
property ChunkTransferQueueCurrentBlockCount
property ChunkTransferStreamID
property ChunkWidth
property ClConfiguration
property ClTimeSlotsCount
property ColorTransformationEnable
property ColorTransformationSelector
property ColorTransformationValue
property ColorTransformationValueSelector
property ComponentActiveCount

property ComponentDestination
property ComponentEnable
property ComponentSelector
property CompressedFrameDropCount
property CompressionSaturationPriority
property ControlPacketsReservedBandwidth
property CounterDelay
property CounterDuration
property CounterEventActivation
property CounterEventSource
property CounterReset
property CounterResetActivation
property CounterResetSource
property CounterSelector
property CounterStatus
property CounterTriggerActivation
property CounterTriggerSource
property CounterValue
property CounterValueAtReset
property CxpConnectionSelector
property CxpConnectionTestErrorCount
property CxpConnectionTestMode
property CxpConnectionTestPacketCount
property CxpLinkConfiguration
property CxpLinkConfigurationPreferred
property CxpLinkConfigurationStatus
property CxpPoCxpAuto
property CxpPoCxpStatus
property CxpPoCxpTripReset
property CxpPoCxpTurnOff
property DecimationHorizontal

property DecimationHorizontalMode
property DecimationSelector
property DecimationVertical
property DecimationVerticalMode
property DefectCorrectStaticEnable
property DefectCorrectionMode
property DefectTableApply
property DefectTableCoordinateX
property DefectTableCoordinateY
property DefectTableFactoryRestore
property DefectTableIndex
property DefectTablePixelCount
property DefectTableSave
property DefectTableSensor
property Deinterlacing
property DeviceCharacterSet
property DeviceClockFrequency
property DeviceClockSelector
property DeviceConnectionSelector
property DeviceConnectionSpeed
property DeviceConnectionStatus
property DeviceEventChannelCount
property DeviceFamilyName
property DeviceFeaturePersistenceEnd
property DeviceFeaturePersistenceStart
property DeviceFirmwareVersion
property DeviceGenCPVersionMajor
property DeviceGenCPVersionMinor
property DeviceID
property DeviceIndicatorMode
property DeviceLinkBandwidthReserve

property DeviceLinkCommandTimeout
property DeviceLinkConnectionCount
property DeviceLinkCurrentThroughput
property DeviceLinkHeartbeatMode
property DeviceLinkHeartbeatTimeout
property DeviceLinkSelector
property DeviceLinkSpeed
property DeviceLinkThroughputLimit
property DeviceLinkThroughputLimitMode
property DeviceManifestEntrySelector
property DeviceManifestPrimaryURL
property DeviceManifestSchemaMajorVersion
property DeviceManifestSchemaMinorVersion
property DeviceManifestSecondaryURL
property DeviceManifestXMLMajorVersion
property DeviceManifestXMLMinorVersion
property DeviceManifestXMLSubMinorVersion
property DeviceManufacturerInfo
property DeviceMaxThroughput
property DeviceModelName
property DevicePowerSupplySelector
property DeviceRegistersCheck
property DeviceRegistersEndianness
property DeviceRegistersStreamingEnd
property DeviceRegistersStreamingStart
property DeviceRegistersValid
property DeviceReset
property DeviceSFNCVersionMajor
property DeviceSFNCVersionMinor
property DeviceSFNCVersionSubMinor
property DeviceScanType

property DeviceSensorChroma
property DeviceSerialNumber
property DeviceSerialPortBaudRate
property DeviceSerialPortSelector
property DeviceStreamChannelCount
property DeviceStreamChannelEndianness
property DeviceStreamChannelLink
property DeviceStreamChannelPacketSize
property DeviceStreamChannelSelector
property DeviceStreamChannelType
property DeviceTLType
property DeviceTLVersionMajor
property DeviceTLVersionMinor
property DeviceTLVersionSubMinor
property DeviceTapGeometry
property DeviceTemperature
property DeviceTemperatureSelector
property DeviceType
property DeviceUptime
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property EncoderDivider
property EncoderMode
property EncoderOutputMode
property EncoderReset
property EncoderResetActivation
property EncoderResetSource
property EncoderSelector
property EncoderSourceA
property EncoderSourceB

property EncoderStatus
property EncoderTimeout
property EncoderValue
property EncoderValueAtReset
property EnumerationCount
property EventAcquisitionEnd
property EventAcquisitionEndFrameID
property EventAcquisitionEndTimestamp
property EventAcquisitionError
property EventAcquisitionErrorFrameID
property EventAcquisitionErrorTimestamp
property EventAcquisitionStart
property EventAcquisitionStartFrameID
property EventAcquisitionStartTimestamp
property EventAcquisitionTransferEnd
property EventAcquisitionTransferEndFrameID
property EventAcquisitionTransferEndTimestamp
property EventAcquisitionTransferStart
property EventAcquisitionTransferStartFrameID
property EventAcquisitionTransferStartTimestamp
property EventAcquisitionTrigger
property EventAcquisitionTriggerFrameID
property EventAcquisitionTriggerTimestamp
property EventActionLate
property EventActionLateFrameID
property EventActionLateTimestamp
property EventCounter0End
property EventCounter0EndFrameID
property EventCounter0EndTimestamp
property EventCounter0Start
property EventCounter0StartFrameID

property EventCounter0StartTimestamp
property EventCounter1End
property EventCounter1EndFrameID
property EventCounter1EndTimestamp
property EventCounter1Start
property EventCounter1StartFrameID
property EventCounter1StartTimestamp
property EventEncoder0Restarted
property EventEncoder0RestartedFrameID
property EventEncoder0RestartedTimestamp
property EventEncoder0Stopped
property EventEncoder0StoppedFrameID
property EventEncoder0StoppedTimestamp
property EventEncoder1Restarted
property EventEncoder1RestartedFrameID
property EventEncoder1RestartedTimestamp
property EventEncoder1Stopped
property EventEncoder1StoppedFrameID
property EventEncoder1StoppedTimestamp
property EventError
property EventErrorCode
property EventErrorFrameID
property EventErrorTimestamp
property EventExposureEnd
property EventExposureEndFrameID
property EventExposureEndTimestamp
property EventExposureStart
property EventExposureStartFrameID
property EventExposureStartTimestamp
property EventFrameBurstEnd
property EventFrameBurstEndFrameID

property EventFrameBurstEndTimestamp
property EventFrameBurstStart
property EventFrameBurstStartFrameID
property EventFrameBurstStartTimestamp
property EventFrameEnd
property EventFrameEndFrameID
property EventFrameEndTimestamp
property EventFrameStart
property EventFrameStartFrameID
property EventFrameStartTimestamp
property EventFrameTransferEnd
property EventFrameTransferEndFrameID
property EventFrameTransferEndTimestamp
property EventFrameTransferStart
property EventFrameTransferStartFrameID
property EventFrameTransferStartTimestamp
property EventFrameTrigger
property EventFrameTriggerFrameID
property EventFrameTriggerTimestamp
property EventLine0AnyEdge
property EventLine0AnyEdgeFrameID
property EventLine0AnyEdgeTimestamp
property EventLine0FallingEdge
property EventLine0FallingEdgeFrameID
property EventLine0FallingEdgeTimestamp
property EventLine0RisingEdge
property EventLine0RisingEdgeFrameID
property EventLine0RisingEdgeTimestamp
property EventLine1AnyEdge
property EventLine1AnyEdgeFrameID
property EventLine1AnyEdgeTimestamp

property EventLine1FallingEdge
property EventLine1FallingEdgeFrameID
property EventLine1FallingEdgeTimestamp
property EventLine1RisingEdge
property EventLine1RisingEdgeFrameID
property EventLine1RisingEdgeTimestamp
property EventLinkSpeedChange
property EventLinkSpeedChangeFrameID
property EventLinkSpeedChangeTimestamp
property EventLinkTrigger0
property EventLinkTrigger0FrameID
property EventLinkTrigger0Timestamp
property EventLinkTrigger1
property EventLinkTrigger1FrameID
property EventLinkTrigger1Timestamp
property EventNotification
property EventSelector
property EventSequencerSetChange
property EventSequencerSetChangeFrameID
property EventSequencerSetChangeTimestamp
property EventSerialData
property EventSerialDataLength
property EventSerialPortReceive
property EventSerialPortReceiveTimestamp
property EventSerialReceiveOverflow
property EventStream0TransferBlockEnd
property EventStream0TransferBlockEndFrameID
property EventStream0TransferBlockEndTimestamp
property EventStream0TransferBlockStart
property EventStream0TransferBlockStartFrameID
property EventStream0TransferBlockStartTimestamp

property EventStream@TransferBlockTrigger
property EventStream@TransferBlockTriggerFrameID
property EventStream@TransferBlockTriggerTimestamp
property EventStream@TransferBurstEnd
property EventStream@TransferBurstEndFrameID
property EventStream@TransferBurstEndTimestamp
property EventStream@TransferBurstStart
property EventStream@TransferBurstStartFrameID
property EventStream@TransferBurstStartTimestamp
property EventStream@TransferEnd
property EventStream@TransferEndFrameID
property EventStream@TransferEndTimestamp
property EventStream@TransferOverflow
property EventStream@TransferOverflowFrameID
property EventStream@TransferOverflowTimestamp
property EventStream@TransferPause
property EventStream@TransferPauseFrameID
property EventStream@TransferPauseTimestamp
property EventStream@TransferResume
property EventStream@TransferResumeFrameID
property EventStream@TransferResumeTimestamp
property EventStream@TransferStart
property EventStream@TransferStartFrameID
property EventStream@TransferStartTimestamp
property EventTest
property EventTestTimestamp
property EventTimer@End
property EventTimer@EndFrameID
property EventTimer@EndTimestamp
property EventTimer@Start
property EventTimer@StartFrameID

property EventTimer0StartTimestamp
property EventTimer1End
property EventTimer1EndFrameID
property EventTimer1EndTimestamp
property EventTimer1Start
property EventTimer1StartFrameID
property EventTimer1StartTimestamp
property ExposureActiveMode
property ExposureAuto
property ExposureMode
property ExposureTime
property ExposureTimeMode
property ExposureTimeSelector
property ExternalVoltageEnable
property ExternalVoltageSelector
property ExternalVoltageValue
property FactoryReset
property FfcEnable
property FfcMode
property FfcUserGain
property FfcUserOffset
property FfcUserTableReset
property FfcUserTableSave
property FfcUserTableXCoordinate
property FileAccessBuffer
property FileAccessLength
property FileAccessOffset
property FileOpenMode
property FileOperationExecute
property FileOperationResult
property FileOperationSelector

property FileOperationStatus

property FileSelector

property FileSize

property Gain

property GainAuto

property GainAutoBalance

property GainConversion

property GainSelector

property Gamma

property GammaEnable

property GevActiveLinkCount

property GevCCP

property GevCurrentDefaultGateway

property GevCurrentIPAddress

property GevCurrentIPConfigurationDHCP

property GevCurrentIPConfigurationLLA

property GevCurrentIPConfigurationPersistentIP

property GevCurrentPhysicalLinkConfiguration

property GevCurrentSubnetMask

property GevDiscoveryAckDelay

property GevFirstURL

property GevGVCPExtendedStatusCodes

property GevGVCPExtendedStatusCodesSelector

property GevGVCPHeartbeatDisable

property GevGVCPPendingAck

property GevGVCPPendingTimeout

property GevGVSPExtendedIDMode

property GevHeartbeatTimeout

property GevIEEE1588

property GevIEEE1588ClockAccuracy

property GevIEEE1588ClockId

property `GevIEEE1588DataSetLatch`
property `GevIEEE1588Mode`
property `GevIEEE1588OffsetFromMasterLatched`
property `GevIEEE1588ParentClockIdLatched`
property `GevIEEE1588Status`
property `GevIEEE1588StatusLatched`
property `GevIPConfigurationStatus`
property `GevInterfaceSelector`
property `GevMACAddress`
property `GevMCDA`
property `GevMCPHostPort`
property `GevMCRC`
property `GevMCSP`
property `GevMCTT`
property `GevNumberOfActiveLinks`
property `GevNumberOfInterfaces`
property `GevPAUSEFrameReception`
property `GevPAUSEFrameTransmission`
property `GevPersistentDefaultGateway`
property `GevPersistentIPAddress`
property `GevPersistentSubnetMask`
property `GevPhysicalLinkConfiguration`
property `GevPhysicalLinkConfigurationCapability`
property `GevPrimaryApplicationIPAddress`
property `GevPrimaryApplicationSocket`
property `GevPrimaryApplicationSwitchoverKey`
property `GevSCCFGAllInTransmission`
property `GevSCCFGExtendedChunkData`
property `GevSCCFGPacketResendDestination`
property `GevSCCFGUnconditionalStreaming`
property `GevSCDA`

property `GevSCPD`

property `GevSCPDDirection`

property `GevSCPHostPort`

property `GevSCPInterfaceIndex`

property `GevSCPSBigEndian`

property `GevSCPSDoNotFragment`

property `GevSCPSFireTestPacket`

property `GevSCPSPacketSize`

property `GevSCSP`

property `GevSCZoneConfigurationLock`

property `GevSCZoneCount`

property `GevSCZoneDirectionAll`

property `GevSecondURL`

property `GevStreamChannelSelector`

property `GevSupportedOption`

property `GevSupportedOptionSelector`

property `GevTimestampTickFrequency`

property `GuiXmlManifestAddress`

property `Height`

property `HeightMax`

property `ImageComponentEnable`

property `ImageComponentSelector`

property `ImageCompressionBitrate`

property `ImageCompressionJPEGFormatOption`

property `ImageCompressionMode`

property `ImageCompressionQuality`

property `ImageCompressionRateOption`

`Init(self)`
void Spinnaker::Camera::Init()

property `IspEnable`

property `LUTEnable`

property LUTIndex
property LUTSelector
property LUTValue
property LUTValueAll
property LargePenalty
property LensShadingCoefficientActiveSet
property LensShadingCorrectionCalibration
property LensShadingCorrectionCalibrationGainLimit
property LensShadingCorrectionCalibrationSetup
property LensShadingCorrectionCalibrationStatus
property LensShadingCorrectionMode
property LensShadingCorrectionStepSize
property LensShadingCorrectionVersion
property LineFilterWidth
property LineFormat
property LineInputFilterSelector
property LineInverter
property LineMode
property LinePitch
property LineSelector
property LineSource
property LineStatus
property LineStatusAll
property LinkErrorCount
property LinkRecoveryCount
property LinkUptime
property LogicBlockLUTInputActivation
property LogicBlockLUTInputSelector
property LogicBlockLUTInputSource
property LogicBlockLUTOutputValue
property LogicBlockLUTOutputValueAll

property LogicBlockLUTRowIndex
property LogicBlockLUTSelector
property LogicBlockSelector
property MaxDataRateThreshold
property MaxDeviceResetTime
property MultiRoiConfigurationInvalidReason
property MultiRoiConfigurationInvalidReasonAll
property MultiRoiEnable
property MultiRoiFeatureEnable
property MultiRoiHeight
property MultiRoiOffsetX
property MultiRoiOffsetY
property MultiRoiSelector
property MultiRoiWidth
property MultiRoiWindows
property NumDirections
property OffsetX
property OffsetY
property PacketResendRequestCount
property PacketResendRequestsDroppedCount
property PauseFrameCount
property PayloadSize
property PixelColorFilter
property PixelDynamicRangeMax
property PixelDynamicRangeMin
property PixelFormat
property PixelFormatInfoID
property PixelFormatInfoSelector
property PixelSize
property PowerSupplyCurrent
property PowerSupplyVoltage

property RegionDestination
property RegionMode
property RegionSelector
property ReverseX
property ReverseY
property RgbTransformLightSource
property Saturation
property SaturationEnable
property Scan3dAxisMax
property Scan3dAxisMin
property Scan3dBaseline
property Scan3dCoordinateOffset
property Scan3dCoordinateReferenceSelector
property Scan3dCoordinateReferenceValue
property Scan3dCoordinateScale
property Scan3dCoordinateSelector
property Scan3dCoordinateSystem
property Scan3dCoordinateSystemReference
property Scan3dCoordinateTransformSelector
property Scan3dDistanceUnit
property Scan3dFocalLength
property Scan3dInvalidDataFlag
property Scan3dInvalidDataValue
property Scan3dOutputMode
property Scan3dPrincipalPointU
property Scan3dPrincipalPointV
property Scan3dTransformValue
property SensorDescription
property SensorDigitizationTaps
property SensorHeight
property SensorShutterMode

property SensorTaps
property SensorWidth
property SequencerConfigurationMode
property SequencerConfigurationReset
property SequencerConfigurationValid
property SequencerFeatureEnable
property SequencerMode
property SequencerPathSelector
property SequencerSetActive
property SequencerSetLoad
property SequencerSetNext
property SequencerSetSave
property SequencerSetSelector
property SequencerSetStart
property SequencerSetValid
property SequencerTriggerActivation
property SequencerTriggerSource
property SerialPortBaudRate
property SerialPortDataBits
property SerialPortParity
property SerialPortSelector
property SerialPortSource
property SerialPortStopBits
property SerialReceiveFramingErrorCount
property SerialReceiveParityErrorCount
property SerialReceiveQueueClear
property SerialReceiveQueueCurrentCharacterCount
property SerialReceiveQueueMaxCharacterCount
property SerialTransmitQueueCurrentCharacterCount
property SerialTransmitQueueMaxCharacterCount
property Sharpening

property SharpeningAuto
property SharpeningEnable
property SharpeningThreshold
property SmallPenalty
property SoftwareSignalPulse
property SoftwareSignalSelector
property SourceCount
property SourceSelector
property StereoHeight
property StereoResolution
property StereoWidth
property TLParamsLocked
property Test0001
property TestEventGenerate
property TestPattern
property TestPatternGeneratorSelector
property TestPendingAck
property TimerDelay
property TimerDuration
property TimerReset
property TimerSelector
property TimerStatus
property TimerTriggerActivation
property TimerTriggerSource
property TimerValue
property Timestamp
property TimestampIncrement
property TimestampLatch
property TimestampLatchValue
property TimestampReset
property TotalDisparity

property TransferAbort
property TransferBlockCount
property TransferBurstCount
property TransferComponentSelector
property TransferControlMode
property TransferOperationMode
property TransferPause
property TransferQueueCurrentBlockCount
property TransferQueueMaxBlockCount
property TransferQueueMode
property TransferQueueOverflowCount
property TransferResume
property TransferSelector
property TransferStart
property TransferStatus
property TransferStatusSelector
property TransferStop
property TransferStreamChannel
property TransferTriggerActivation
property TransferTriggerMode
property TransferTriggerSelector
property TransferTriggerSource
property TransmissionDelay
property TransmissionDelayAverage
property TransmissionDelayMax
property TriggerActivation
property TriggerDelay
property TriggerDivider
property TriggerEventTest
property TriggerMode
property TriggerMultiplier

property TriggerOverlap
property TriggerSelector
property TriggerSoftware
property TriggerSource
property U3VAccessPrivilege
property U3VCPCapability
property U3VCPEIRMAvailable
property U3VCPIIDC2Available
property U3VCPSIRMAvailable
property U3VCurrentSpeed
property U3VMaxAcknowledgeTransferLength
property U3VMaxCommandTransferLength
property U3VMaxDeviceResponseTime
property U3VMessageChannelID
property U3VNumberOfStreamChannels
property U3VVersionMajor
property U3VVersionMinor
property UniquenessRatio
property UserOutputSelector
property UserOutputValue
property UserOutputValueAll
property UserOutputValueAllMask
property UserSetDefault
property UserSetFeatureEnable
property UserSetLoad
property UserSetSave
property UserSetSelector
property V3_3Enable
property WhiteClip
property WhiteClipSelector
property Width

property WidthMax

property WindowSizeH

property WindowSizeW

property aPAUSEMACtrlFramesReceived

property aPAUSEMACtrlFramesTransmitted

property thisown

The membership flag

4.3 PySpin.CameraBase

class PySpin.CameraBase(*args, **kwargs)

The base class for the camera object.

C++ includes: CameraBase.h

BeginAcquisition(self)

void Spinnaker::CameraBase::BeginAcquisition()

Starts the image acquisition engine. The camera must be initialized via a call to Init() before starting an acquisition.

See: Init()

DeInit(self)

void Spinnaker::CameraBase::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: Init()

See: UnregisterEvent(Event & evtToUnregister)

DiscoverMaxPacketSize(self) → unsigned int

unsigned int Spinnaker::CameraBase::DiscoverMaxPacketSize()

Returns the largest packet size that can be safely used on the interface that device is connected to

The maximum packet size returned.

EndAcquisition(self)

void Spinnaker::CameraBase::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: Init()

See: BeginAcquisition()

See: GetNextImage(grabTimeout)

See: Image::Release()

ForceIP(*self*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GenApi::EAccessMode Spinnaker::CameraBase::GetAccessMode() const

Returns the access mode that the software has on the Camera. The camera does not need to be initialized before calling this function.

See: Init()

An enumeration value indicating the access mode

GetActiveNumDataStreams(*self*) → unsigned int

GetBufferOwnership(*self*) → Spinnaker::BufferOwnership

GetDeviceID(*self*) → *gcstring*

GetGuiXml(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetGuiXml() const

Returns the GUI XML that can be passed into the Spinnaker GUI framework

GenICam::gcstring that represents the uncompressed GUI XML file

GetNextImage(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE, *streamIndex*=0) → *ImagePtr*

Parameters

- **grabTimeout** (a 64bit value that represents a timeout in milliseconds)
- **streamIndex** (uint64_t)
- **ImagePtr**
- **Spinnaker::CameraBase::GetNextImage(uint64_t**
- **grabTimeout=EVENT_TIMEOUT_INFINITE**
- **streamID=0)** (uint64_t)
- **This** (Gets the next image that was received by the transport layer.)
- **cameras** (function will block indefinitely until an image arrives. Most)
- **camera** (support one stream so the default streamID is 0 but if a)
- **select** (supports multiple streams the user can input the streamID to)
- **images** (from which stream to grab)
- **See** (EndAcquisition())
- **See**
- **See**
- **Parameters**
- -----
- **grabTimeout**
- **streamID** (The stream to grab the image.)
- **object** (pointer to an Image)

GetNextImageSync(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE) → *ImageList*

Parameters

grabTimeout (*uint64_t*)

GetNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetNodeMap() const

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: Init()

A reference to the INodeMap.

GetNumDataStreams(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::GetNumDataStreams()

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::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().

The number of images that needs to be cleaned up.

GetTLDeviceNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetTLDeviceNodeMap() const

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.

A reference to the INodeMap.

GetTLStreamNodeMap(*self*, *streamIndex*=0) → *INodeMap*

Parameters

- **streamIndex** (*uint64_t*)
- **const** (GenApi::INodeMap& Spinnaker::CameraBase::GetTLStreamNodeMap())
- **XML** (Gets a reference to the node map that is generated from a GenICam)
- **be** (file for the GenTL Stream module. The camera does not need to)
- **map.** (initialized before acquiring this node)
- **INodeMap.** (A reference to the)

GetUniqueID(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetUniqueID()

This returns a unique id string that identifies the camera. This is the camera serial number.

string that uniquely identifies the camera (serial number)

GetUserBufferCount(*self*) → *uint64_t*

GetUserBufferSize(*self*) → *uint64_t*

GetUserBufferTotalSize(*self*) → uint64_t

Init(*self*)

void Spinnaker::CameraBase::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: BeginAcquisition()

See: EndAcquisition()

See: GetNodeMap()

See: GetNextImage()

IsInitialized(*self*) → bool

bool Spinnaker::CameraBase::IsInitialized()

Checks if camera is initialized. This function needs to return true in order to retrieve a valid NodeMap from the GetNodeMap() call.

See: GetNodeMap()

If camera is initialized or not

IsStreaming(*self*) → bool

bool Spinnaker::CameraBase::IsStreaming() const

Returns true if the camera is currently streaming or false if it is not.

See: Init()

returns true if camera is streaming and false otherwise.

IsValid(*self*) → bool

bool Spinnaker::CameraBase::IsValid()

Checks a flag to determine if camera is still valid for use.

If camera is valid or not

Note that CameraPtr and CameraBase both define an IsValid() function. In order to determine the validity of the camera using a CameraPtr, user must first call get() to retrieve the CameraBase object.

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::ImageEventHandler &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (*Spinnaker::GenICam::gcstring const &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**

- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self, mode*)

Parameters

- mode** (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self, pMemBuffers, totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

UnregisterEventHandler(*self, evtHandlerToUnregister*)

Parameters

- evtHandlerToUnregister** (*Spinnaker::EventHandler &*)

property thisown

The membership flag

4.4 PySpin.CameraList

class **PySpin.CameraList**(*args)

Used to hold a list of camera objects.

C++ includes: CameraList.h

Add(*self, camera*)

Parameters

- camera** (*Spinnaker::CameraPtr*)

Append(*self, list*)

Parameters

- **list** (*Spinnaker::CameraList const &*)
- **void**
- **&otherList**) (*Spinnaker::CameraList::Append(CameraList)*)
- **list.** (*Appends a camera list to the current*)
- **Parameters**
- -----

- **otherList** (*The other list to append to this list*)

Clear(*self*)

void Spinnaker::CameraList::Clear()

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: `System:ReleaseInstance()`

GetByDeviceID(*self, deviceID*) → *CameraPtr*

Parameters

deviceID (*std::string*)

GetByIndex(*self, index*) → *CameraPtr*

Parameters

- **index** (*The index at which to retrieve the camera object*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to a camera object at the*)
- **Parameters**
- **-----**
- **index**
- **object.** (*A pointer to an camera*)

GetBySerial(*self, serialNumber*) → *CameraPtr*

Parameters

- **serialNumber** (*The serial number of the camera object to retrieve*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetBySerial(std::string serialNumber)*)
- **number.** (*Returns a pointer to a camera object with the specified serial*)
- **Parameters**
- **-----**
- **serialNumber**
- **object.** (*A pointer to an camera*)

GetSize(*self*) → unsigned int

int Spinnaker::CameraList::GetSize() const

Returns the size of the camera list. The size is the number of Camera objects stored in the list.

An integer that represents the list size.

Remove(*self*, *camera*)

Parameters

camera (*Spinnaker::CameraPtr*)

RemoveByDeviceID(*self*, *deviceID*)

Parameters

deviceID (*std::string*)

RemoveByIndex(*self*, *index*)

Parameters

- **index** (*The index at which to remove the Camera object*)
- **void**
- **index**) (*Spinnaker::CameraList::RemoveByIndex(int)*)
- **reference** (*Removes a camera at "index" and destroys its corresponding*)
- **object.** (*counted*)
- **Parameters**
- -----
- **index**

RemoveBySerial(*self*, *serialNumber*)

Parameters

- **serialNumber** (*The serial number of the Camera object to remove*)
- **void**
- **serialNumber**) (*Spinnaker::CameraList::RemoveBySerial(std::string)*)
- **its** (*Removes a camera using its serial number and destroys*)
- **object.** (*corresponding reference counted*)
- **Parameters**
- -----
- **serialNumber**

property thisown

The membership flag

4.5 PySpin.CameraPtr

class *PySpin.CameraPtr*(*args)

A reference tracked pointer to a camera object.

C++ includes: *CameraPtr.h*

property thisown

The membership flag

4.6 PySpin.ChannelStatistics

class PySpin.ChannelStatistics(*image, channel*)

Class used to store statistics (as properties) for one channel of an image. Properties:

- **channel**: The image channel that the statistics are based on (as an int).
- **range_min**: The smallest possible pixel value.
- **range_max**: The largest possible pixel value.
- **pixel_value_min**: The smallest pixel value in the current channel.
- **pixel_value_max**: The largest pixel value in the current channel.
- **num_pixel_values**: The total number of pixel values in the current channel.
- **pixel_value_mean**: The average pixel value in the current channel.
- **histogram**: NumPy array representing the histogram of the current channel.

property channel

property histogram

property num_pixel_values

property pixel_value_max

property pixel_value_mean

property pixel_value_min

property range_max

property range_min

property thisown

The membership flag

4.7 PySpin.ChunkData

class PySpin.ChunkData(*args)

The chunk data which contains additional information about an image.

C++ includes: ChunkData.h

GetBlackLevel(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetBlackLevel() const

Description: Returns the black level used to capture the image included in the payload. Visibility: Expert

GetCRC(*self*) → int64_t

GetCompressionMode(*self*) → int64_t

GetCompressionRatio(*self*) → float64_t

GetCounterValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetCounterValue() const

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event. Visibility: Expert

GetCurrentDatarate(*self*) → int64_t

GetEnable(*self*) → bool

GetEncoderValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::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. Visibility: Expert

GetExposureEndLineStatusAll(*self*) → int64_t

GetExposureTime(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetExposureTime() const

Description: Returns the exposure time used to capture the image. Visibility: Expert

GetFrameID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetFrameID() const

Description: Returns the unique Identifier of the frame (or image) included in the payload. Visibility: Expert

GetGain(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetGain() const

Description: Returns the gain used to capture the image. Visibility: Expert

GetHeight(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetHeight() const

Description: Returns the Height of the image included in the payload. Visibility: Expert

GetImage(*self*) → int64_t

GetInferenceBoundingBoxResult(*self*) → *InferenceBoundingBoxResult*

GetInferenceConfidence(*self*) → float64_t

GetInferenceFrameId(*self*) → int64_t

GetInferenceResult(*self*) → int64_t

GetLinePitch(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLinePitch() const

Description: Returns the LinePitch of the image included in the payload. Visibility: Expert

GetLineStatusAll(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLineStatusAll() const

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event. Visibility: Expert

GetModeActive(*self*) → bool

GetOffsetX(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetX() const

Description: Returns the OffsetX of the image included in the payload. Visibility: Expert

GetOffsetY(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetY() const

Description: Returns the OffsetY of the image included in the payload. Visibility: Expert

GetPartSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPartSelector() const

Description: Selects the part to access in chunk data in a multipart transmission. Visibility: Expert

GetPixelDynamicRangeMax(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMax() const

Description: Returns the maximum value of dynamic range of the image included in the payload. Visibility: Expert

GetPixelDynamicRangeMin(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMin() const

Description: Returns the minimum value of dynamic range of the image included in the payload. Visibility: Expert

GetScan3dAxisMax(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMax() const

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dAxisMin(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMin() const

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateOffset(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateOffset() const

Description: Returns the Offset for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateReferenceValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateReferenceValue() const

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point. Visibility: Expert

GetScan3dCoordinateScale(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

Description: Returns the Scale for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dInvalidDataFlag(*self*) → bool

GetScan3dInvalidDataValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetScan3dInvalidDataValue() const
Description: Returns the Invalid Data Value used for the image included in the payload. Visibility: Expert

GetScan3dTransformValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const
Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetScanLineSelector() const
Description: Index for vector representation of one chunk value per line in an image. Visibility: Expert

GetSequencerSetActive(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetSequencerSetActive() const
Description: Return the index of the active set of the running sequencer included in the payload. Visibility: Expert

GetSerialData(*self*) → uint8_t *

GetSerialDataLength(*self*) → int64_t

GetSerialReceiveOverflow(*self*) → bool

GetStreamChannelID(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetStreamChannelID() const
Description: Returns identifier of the stream channel used to carry the block. Visibility: Expert

GetTimerValue(*self*) → float64_t
float64_t Spinnaker::ChunkData::GetTimerValue() const
Description: Returns the value of the selected Timer at the time of the FrameStart internal event. Visibility: Expert

GetTimestamp(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTimestamp() const
Description: Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event. Visibility: Expert

GetTimestampLatchValue(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const
Description: Returns the last Timestamp latched with the TimestampLatch command. Visibility: Expert

GetTransferBlockID(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTransferBlockID() const
Description: Returns the unique identifier of the transfer block used to transport the payload. Visibility: Expert

GetTransferQueueCurrentBlockCount(*self*) → int64_t
int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const
Description: Returns the current number of blocks in the transfer queue. Visibility: Expert

GetWidth(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetWidth() const

Description: Returns the Width of the image included in the payload. Visibility: Expert

SetChunks(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap &*)
- **void**
- **&pNodeMap** (*Spinnaker::ChunkData::SetChunks(GenApi::INodeMap)*)

property thisown

The membership flag

4.8 PySpin.Image

class PySpin.**Image**(*args, **kwargs)

The image object class.

C++ includes: Image.h

CheckCRC(*self*) → bool

bool Spinnaker::Image::CheckCRC() const

Checks if the computed checksum matches with chunk data's ImageCRC

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

static Create() → *ImagePtr*

static Create(*image*) → *ImagePtr*

Parameters

- **image** (*Spinnaker::ImagePtr const*)
- **Create**(width
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ImagePtr** (*copied from another*)
- **width** (*or using*)
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **Create**(width

- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **ImagePtr**
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**
- **dataSize** (*size_t*)
- **object** (*Creates a new Image*)
- **constructor** (*either using a default*)
- **ImagePtr**
- **width**
- **height**

:param : :param offset_x: :param offset_y: :param pixel format: :param and a NumPy array containing 8-bit unsigned ints representing the image data: :param (replaces the void* pData argument).:

DeepCopy(*self, pSrcImage*)

Parameters

- **pSrcImage** (*The Image to copy the data from.*)
- **void**
- **pSrcImage** (*Spinnaker::Image::DeepCopy(const ImagePtr)*)
- **operation** (*Performs a deep copy of the Image. After this*)
- **image** (*the*)
- **not** (*contents and member variables will be the same. The Images will*)
- **released.** (*share a buffer. The Image's current buffer will not be*)
- **Parameters**
- -----
- **pSrcImage**

GetBitsPerPixel(*self*) → size_t

size_t Spinnaker::Image::GetBitsPerPixel() const

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.

The number of bits used per pixel.

GetBufferSize(*self*) → size_t

size_t Spinnaker::Image::GetBufferSize() const

Gets the size of the buffer associated with the image in bytes.

The size of the buffer, in bytes.

GetChunkData(*self*) → [*ChunkData*](#)

const ChunkData& Spinnaker::Image::GetChunkData() const

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.

ChunkData interface that provides access to image chunks.

GetChunkLayoutId(*self*) → uint64_t

uint64_t Spinnaker::Image::GetChunkLayoutId() const

Returns the id of the chunk data layout.

uint64_t value representing the id of the chunk data layout.

GetColorProcessing(*self*) → Spinnaker::ColorProcessingAlgorithm

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

GetDataAbsoluteMax(*self*) → float

GetDataAbsoluteMin(*self*) → float

GetFrameID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight(*self*) → size_t

size_t Spinnaker::Image::GetHeight() const

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.

The height in pixels.

GetID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetID() const

Gets a unique ID for this image. Each image in a stream will have a unique ID to help identify it.

The 64 bit unique id for this image.

GetImagePayloadType(*self*) → Spinnaker::ImagePayloadType

GetImageSize(*self*) → size_t

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus(*self*) → Spinnaker::ImageStatus

ImageStatus Spinnaker::Image::GetImageStatus() const

Returns data integrity status of the image returned from GetNextImage()

Returns whether image has any data integrity issues.

static GetImageStatusDescription(*status*) → char const *

Parameters

status (enum Spinnaker::ImageStatus)

GetNumChannels(*self*) → size_t

GetPayloadType(*self*) → size_t

size_t Spinnaker::Image::GetPayloadType() const

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.

Device types specific payload type.

GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums

Spinnaker::PixelFormatEnums Spinnaker::Image::GetPixelFormat() const

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: Convert()

enum value representing the PixelFormat.

GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType

GetPixelFormatName(*self*) → gcstring

GenICam::gcstring Spinnaker::Image::GetPixelFormatName() const

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.

string value representing the PixelFormat.

GetPrivateData(*self*) → void *

void* Spinnaker::Image::GetPrivateData() const

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.

A pointer to the user passed data pointer.

GetStreamIndex(*self*) → uint64_t

GetStride(*self*) → size_t

size_t Spinnaker::Image::GetStride() const

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.

The stride in bytes.

GetTLPayloadType(*self*) → Spinnaker::TLPayloadType

PayloadTypeInfoIDs Spinnaker::Image::GetTLPayloadType() const

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.

Transport Layer specific payload type.

GetTLPixelFormat(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTLPixelFormat() const

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: GetTLPixelFormatNamespace()

Transport Layer specific pixel format.

GetTLPixelFormatNamespace(*self*) → Spinnaker::TLPixelFormatNamespace

PixelFormatNamespaceID Spinnaker::Image::GetTLPixelFormatNamespace() const

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: GetTLPixelFormat()

enum value representing the PixelFormatNamespace.

GetTimeStamp(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTimeStamp() const

Gets the time stamp for the image in nanoseconds.

The time stamp of the image.

GetValidPayloadSize(*self*) → size_t

size_t Spinnaker::Image::GetValidPayloadSize() const

Returns the size of valid data in the image payload. This is the actual amount of data read from the device. A user created image has a payload size of zero. GetBufferSize() returns the total size of bytes allocated for the image.

See: GetBufferSize()

size_t value representing valid payload.

GetWidth(*self*) → size_t

size_t Spinnaker::Image::GetWidth() const

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.

The width in pixels.

GetXOffset(*self*) → size_t

size_t Spinnaker::Image::GetXOffset() const

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.

The x offset in pixels.

GetXPadding(*self*) → size_t

size_t Spinnaker::Image::GetXPadding() const

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.

The x padding in bytes.

GetYOffset(*self*) → size_t

size_t Spinnaker::Image::GetYOffset() const

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.

The y offset in pixels.

GetYPadding(*self*) → size_t

size_t Spinnaker::Image::GetYPadding() const

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.

The y padding in bytes.

HasCRC(*self*) → bool

bool Spinnaker::Image::HasCRC() const

Checks if the image contains ImageCRC checksum from chunk data

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

HasChunkData(*self*) → bool**IsCompressed(*self*)** → bool**IsInUse(*self*)** → bool

bool Spinnaker::Image::IsInUse()

Returns true if the image is still in use by the stream

Returns true if the image is in use and false otherwise.

IsIncomplete(*self*) → bool

bool Spinnaker::Image::IsIncomplete() const

Returns a boolean value indicating if this image was incomplete. An image is marked as incomplete if the transport layer received less data than it requested.

Returns true if image is incomplete, false otherwise.

static Load(*pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) → *ImagePtr*

Parameters

- **pFilename** (*char const **)
- **format** (*enum Spinnaker::ImageFileFormat*)

Release(*self*)

void Spinnaker::Image::Release()

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

Parameters

- **width** (*The width of image in pixels to set.*)
- **height** (*The height of image in pixels to set.*)
- **offsetX** (*The x offset in pixels to set.*)
- **offsetY** (*The y offset in pixels to set.*)
- **pixelFormat** (*Pixel format to set.*)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**)
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*Pointer to the image buffer.*)
- **ResetImage**(*self*
- **width**
- **height**
- **offsetX**
- **offsetY**

- `pixelFormat`
- `pData`
- `dataPayloadType` (*enum Spinnaker::TLPayloadType*)
- `dataSize`
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`
- `dataPayloadType`
- `dataSize` (*size_t*)
- `void`
- `width`
- `height`
- `size_t`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `void`
- `*pData`)
- `object.` (*Sets new dimensions of the image*)
- `Parameters`
- `-----`
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`

Save(*self*, *pFilename*, *format*=*SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT*)

Parameters

- `pFilename` (*Filename to save image with.*)
- `format` (*enum Spinnaker::ImageFileFormat*)
- `Save`(*self*
- `pFilename`

- pOption)
- pFilename
- pOption(*Options to use while saving image.*)
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)
- pFilename
- pOption
- Save(self
- pFilename
- pOption)

- **pFilename**
- **pOption**
- **void**
- ***pFilename** (*Spinnaker::Image::Save(const char) –*
- **&pOption**) (*BMPOption*)
- **specified.** (*Saves the image to the specified file name with the options*)
- **Parameters**
- **-----**
- **pFilename**
- **pOption**

property thisown

The membership flag

4.9 PySpin.ImageList

class `PySpin.ImageList(*args)`

Proxy of C++ Spinnaker::ImageList class.

Add(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

Append(*self*, *list*)

Parameters

list (*Spinnaker::ImageList const &*)

Clear(*self*)

GetByIndex(*self*, *index*) → *ImagePtr*

Parameters

index (*unsigned int*)

GetByPayloadType(*self*, *payloadType*) → *ImagePtr*

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

GetByPixelFormat(*self*, *pixelFormat*) → *ImagePtr*

Parameters

pixelFormat (*enum Spinnaker::PixelFormatEnums*)

GetByStreamIndex(*self*, *streamIndex*) → *ImagePtr*

Parameters

streamIndex (*uint64_t const*)

GetSize(*self*) → *unsigned int*

static Load(*filename*) → *ImageList*

Parameters

filename (*char const **)

Release(*self*)

RemoveByIndex(*self*, *index*)

Parameters

index (*unsigned int*)

RemoveByPayloadType(*self*, *payloadType*)

Parameters

payloadType (*enum Spinnaker::ImagePayloadType const*)

RemoveByPixelFormat(*self*, *pixelFormat*)

Parameters

pixelFormat (*enum Spinnaker::PixelFormatEnums*)

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (*uint64_t const*)

Save(*self*, *filename*)

Parameters

filename (*char const **)

property thisown

The membership flag

4.10 PySpin.ImageProcessor

class `PySpin.ImageProcessor(*args)`

Proxy of C++ `Spinnaker::ImageProcessor` class.

ApplyGamma(*self*, *srcImage*, *gamma*, *applyGammaInverse=False*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **gamma** (*float*)
- **applyGammaInverse** (*bool*)
- **ApplyGamma**(*self*
 - **srcImage**
 - **destImage** (*Spinnaker::ImagePtr &*)
 - **gamma**
 - **applyGammaInverse=False**)
 - **srcImage**
 - **destImage**)

- `gamma`
- `applyGammaInverse`

Convert(*self*, *srcImage*, *destFormat*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `destFormat` (*enum Spinnaker::PixelFormatEnums*)
- `Convert(self`
- `srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)
- `destFormat`)
- `srcImage`
- `destImage`
- `destFormat`
- `Convert(self`
- `srcImageList` (*Spinnaker::ImageList const &*)
- `ImagePtr` (*destFormat*) ->)
- `srcImageList`
- `destFormat`
- `Convert(self`
- `srcImageList`
- `destImage`
- `destFormat`)
- `srcImageList`
- `destImage`
- `destFormat`

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

- `colorAlgorithm` (*enum Spinnaker::ColorProcessingAlgorithm*)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

- `numThreads` (*unsigned int*)

property thisown

The membership flag

4.11 PySpin.ImagePtr

class PySpin.**ImagePtr**(*args)

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.

C++ includes: ImagePtr.h

property thisown

The membership flag

4.12 PySpin.ImageUtility

class PySpin.**ImageUtility**

Proxy of C++ Spinnaker::ImageUtility class.

static CreateNormalized(srcImage, destPixelFormat, src-
DataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)
→ *ImagePtr*

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **destPixelFormat** (enum Spinnaker::PixelFormatEnums const)
- **srcDataRange** (enum Spinnaker::SourceDataRange)
- **CreateNormalized**(srcImage
- **min** (double const)
- **max** (double const)
- **ImagePtr** (srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE
->)
- **srcImage**
- **min**
- **max**
- **srcDataRange**
- **CreateNormalized**(srcImage
- **min**
- **max**
- **destPixelFormat**
- **ImagePtr**
- **srcImage**
- **min**
- **max**
- **destPixelFormat**
- **srcDataRange**

- `CreateNormalized(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `srcDataRange`
- `CreateNormalized(srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange`

static `CreateScaled(srcImage, scalingAlg, scalingFactor) → ImagePtr`

Parameters

- `srcImage (Spinnaker::ImagePtr const &)`
- `scalingAlg (enum Spinnaker::ImageScalingAlgorithm)`
- `scalingFactor (double)`
- `CreateScaled(srcImage`
- `destImage (Spinnaker::ImagePtr &)`
- `scalingAlg`
- `scalingFactor)`
- `srcImage`
- `destImage`
- `scalingAlg`
- `scalingFactor`

property thisown

The membership flag

4.13 PySpin.ImageUtilityCCM

class `PySpin.ImageUtilityCCM`

Proxy of C++ Spinnaker::ImageUtilityCCM class.

```
static ApplicationToString(application) → std::string

    Parameters
        application (Spinnaker::CCMApplication const &)

static ColorSpaceToString(colorSpace) → std::string

    Parameters
        colorSpace (Spinnaker::CCMColorSpace const &)

static ColorTemperatureToString(colorTemperature) → std::string

    Parameters
        colorTemperature (Spinnaker::CCMColorTemperature const &)

static CreateColorCorrected(srcImage, settings) → ImagePtr

    Parameters
        • srcImage (Spinnaker::ImagePtr const &)
        • settings (Spinnaker::CCMSettings const &)
        • CreateColorCorrected(srcImage
        • destImage (Spinnaker::ImagePtr &)
        • settings)
        • srcImage
        • destImage
        • settings

static EncryptColorCorrectionMatrix(ccmMatrixEntries) → std::string

    Parameters
        ccmMatrixEntries (std::string)

static SensorToString(sensor) → std::string

    Parameters
        sensor (Spinnaker::CCMSensor const &)

static TypeToString(type) → std::string

    Parameters
        type (Spinnaker::CCMType const &)

property thisown
    The membership flag
```

4.14 PySpin.ImageUtilityHeatmap

```
class PySpin.ImageUtilityHeatmap
    Proxy of C++ Spinnaker::ImageUtilityHeatmap class.

    static CreateHeatmap(srcImage) → ImagePtr

        Parameters
            • srcImage (Spinnaker::ImagePtr const &)
```


- **CreateHeatmap**(srcImage
- **min** (*float const*)
- **max** (*float const*)
- **lowColor** (*enum Spinnaker::HeatmapColor const*)
- **highColor** (*enum Spinnaker::HeatmapColor const*)
- **doCheckInvalidVal** (*bool const*)
- **ImagePtr** (*invalidVal*) ->)
- **srcImage**
- **min**
- **max**
- **lowColor**
- **highColor**
- **doCheckInvalidVal**
- **invalidVal** (*unsigned int const*)
- **CreateHeatmap**(srcImage
- **destImage**)
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)

static GetHeatmapColorGradient(*currentLowColor, currentHighColor*)

Parameters

- **currentLowColor** (*Spinnaker::HeatmapColor &*)
- **currentHighColor** (*Spinnaker::HeatmapColor &*)

static GetHeatmapRange(*currentLowValue, currentHighValue*)

Parameters

- **currentLowValue** (*unsigned int &*)
- **currentHighValue** (*unsigned int &*)

static SetHeatmapColorGradient(*newLowColor, newHighColor*)

Parameters

- **newLowColor** (*enum Spinnaker::HeatmapColor const*)
- **newHighColor** (*enum Spinnaker::HeatmapColor const*)

static SetHeatmapRange(*newLowValue, newHighValue*)

Parameters

- **newLowValue** (*unsigned int const*)
- **newHighValue** (*unsigned int const*)

property thisown

The membership flag

4.15 PySpin.ImageUtilityPolarization

class PySpin.ImageUtilityPolarization

Proxy of C++ Spinnaker::ImageUtilityPolarization class.

static CreateAolp(*srcImage*, *colorProcessingAlg*=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateAolp**(**srcImage**
- **destAolpImg** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destAolpImg**
- **colorProcessingAlg**

static CreateDolp(*srcImage*, *colorProcessingAlg*=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateDolp**(**srcImage**
- **destDolpImage** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destDolpImage**
- **colorProcessingAlg**

static CreateGlareReduced(*srcImage*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **CreateGlareReduced**(**srcImage**
- **destGlareReducedImage**)
- **srcImage**
- **destGlareReducedImage** (*Spinnaker::ImagePtr &*)

```
static CreateStokesS0(srcImage, colorProcessin-  
                    gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                    → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS0**(**srcImage**
- **destStokesS0Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS0Image**
- **colorProcessingAlg**

```
static CreateStokesS1(srcImage, colorProcessin-  
                    gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                    → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS1**(**srcImage**
- **destStokesS1Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS1Image**
- **colorProcessingAlg**

```
static CreateStokesS2(srcImage, colorProcessin-  
                    gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                    → ImagePtr
```

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **colorProcessingAlg** (*enum Spinnaker::ColorProcessingAlgorithm const*)
- **CreateStokesS2**(**srcImage**
- **destStokesS2Image** (*Spinnaker::ImagePtr &*)
- **colorProcessingAlg**=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- **srcImage**
- **destStokesS2Image**
- **colorProcessingAlg**

static `ExtractPolarQuadrant`(*srcImage*, *desiredQuadrant*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `desiredQuadrant` (*enum Spinnaker::PolarizationQuadrant const*)
- `ExtractPolarQuadrant`(`srcImage`
- `destQuadImage` (*Spinnaker::ImagePtr &*)
- `desiredQuadrant`)
- `srcImage`
- `destQuadImage`
- `desiredQuadrant`

property `thisown`

The membership flag

4.16 PySpin.ImageUtilityStereo

class `PySpin.ImageUtilityStereo`

Proxy of C++ `Spinnaker::ImageUtilityStereo` class.

static `Compute3DPointFromPixel`(*disparity*, *stereoCameraParameters*, *stereo3DPoint*) → `bool`

Parameters

- `disparity` (*uint16_t const*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `stereo3DPoint` (*Spinnaker::Stereo3DPoint &*)

static `ComputeDistanceBetweenPoints`(*disparityImage*, *stereoParam*, *imagePixel1*, *imagePixel2*) → `PyObject *`

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoParam` (*Spinnaker::StereoCameraParameters const &*)
- `imagePixel1` (*Spinnaker::ImagePixel const &*)
- `imagePixel2` (*Spinnaker::ImagePixel const &*)

static `ComputeDistanceToPoint`(*disparityImage*, *stereoParam*, *imagePixel*) → `PyObject *`

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoParam` (*Spinnaker::StereoCameraParameters const &*)
- `imagePixel` (*Spinnaker::ImagePixel const &*)

static `ComputePointCloud`(*disparityImage*, *rectifiedImage*, *pointCloudParameters*, *stereoCameraParameters*) → *PointCloud*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `rectifiedImage` (*Spinnaker::ImagePtr const &*)
- `pointCloudParameters` (*Spinnaker::PointCloudParameters const &*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `ComputePointCloud`(`disparityImage`
`rectifiedImage`
`pointCloudParameters`
`stereoCameraParameters`
`pointCloud`)
- `disparityImage`
- `rectifiedImage`
- `pointCloudParameters`
- `stereoCameraParameters`
- `pointCloud` (*Spinnaker::PointCloud &*)

static `CreateDepthImage`(*disparityImage, stereoCameraParameters, invalidDepthVal, depth_range_list*)
→ *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `invalidDepthVal` (*uint16_t const*)
- `depth_range_list` (*PyObject **)
- `CreateDepthImage`(`disparityImage`
`stereoCameraParameters`
`invalidDepthVal`
`depthImage` (*Spinnaker::ImagePtr &*)
`depth_range_list`)
- `disparityImage`
- `stereoCameraParameters`
- `invalidDepthVal`
- `depthImage`
- `depth_range_list`

static `FilterSpeckles`(*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*) → *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `maxSpeckleSize` (*int const*)

- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static FilterSpecklesFromImage (*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor, invalidDataValue*)

Parameters

- **disparityImage** (*Spinnaker::ImagePtr &*)
- **maxSpeckleSize** (*int const*)
- **speckleThreshold** (*int const*)
- **disparityScaleFactor** (*float const*)
- **invalidDataValue** (*float const*)

static IsStereoCamera (*pCamera*) → *bool*

Parameters

pCamera (*Spinnaker::CameraPtr*)

property maxDepthThresholdInMeter

property maxDepthThresholdInMm

property thisown

The membership flag

4.17 PySpin.IInterface

class PySpin.IInterface (**args, **kwargs*)

Proxy of C++ Spinnaker::IInterface class.

GetCameras (*self, updateCameras=True*) → *CameraList*

Parameters

updateCameras (*bool*)

GetTLNodeMap (*self*) → *INodeMap*

IsCameraInUse (*self*) → *bool*

IsValid (*self*) → *bool*

RegisterEventHandler (*self, evtHandlerToRegister*)

Parameters

evtHandlerToRegister (*Spinnaker::EventHandler &*)

SendActionCommand (*self, deviceKey, groupKey, groupMask, actionTime=0, requestAck=False, pResultSize=None, results=0*)

Parameters

- **deviceKey** (*unsigned int*)
- **groupKey** (*unsigned int*)

- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)

property TLInterface

UnregisterEventHandler (*self, evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

UpdateCameras (*self*) → *bool*

property thisown

The membership flag

4.18 PySpin.InterfaceList

class **PySpin.InterfaceList** (**args*)

A list of the available interfaces on the system.

C++ includes: InterfaceList.h

Add (*self, iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

Append (*self, list*)

Parameters

list (*Spinnaker::InterfaceList const **)

Clear (*self*)

void *Spinnaker::InterfaceList::Clear()*

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.

GetByIndex (*self, index*) → *InterfacePtr*

Parameters

- **index** (*The index at which to retrieve the Interface object*)
- **const** (*InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to an Interface object at the*)
- **Parameters**
- -----
- **index**
- **object**. (*A pointer to an Interface*)

GetByInterfaceID(*self*, *interfaceID*) → *InterfacePtr*

Parameters

interfaceID (*std::string*)

GetSize(*self*) → unsigned int

int Spinnaker::InterfaceList::GetSize() const

Returns the size of the interface list. The size is the number of Interface objects stored in the list.

An integer that represents the list size.

Remove(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

4.19 PySpin.InterfacePtr

class PySpin.**InterfacePtr**(*args)

A reference tracked pointer to the interface object.

C++ includes: InterfacePtr.h

property thisown

The membership flag

4.20 PySpin.PointCloud

class PySpin.**PointCloud**(*args)

Proxy of C++ Spinnaker::PointCloud class.

AddPoint(*self*, *point*)

Parameters

point (*Spinnaker::Stereo3DPoint const*)

GetNumPoints(*self*) → size_t

GetPoint(*self*, *index*) → *Stereo3DPoint*

Parameters

index (*unsigned int const*)

GetPointCloudData(*self*) → Spinnaker::IPointCloud::PointCloudData *

LoadPointCloudFromPly(*self*, *filename*)

Parameters

filename (*std::string const &*)

PrintPoints(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)

SavePointCloudAsPly(*self*, *arg0*)

Parameters

arg0 (*std::string const &*)

property thisown

The membership flag

4.21 PySpin.SpinnakerException

class PySpin.SpinnakerException

Exception class for the PySpin module. This class has these attributes: message, errorcode, fullmessage

errorcode = 0

fullmessage = ''

message = ''

4.22 PySpin.SpinVideo

class PySpin.SpinVideo

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

C++ includes: SpinVideo.h

Append(*self*, *pImage*)

Parameters

- **pImage** (*The image to append.*)
- **virtual**
- **pImage** (*void Spinnaker::Video::SpinVideo::Append(ImagePtr)*)
- **file.** (*Append an image to the AVI/MP4*)
- **Parameters**
- -----
- **pImage**

Close(*self*)

virtual void Spinnaker::Video::SpinVideo::Close()

Close the AVI/MP4 file.

See: Open()

Open(*self*, *pFileName*, *pOption*)

Parameters

- **pFileName** (*The filename of the MP4 file.*)
- **pOption** (*H264 options to apply to the MP4 file.*)
- **Open**(*self*
- **pFileName**

- `pOption`)
- `pFileName`
- `pOption`
- `Open(self`
- `pFileName`
- `pOption`)
- `pFileName`
- `pOption`
- `void (virtual)`
- `*pFileName (Spinnaker::Video::SpinVideo::Open(const char) –`
- `Video::H264Option`
- `&pOption)`
- *The (Open an H264 MP4 file in preparation for writing Images to disk.*
)
- *automatically (size of MP4 files is limited to 2GB. The filenames*
are)
- *specified. (generated using the filename)*
- **Parameters**
- `-----`
- `pFileName`
- `pOption`
- *See ([H264Option](#))*
- *See*

`SetMaximumFileSize(self, size)`

Parameters

`size (unsigned int)`

property thisown

The membership flag

4.23 PySpin.System

class `PySpin.System(*args, **kwargs)`

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

C++ includes: `System.h`

GetCameras(`self, updateInterfaces=True, updateCameras=True`) → *CameraList*

Parameters

- **updateInterfaces** (*Determines whether or not `updateInterfaceList()` is*
- **updateCameras** (*Determines whether or not `UpdateCameras()` is called*)

- **CameraList**
- **updateInterfaces=true** (*Spinnaker::System::GetCameras(bool)*)
- **bool**
- **updateCameras=true**)
- **call** (*Returns a list of cameras that are available on the system. This*)
- **interfaces.** (*returns both GigE Vision and Usb3 Vision cameras from all*)
- **It** (*The camera list object will reference count the cameras it returns.*)
- **before** (*is important that the camera list is destroyed or is cleared*)
- **system->** (*calling system-> ReleaseInstance() or else the call to*
- **a** (*ReleaseInstance() will result in an error message thrown that*
- **held.** (*reference to the camera is still*)
- **See** (*CameraList::Clear()*)
- **See**
- **Parameters**
- -----
- **updateInterfaces**
- **system** (*before getting cameras from available interfaces on the*)
- **updateCameras**
- **system**
- **cameras.** (*An CameraList object that contains a list of all*)

static **GetInstance()** → *SystemPtr*

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

- **updateInterface** (*Determines whether or not UpdateInterfaceList() is*
- **Spinnaker::System::GetInterfaces**(*bool* (*InterfaceList*))
- **updateInterface=true**)
- **call** (*Returns a list of interfaces available on the system. This*)
- **interfaces.** (*An InterfaceList object that contains a list of all*)
- **Parameters**
- -----
- **updateInterface**
- **interfaces** (*called before getting available*)
- **interfaces.**

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

SpinnakerLogLevel Spinnaker::System::GetLoggingEventPriorityLevel()

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: SpinnakerLogLevel

Level The threshold level

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → bool

bool Spinnaker::System::IsInUse()

Checks if the system is in use by any interface or camera objects.

Returns true if the system is in use and false otherwise.

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::EventHandler &*)
- **updateInterface** (*bool*)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

ReleaseInstance(*self*)

void Spinnaker::System::ReleaseInstance()

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: Error

See: GetInstance()

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*Spinnaker::System::SendActionCommand(unsigned int)*)
- **groupKey** (*int*)
- **groupMask** (*unsigned int*)

- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)
- **void**
- **deviceKey**
- **unsigned**
- **groupKey**
- **groupMask**
- **actionTime=0** (*unsigned long long*)

:param : :param unsigned int *pResultSize=0: :param ActionCommandResult results[]=NULL): :param Broadcast an Action Command to all devices on system: :param Parameters: :param ———: :param deviceKey: :type deviceKey: The Action Command's device key :param groupKey: :type groupKey: The Action Command's group key :param groupMask: :type groupMask: The Action Command's group mask :param actionTime: :type actionTime: (Optional) Time when to assert a future action. Zero :param means immediate action.: :param pResultSize: :type pResultSize: (Optional) The number of results in the results array. :param The value passed should be equal to the expected number of devices: :param that acknowledge the command. Returns the number of received results.: :param results: :type results: (Optional) An Array with *pResultSize elements to hold the :param action command result status. The buffer is filled starting from index: :param 0. If received results are less than expected number of devices that: :param acknowledge the command: :param remaining results are not changed. If: :param received results are more than expected number of devices that: :param acknowledge the command: :param extra results are ignored and not appended to: :param array. This parameter is ignored if pResultSize is 0. Thus this: :param parameter can be NULL if pResultSize is 0 or NULL.:

SetLoggingEventPriorityLevel(*self, level*)

Parameters

- **level** (*enum Spinnaker::SpinnakerLogLevel*)
- **void**
- **Spinnaker::System::SetLoggingEventPriorityLevel**(*SpinnakerLogLevel*
- *level*)
- **events** (*Sets a threshold priority level for logging event. Logging*)
- **callbacks.** (*below such level will not trigger*)
- **logging** (*Spinnaker uses five levels of*)
- **intervention.** (*Warning - failures that are recoverable without user*)
- **intervention.**
- **removal** (*Notice - information about events such as camera arrival and*)

:param : :param initialization and deinitialization: :param starting and stopping image: :param acquisition: :param and feature modification.: :param Info - information about recurring events that are generated regularly: :param such as information on individual images.: :param Debug - information that can be used to troubleshoot the system.: :param See: :type See: SpinnakerLogLevel :param Parameters: :param ———: :param level: :type level: The threshold level

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

- **updateInterfaces** (*bool*)
- **bool**
- **updateInterfaces=true**) (*Spinnaker::System::UpdateCameras(bool)*)
- **that** (*Updates the list of cameras on the system. Note*)
- **each** (*System::GetCameras() internally calls UpdateCameras() for*)
- **the** (*interface it enumerates. If the list changed between this call and*)
- **true** (*last time UpdateCameras was called then the return value will be*)

:param : :param otherwise it is false.: :param See: :type See: GetCameras() :param Parameters: :param
———: :param updateInterfaces: :type updateInterfaces: Determines whether or not UpdateInter-
faceList() is :param called before updating cameras for available interfaces on the system: :param True
if cameras changed on interface and false otherwise.:

UpdateInterfaceList(*self*)

property thisown

The membership flag

4.24 PySpin.SystemPtr

class PySpin.**SystemPtr**(*args)

A reference tracked pointer to a system object.

C++ includes: SystemPtr.h

property thisown

The membership flag

QUICKSPIN CLASSES

- *PySpin.TransportLayerDevice*
- *PySpin.TransportLayerInterface*
- *PySpin.TransportLayerStream*

5.1 PySpin.TransportLayerDevice

class `PySpin.TransportLayerDevice`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerDevice.h`

property `DeviceAccessStatus`

property `DeviceBootloaderVersion`

property `DeviceCurrentSpeed`

property `DeviceDisplayName`

property `DeviceDriverVersion`

property `DeviceEndiannessMechanism`

property `DeviceID`

property `DeviceInstanceId`

property `DeviceIsUpdater`

property `DeviceLinkSpeed`

property `DeviceLocation`

property `DeviceModelName`

property `DeviceMulticastMonitorMode`

property `DevicePortId`

property `DeviceReset`

property DeviceSerialNumber
property DeviceType
property DeviceU3VProtocol
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property GUIXMLLocation
property GUIXMLPath
property GenICamXMLLocation
property GenICamXMLPath
property GevCCP
property GevDeviceAutoForceIP
property GevDeviceDiscoverMaximumPacketSize
property GevDeviceForceGateway
property GevDeviceForceIP
property GevDeviceForceIPAddress
property GevDeviceForceSubnetMask
property GevDeviceGateway
property GevDeviceIPAddress
property GevDeviceIsWrongSubnet
property GevDeviceMACAddress
property GevDeviceMaximumPacketSize
property GevDeviceMaximumRetryCount
property GevDeviceModeIsBigEndian
property GevDevicePort
property GevDeviceReadAndWriteTimeout
property GevDeviceSubnetMask
property GevVersionMajor
property GevVersionMinor
property StreamID
property StreamSelector
property thisown
 The membership flag

5.2 PySpin.TransportLayerInterface

class `PySpin.TransportLayerInterface`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerInterface.h`

property `ActionCommand`

property `DeviceAccessStatus`

property `DeviceCount`

property `DeviceID`

property `DeviceModelName`

property `DeviceSelector`

property `DeviceSerialNumber`

property `DeviceUnlock`

property `DeviceUpdateList`

property `DeviceVendorName`

property `FLIRFilterDriverStatus`

property `GevActionAckRequired`

property `GevActionDeviceKey`

property `GevActionGroupKey`

property `GevActionGroupMask`

property `GevActionTime`

property `GevDeviceAutoForceIP`

property `GevDeviceDisableDiscovery`

property `GevDeviceDiscoveryEnabled`

property `GevDeviceEnableDiscovery`

property `GevDeviceForceGateway`

property `GevDeviceForceIP`

property `GevDeviceForceIPAddress`

property `GevDeviceForceSubnetMask`

property `GevDeviceGateway`

property `GevDeviceIPAddress`

property `GevDeviceMACAddress`

property `GevDeviceSubnetMask`
property `GevInterfaceGateway`
property `GevInterfaceGatewaySelector`
property `GevInterfaceIsIPConflict`
property `GevInterfaceMACAddress`
property `GevInterfaceMTU`
property `GevInterfaceReceiveLinkSpeed`
property `GevInterfaceSubnetIPAddress`
property `GevInterfaceSubnetMask`
property `GevInterfaceSubnetSelector`
property `GevInterfaceTransmitLinkSpeed`
property `HostAdapterDriverVersion`
property `HostAdapterName`
property `HostAdapterVendor`
property `IncompatibleDeviceCount`
property `IncompatibleDeviceID`
property `IncompatibleDeviceModelName`
property `IncompatibleDeviceSelector`
property `IncompatibleDeviceVendorName`
property `IncompatibleGevDeviceIPAddress`
property `IncompatibleGevDeviceMACAddress`
property `IncompatibleGevDeviceSubnetMask`
property `InterfaceDisplayName`
property `InterfaceID`
property `InterfaceType`
property `POEStatus`
property `TeledyneGigeVisionFilterDriverStatus`
property `thisown`

The membership flag

5.3 PySpin.TransportLayerStream

class `PySpin.TransportLayerStream`(*nodeMapTLDevice*)

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

C++ includes: `TransportLayerStream.h`

property `StreamAnnounceBufferMinimum`

property `StreamAnnouncedBufferCount`

property `StreamBlockTransferSize`

property `StreamBlocksProcessingTimeLast`

property `StreamBlocksProcessingTimeMax`

property `StreamBlocksProcessingTimeMin`

property `StreamBlocksReceptionTimeLast`

property `StreamBlocksReceptionTimeMax`

property `StreamBlocksReceptionTimeMin`

property `StreamBufferAlignment`

property `StreamBufferCountManual`

property `StreamBufferCountMax`

property `StreamBufferCountMode`

property `StreamBufferCountResult`

property `StreamBufferHandlingMode`

property `StreamCRCCheckEnable`

property `StreamChunkCountMaximum`

property `StreamDeliveredFrameCount`

property `StreamDroppedFrameCount`

property `StreamID`

property `StreamIncompleteFrameCount`

property `StreamInputBufferCount`

property `StreamIsGrabbing`

property `StreamLostFrameCount`

property `StreamMissedPacketCount`

property `StreamMode`

property `StreamOutputBufferCount`

property `StreamPacketResendEnable`
property `StreamPacketResendMaxRequests`
property `StreamPacketResendReceivedPacketCount`
property `StreamPacketResendRequestCount`
property `StreamPacketResendRequestTimeoutCount`
property `StreamPacketResendRequestedPacketCount`
property `StreamPacketResendTimeout`
property `StreamPacketsDuplicatedCount`
property `StreamPacketsNotYetAvailableCount`
property `StreamPacketsPerFrameCount`
property `StreamPacketsTemporarilyUnavailableCount`
property `StreamPacketsTimeoutCount`
property `StreamPacketsUnavailableCount`
property `StreamReceivedFrameCount`
property `StreamReceivedPacketCount`
property `StreamStartedFrameCount`
property `StreamType`
property `thisown`
 The membership flag

PYSPIN MODULE

class PySpin.PySpin.AVIOption

Bases: object

Options for saving AVI files.

C++ includes: SpinVideoDefs.h

property frameRate

property height

property reserved

property thisown

The membership flag

property width

class PySpin.PySpin.ActionCommandResult

Bases: object

Action Command Result

C++ includes: SpinnakerDefs.h

property DeviceAddress

property Status

property thisown

The membership flag

class PySpin.PySpin.BMPOption

Bases: object

Options for saving Bitmap image.

C++ includes: SpinnakerDefs.h

property indexedColor_8bit

property reserved

property thisown

The membership flag

class PySpin.PySpin.**BooleanNode**(*args, **kwargs)

Bases: *IBoolean*, *ValueNode*

Interface for string properties.

C++ includes: BooleanNode.h

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → bool

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **bool**
- **Verify=false** (*Spinnaker::GenApi::BooleanNode::GetValue(bool)*)
- **bool**
- **const** (*IgnoreCache=false*)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked.** (*is always*)
- **IgnoreCache**
- **false).**
- **read.** (*The value*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::BooleanNode::SetReference(INode (virtual void)**
- ***pBase)**
- **Value** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*The value to set.*)
- **Verify** (*Enables AccessMode and Range verification (default = true).*)
- **void**
- **Value**
- **Verify=true)** (*bool*)
- **value** (*Set node*)
- **Parameters**

- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.PySpin.CBasePtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

IsValid(self) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.PySpin.CBooleanPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

GetAlias(self) → *Inode*

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → *Inode*

GetChildren(self, LinkType=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* *const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *bool*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (enum *Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum *Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*bool*)
- **Verify** (*bool*)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.CCMSettings*

Bases: *object*

Proxy of C++ *Spinnaker::CCMSettings* class.

property Application

property ColorSpace

property ColorTemperature

property CustomCCMCode

property Sensor

property Type

property thisown

The membership flag

class PySpin.PySpin.CCategoryPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(self) → Spinnaker::GenApi::EAccessMode

GetAlias(self) → *INode*

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → *INode*

GetChildren(self, LinkType=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(self) → *gcstring*

GetEventID(self) → *gcstring*

GetFeatures(self)

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.PySpin.CCommandPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

Execute(*self*, *Verify*=True)

Parameters

Verify (bool)

FromString(*self*, *ValueStr*, *Verify*=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetName(*self*, *FullQualified*=*False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (Spinnaker::GenICam::gcstring const &)
- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **AttributeStr** (Spinnaker::GenICam::gcstring &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsDone(*self*, *Verify=True*) → *bool*

Parameters

Verify (*bool*)

IsFeature(*self*) → *bool*

IsSelector(*self*) → *bool*

IsStreamable(*self*) → *bool*

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**CDeviceInfoPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetDeviceVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetGenApiVersion(*self*, *Version*, *Build*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t* &)
- **Build** (*uint16_t* &)

GetModelName(*self*) → *gcstring*

GetProductGuid(*self*) → *gcstring*

GetSchemaVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetStandardNameSpace(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetVendorName(*self*) → *gcstring*

GetVersionGuid(*self*) → *gcstring*

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.PySpin.**CEnumEntryPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode*

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → *Spinnaker::GenApi::ECachingMode*

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType=ctReadingChildren*)

Parameters

LinkType (*enum Spinnaker::GenApi::ELinkType*)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t &*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetNumericValue(*self*) → double

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t &*)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* const &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSymbolic(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetValue(*self*) → int64_t

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (enum *Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum *Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsSelfClearing(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.PySpin.CEnumerationPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify*=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=ctReadingChildren)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetCurrentEntry(*self*, *Verify*=False, *IgnoreCache*=False) → *IEnumEntry*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEntries(*self*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

IntValue (*int64_t* *const*)

GetEntryByName(*self*, *Symbolic*) → *IEnumEntry*

Parameters

Symbolic (*Spinnaker::GenICam::gcstring const &*)

GetEventID(*self*) → *gcstring*

GetIntValue(*self*, *Verify=False*, *IgnoreCache=False*) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t &*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t &*)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSymbolics(*self*, *Symbolics*)

Parameters

Symbolics (*Spinnaker::GenApi::StringList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsFeature(*self*) → *bool*

IsSelector(*self*) → *bool*

IsStreamable(*self*) → *bool*

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetIntValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*int64_t*)

- **Verify** (*bool*)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode* *)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class `PySpin.PySpin.CFeatureBag`

Bases: *IPersistScript*

Bag holding streamable features of a nodetree.

C++ includes: Persistence.h

GetFeatureBagHandle(*self*) → void *

void* Spinnaker::GenApi::CFeatureBag::GetFeatureBagHandle()

LoadFromBag(*self*, *pNodeMap*, *Verify=True*, *pErrorList=None*) → bool

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap* *)
- **Verify** (*bool*)
- **pErrorList** (*Spinnaker::GenICam::gcstring_vector* *)
- ***pNodeMap** (*bool* *Spinnaker::GenApi::CFeatureBag::LoadFromBag*(*INodeMap*)

:param : :param bool Verify=true: :param GenICam::gcstring_vector *pErrorList=NULL): :param Loads the features from the bag to the node tree: :param Parameters: :param ———: :param pNodeMap: :type pNodeMap: The node map :param Verify: :type Verify: If true, all streamable features are read back :param pErrorList: :type pErrorList: If an error occurs during loading the error message is :param stored in the list and the loading continues: :param For Verify=true the list of names in the feature bag is replayed: :param again. If a node is a selector it's value is set to the value from the: :param feature bag If not the value is read from the camera and compared with: :param the value from the feature bag.:

PersistFeature(*self*, *item*)

Parameters

- **item** (*Spinnaker::GenApi::IValue* &)
- **Spinnaker::GenApi::CFeatureBag::PersistFeature**(*IValue* (*virtual void*)
- **&item**)
- **feature** (*Stores a*)

SetInfo(*self*, *Info*)

Parameters

- **Info** (*Spinnaker::GenICam::gcstring* &)

- **Spinnaker::GenApi::CFeatureBag::SetInfo**(GenICam::gcstring (virtual void)
- **&Info**)
- **map** (sets information about the node)

StoreToBag(self, pNodeMap, MaxNumPersistSkriptEntries=-1) → int64_t

Parameters

- **pNodeMap** (Spinnaker::GenApi::INodeMap *)
- **MaxNumPersistSkriptEntries** (int const)
- ***pNodeMap** (int64_t Spinnaker::GenApi::CFeatureBag::StoreToBag(INodeMap) —

:param : :param const int MaxNumPersistSkriptEntries=-1): :param Stores the streamable nodes to this feature bag.: :param Parameters: :param —: :param pNodeMap: :type pNodeMap: The node map to persist :param MaxNumPersistSkriptEntries: :type MaxNumPersistSkriptEntries: The max number of entries in the :param container; -1 means unlimited: :param number of entries in the bag:

property thisown

The membership flag

class PySpin.PySpin.CFloatPtr(*args)

Bases: _SWIG_CFltPtr

SmartPointer for IFloat interface pointer

C++ includes: Pointer.h

GetEnumAlias(self) → IEnumeration

IEnumeration* Spinnaker::GenApi::CFloatPtr::GetEnumAlias()

gets the interface of an enum alias node.

GetIntAlias(self) → IInteger

IInteger* Spinnaker::GenApi::CFloatPtr::GetIntAlias()

gets the interface of an integer alias node.

property thisown

The membership flag

class PySpin.PySpin.CIntegerPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(self, hCallback) → bool

Parameters

- **hCallback** (Spinnaker::GenApi::CallbackHandleType)

FromString(self, ValueStr, Verify=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetInc(*self*) → *int64_t*

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

GetListOfValidValues(*self*, *bounded*=*True*) → *int64_autovector_t*

Parameters

bounded (*bool*)

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetMax(*self*) → *int64_t*

GetMin(*self*) → *int64_t*

GetName(*self*, *FullQualified*=*False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* const &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetRepresentation(*self*) → *Spinnaker::GenApi::ERepresentation*

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetUnit(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeMax(*self*, *Value*)

Parameters

Value (*int64_t*)

ImposeMin(*self*, *Value*)

Parameters

Value (*int64_t*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

SetValue(*self*, *Value*, *Verify*=True)

Parameters

- **Value** (int64_t)
- **Verify** (bool)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.PySpin.CNodeMapDynPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

ClearAllNodes(*self*)

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (IPort *)
- **PortName** (Spinnaker::GenICam::gcstring const &)
- **Connect**(self
- **bool** (pPort) ->)
- **pPort**

ExtractIndependentSubtree(*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *ExtractedSubtree*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **SubTreeRootNodeName** (*Spinnaker::GenICam::gcstring const &*)
- **ExtractedSubtree** (*Spinnaker::GenICam::gcstring &*)

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring const &*)

GetNodes(*self*)

GetNumNodes(*self*) → *uint64_t*

GetSupportedSchemaVersions(*self*)

InvalidateNodes(*self*)

IsValid(*self*) → *bool*

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* *const throw ()* true if the pointer is valid

LoadXMLFromFile(*self*, *FileName*)

Parameters

FileName (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromFileInject(*self*, *TargetFileName*, *InjectFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromString(*self*, *XMLData*)

Parameters

XMLData (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromStringInject(*self*, *TargetXMLData*, *InjectXMLData*)

Parameters

- **TargetXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromZIPData(*self*, *zipData*, *zipSize*)

Parameters

- **zipData** (*void const **)
- **zipSize** (*size_t*)

LoadXMLFromZIPFile(*self*, *ZipFileName*)

Parameters

ZipFileName (*Spinnaker::GenICam::gcstring const &*)

MergeXMLFiles(*self*, *TargetFileName*, *InjectedFileName*, *OutputFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectedFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

PreprocessXMLFromFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

PreprocessXMLFromZIPFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

property thisown

The membership flag

class PySpin.PySpin.CNodeMapPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)
- **Connect**(*self*
- **bool** (*pPort*) →)

- **pPort**

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring* const &)

GetNodes(*self*)

GetNumNodes(*self*) → *uint64_t*

InvalidateNodes(*self*)

IsValid(*self*) → *bool*

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

property thisown

The membership flag

class PySpin.PySpin.**CNodePtr**(*args)

Bases: *object*

Encapsulates a GenApi pointer dealing with the *dynamic_cast* automatically.

C++ includes: *Pointer.h*

DeregisterCallback(*self*, *hCallback*) → *bool*

Parameters

hCallback (*Spinnaker::GenApi::CallbackHandleType*)

GetAccessMode(*self*) → *Spinnaker::GenApi::EAccessMode*

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → *Spinnaker::GenApi::ECachingMode*

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=*ctReadingChildren*)

Parameters

LinkType (*enum Spinnaker::GenApi::ELinkType*)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)

- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

property thisown

The membership flag

class PySpin.PySpin.**CRegisterPtr**(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify*=True)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

Get(*self*, *pBuffer*, *Length*, *Verify*=False, *IgnoreCache*=False)

Parameters

- **pBuffer** (uint8_t *)
- **Length** (int64_t)
- **Verify** (bool)
- **IgnoreCache** (bool)
- **Get**(*self*
 - > (ignore_cache) -> std::vector< uint8_t)
- **size_read** (int64_t)
- **Get**(*self*

- `size_read`
- `>`
- `size_read`
- `verify_range (bool)`
- `Get(self`
- `size_read`
- `verify_range`
- `>`
- `size_read`
- `verify_range`
- `ignore_cache (bool)`

Gets a NumPy array representing the contents of the register, as 8-bit unsigned ints.

6.1 Parameters:

`pBuffer`: The number of bytes to retrieve

`Verify`: Enables Range verification (default = false). The `AccessMode` is always checked

`IgnoreCache`: If true the value is read ignoring any caches (default = false)

GetAccessMode(*self*) → `Spinnaker::GenApi::EAccessMode`

GetAddress(*self*) → `int64_t`

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → `Spinnaker::GenApi::ECachingMode`

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType*=`ctReadingChildren`)

Parameters

LinkType (*enum* `Spinnaker::GenApi::ELinkType`)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLength(*self*) → `int64_t`

GetLockNodes(*self*)

Parameters

LockNodes (`Spinnaker::GenApi::NodeList_t` &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (Spinnaker::GenICam::gcstring const &)
- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **AttributeStr** (Spinnaker::GenICam::gcstring &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (FeatureList_t &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (FeatureList_t &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → Spinnaker::GenApi::EVisibility

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (enum Spinnaker::GenApi::EAccessMode)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum Spinnaker::GenApi::EVisibility)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

Set(*self*, *pBuffer*, *Verify*=True)

Parameters

- **pBuffer** (uint8_t const *)

- **Verify** (bool)

Set the register's contents with the contents (as 8-bit unsigned ints) of the given array.

6.2 Parameters:

pBuffer: The NumPy array containing the data to set

Verify: Enables AccessMode and Range verification (default = true)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

ToString(*self*, *Verify*=False, *IgnoreCache*=False) → gcstring

Parameters

- **Verify** (bool)

- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.PySpin.CSelectorPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (FeatureList_t &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

IsSelector(*self*) → bool

IsValid(*self*) → bool

bool Spinnaker::GenApi::CPointer< T, B >::IsValid() const throw () true if the pointer is valid

property thisown

The membership flag

class PySpin.PySpin.CSelectorSet(*args, **kwargs)

Bases: [Node](#)

The set of selectors selecting a given node

C++ includes: SelectorSet.h

GetSelectorList(*self*, *Incremental=False*)

Parameters

- **Incremental** (*bool*)
- **void** (*virtual*)
- **Spinnaker::GenApi::CSelectorSet::GetSelectorList**(*FeatureList_t*
- **&SelectorList**
- **Incremental=false**) (*bool*)

IsEmpty(*self*) → bool

bool Spinnaker::GenApi::CSelectorSet::IsEmpty()

returns true if no selectors are present

Restore(*self*)

virtual void Spinnaker::GenApi::CSelectorSet::Restore()

SetFirst(*self*) → bool

virtual bool Spinnaker::GenApi::CSelectorSet::SetFirst()

SetNext(*self*, *Tick=True*) → bool

Parameters

- **Tick** (*bool*)
- **Tick=true**) (*virtual bool Spinnaker::GenApi::CSelectorSet::SetNext*(*bool*)

ToString(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::CSelectorSet::ToString()

property thisown

The membership flag

class PySpin.PySpin.CStringPtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → *INode*

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → *INode*

GetChildren(*self*, *LinkType=ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → *gcstring*

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (Spinnaker::GenApi::NodeList_t &)

GetMaxLength(*self*) → int64_t

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (bool)

GetNameSpace(*self*) → Spinnaker::GenApi::ENamespace

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (Spinnaker::GenApi::NodeList_t &)

GetPollingTime(*self*) → int64_t

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring* const &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetValue(*self*, *Verify*=False, *IgnoreCache*=False) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (Spinnaker::GenApi::CNodeCallback *)

SetReference(*self*, *pBase*)

Parameters

pBase (INode *)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → gcstring

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

property thisown

The membership flag

class PySpin.PySpin.CValuePtr(*args)

Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **Verify** (bool)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetAlias(*self*) → INode

GetCachingMode(*self*) → Spinnaker::GenApi::ECachingMode

GetCastAlias(*self*) → INode

GetChildren(*self*, *LinkType=ctReadingChildren*)

Parameters

LinkType (enum Spinnaker::GenApi::ELinkType)

GetDescription(*self*) → gcstring

GetDeviceName(*self*) → *gcstring*

GetDisplayName(*self*) → *gcstring*

GetDocuURL(*self*) → *gcstring*

GetEventID(*self*) → *gcstring*

GetLockNodes(*self*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNode(*self*) → *INode*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)
- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (*FeatureList_t* &)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (enum *Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → bool

IsDeprecated(*self*) → bool

IsFeature(*self*) → bool

IsSelector(*self*) → bool

IsStreamable(*self*) → bool

IsValid(*self*) → bool

bool *Spinnaker::GenApi::CPointer< T, B >::IsValid()* const throw () true if the pointer is valid

IsValueCacheValid(*self*) → bool

RegisterCallback(*self*, *pCallback*) → *Spinnaker::GenApi::CallbackHandleType*

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.Camera*(*args, **kwargs)

Bases: *CameraBase*

The camera object class.

C++ includes: *Camera.h*

property AasRoiEnable

property AasRoiHeight

property AasRoiOffsetX

property AasRoiOffsetY

property AasRoiWidth

property AcquisitionAbort
property AcquisitionArm
property AcquisitionBurstFrameCount
property AcquisitionFrameCount
property AcquisitionFrameRate
property AcquisitionFrameRateEnable
property AcquisitionFrameRatePersistence
property AcquisitionLineRate
property AcquisitionMode
property AcquisitionResultingFrameRate
property AcquisitionStart
property AcquisitionStatus
property AcquisitionStatusSelector
property AcquisitionStop
property AcquisitionTransferFrameRate
property ActionDeviceKey
property ActionGroupKey
property ActionGroupMask
property ActionQueueEmpty
property ActionQueueSize
property ActionSelector
property ActionSignalSize
property ActionUnconditionalMode
property AdaptiveCompressionEnable
property AdcBitDepth
property AutoAlgorithmSelector
property AutoExposureControlLoopDamping
property AutoExposureControlPriority
property AutoExposureEVCompensation
property AutoExposureExposureTimeLowerLimit
property AutoExposureExposureTimeUpperLimit

property AutoExposureGainLowerLimit
property AutoExposureGainUpperLimit
property AutoExposureGreyValueLowerLimit
property AutoExposureGreyValueUpperLimit
property AutoExposureLightingMode
property AutoExposureMeteringMode
property AutoExposureTargetGreyValue
property AutoExposureTargetGreyValueAuto
property BalanceRatio
property BalanceRatioSelector
property BalanceWhiteAuto
property BalanceWhiteAutoDamping
property BalanceWhiteAutoLowerLimit
property BalanceWhiteAutoProfile
property BalanceWhiteAutoUpperLimit
property BinningHorizontal
property BinningHorizontalMode
property BinningSelector
property BinningVertical
property BinningVerticalMode
property BlackLevel
property BlackLevelAuto
property BlackLevelAutoBalance
property BlackLevelClampingEnable
property BlackLevelRaw
property BlackLevelSelector
property BsiFlatFieldCorrectionAuto
property BsiFlatFieldCorrectionAutoDamping
property BsiFlatFieldCorrectionEnable
property BsiFlatFieldCorrectionGain
property BsiFlatFieldCorrectionGainSelector

property BufferedBurstFrameCountMax
property BufferedBurstMode
property ChunkBlackLevel
property ChunkBlackLevelSelector
property ChunkCRC
property ChunkCompressionMode
property ChunkCompressionRatio
property ChunkCounterSelector
property ChunkCounterValue
property ChunkCurrentDataRate
property ChunkEnable
property ChunkEncoderSelector
property ChunkEncoderStatus
property ChunkEncoderValue
property ChunkExposureEndLineStatusAll
property ChunkExposureTime
property ChunkExposureTimeSelector
property ChunkFrameID
property ChunkGain
property ChunkGainSelector
property ChunkHeight
property ChunkImage
property ChunkImageComponent
property ChunkInferenceBoundingBoxResult
property ChunkInferenceConfidence
property ChunkInferenceFrameId
property ChunkInferenceResult
property ChunkLinePitch
property ChunkLineStatusAll
property ChunkModeActive
property ChunkOffsetX

property ChunkOffsetY

property ChunkPartSelector

property ChunkPixelDynamicRangeMax

property ChunkPixelDynamicRangeMin

property ChunkPixelFormat

property ChunkRegionID

property ChunkScan3dAxisMax

property ChunkScan3dAxisMin

property ChunkScan3dCoordinateOffset

property ChunkScan3dCoordinateReferenceSelector

property ChunkScan3dCoordinateReferenceValue

property ChunkScan3dCoordinateScale

property ChunkScan3dCoordinateSelector

property ChunkScan3dCoordinateSystem

property ChunkScan3dCoordinateSystemReference

property ChunkScan3dCoordinateTransformSelector

property ChunkScan3dDistanceUnit

property ChunkScan3dInvalidDataFlag

property ChunkScan3dInvalidDataValue

property ChunkScan3dOutputMode

property ChunkScan3dTransformValue

property ChunkScanLineSelector

property ChunkSelector

property ChunkSequencerSetActive

property ChunkSerialData

property ChunkSerialDataLength

property ChunkSerialReceiveOverflow

property ChunkSourceID

property ChunkStreamChannelID

property ChunkTimerSelector

property ChunkTimerValue

property ChunkTimestamp

property ChunkTimestampLatchValue

property ChunkTransferBlockID

property ChunkTransferQueueCurrentBlockCount

property ChunkTransferStreamID

property ChunkWidth

property ClConfiguration

property ClTimeSlotsCount

property ColorTransformationEnable

property ColorTransformationSelector

property ColorTransformationValue

property ColorTransformationValueSelector

property ComponentActiveCount

property ComponentDestination

property ComponentEnable

property ComponentSelector

property CompressedFrameDropCount

property CompressionSaturationPriority

property ControlPacketsReservedBandwidth

property CounterDelay

property CounterDuration

property CounterEventActivation

property CounterEventSource

property CounterReset

property CounterResetActivation

property CounterResetSource

property CounterSelector

property CounterStatus

property CounterTriggerActivation

property CounterTriggerSource

property CounterValue

property CounterValueAtReset

property CxpConnectionSelector

property CxpConnectionTestErrorCount

property CxpConnectionTestMode

property CxpConnectionTestPacketCount

property CxpLinkConfiguration

property CxpLinkConfigurationPreferred

property CxpLinkConfigurationStatus

property CxpPoCxpAuto

property CxpPoCxpStatus

property CxpPoCxpTripReset

property CxpPoCxpTurnOff

property DecimationHorizontal

property DecimationHorizontalMode

property DecimationSelector

property DecimationVertical

property DecimationVerticalMode

property DefectCorrectStaticEnable

property DefectCorrectionMode

property DefectTableApply

property DefectTableCoordinateX

property DefectTableCoordinateY

property DefectTableFactoryRestore

property DefectTableIndex

property DefectTablePixelCount

property DefectTableSave

property DefectTableSensor

property Deinterlacing

property DeviceCharacterSet

property DeviceClockFrequency

property DeviceClockSelector

property DeviceConnectionSelector
property DeviceConnectionSpeed
property DeviceConnectionStatus
property DeviceEventChannelCount
property DeviceFamilyName
property DeviceFeaturePersistenceEnd
property DeviceFeaturePersistenceStart
property DeviceFirmwareVersion
property DeviceGenCPVersionMajor
property DeviceGenCPVersionMinor
property DeviceID
property DeviceIndicatorMode
property DeviceLinkBandwidthReserve
property DeviceLinkCommandTimeout
property DeviceLinkConnectionCount
property DeviceLinkCurrentThroughput
property DeviceLinkHeartbeatMode
property DeviceLinkHeartbeatTimeout
property DeviceLinkSelector
property DeviceLinkSpeed
property DeviceLinkThroughputLimit
property DeviceLinkThroughputLimitMode
property DeviceManifestEntrySelector
property DeviceManifestPrimaryURL
property DeviceManifestSchemaMajorVersion
property DeviceManifestSchemaMinorVersion
property DeviceManifestSecondaryURL
property DeviceManifestXMLMajorVersion
property DeviceManifestXMLMinorVersion
property DeviceManifestXMLSubMinorVersion
property DeviceManufacturerInfo

property DeviceMaxThroughput

property DeviceModelName

property DevicePowerSupplySelector

property DeviceRegistersCheck

property DeviceRegistersEndianness

property DeviceRegistersStreamingEnd

property DeviceRegistersStreamingStart

property DeviceRegistersValid

property DeviceReset

property DeviceSFNCVersionMajor

property DeviceSFNCVersionMinor

property DeviceSFNCVersionSubMinor

property DeviceScanType

property DeviceSensorChroma

property DeviceSerialNumber

property DeviceSerialPortBaudRate

property DeviceSerialPortSelector

property DeviceStreamChannelCount

property DeviceStreamChannelEndianness

property DeviceStreamChannelLink

property DeviceStreamChannelPacketSize

property DeviceStreamChannelSelector

property DeviceStreamChannelType

property DeviceTLType

property DeviceTLVersionMajor

property DeviceTLVersionMinor

property DeviceTLVersionSubMinor

property DeviceTapGeometry

property DeviceTemperature

property DeviceTemperatureSelector

property DeviceType

property DeviceUptime
property DeviceUserID
property DeviceVendorName
property DeviceVersion
property EncoderDivider
property EncoderMode
property EncoderOutputMode
property EncoderReset
property EncoderResetActivation
property EncoderResetSource
property EncoderSelector
property EncoderSourceA
property EncoderSourceB
property EncoderStatus
property EncoderTimeout
property EncoderValue
property EncoderValueAtReset
property EnumerationCount
property EventAcquisitionEnd
property EventAcquisitionEndFrameID
property EventAcquisitionEndTimestamp
property EventAcquisitionError
property EventAcquisitionErrorFrameID
property EventAcquisitionErrorTimestamp
property EventAcquisitionStart
property EventAcquisitionStartFrameID
property EventAcquisitionStartTimestamp
property EventAcquisitionTransferEnd
property EventAcquisitionTransferEndFrameID
property EventAcquisitionTransferEndTimestamp
property EventAcquisitionTransferStart

property EventAcquisitionTransferStartFrameID
property EventAcquisitionTransferStartTimestamp
property EventAcquisitionTrigger
property EventAcquisitionTriggerFrameID
property EventAcquisitionTriggerTimestamp
property EventActionLate
property EventActionLateFrameID
property EventActionLateTimestamp
property EventCounter0End
property EventCounter0EndFrameID
property EventCounter0EndTimestamp
property EventCounter0Start
property EventCounter0StartFrameID
property EventCounter0StartTimestamp
property EventCounter1End
property EventCounter1EndFrameID
property EventCounter1EndTimestamp
property EventCounter1Start
property EventCounter1StartFrameID
property EventCounter1StartTimestamp
property EventEncoder0Restarted
property EventEncoder0RestartedFrameID
property EventEncoder0RestartedTimestamp
property EventEncoder0Stopped
property EventEncoder0StoppedFrameID
property EventEncoder0StoppedTimestamp
property EventEncoder1Restarted
property EventEncoder1RestartedFrameID
property EventEncoder1RestartedTimestamp
property EventEncoder1Stopped
property EventEncoder1StoppedFrameID

property EventEncoder1StoppedTimestamp
property EventError
property EventErrorCode
property EventErrorFrameID
property EventErrorTimestamp
property EventExposureEnd
property EventExposureEndFrameID
property EventExposureEndTimestamp
property EventExposureStart
property EventExposureStartFrameID
property EventExposureStartTimestamp
property EventFrameBurstEnd
property EventFrameBurstEndFrameID
property EventFrameBurstEndTimestamp
property EventFrameBurstStart
property EventFrameBurstStartFrameID
property EventFrameBurstStartTimestamp
property EventFrameEnd
property EventFrameEndFrameID
property EventFrameEndTimestamp
property EventFrameStart
property EventFrameStartFrameID
property EventFrameStartTimestamp
property EventFrameTransferEnd
property EventFrameTransferEndFrameID
property EventFrameTransferEndTimestamp
property EventFrameTransferStart
property EventFrameTransferStartFrameID
property EventFrameTransferStartTimestamp
property EventFrameTrigger
property EventFrameTriggerFrameID

property EventFrameTriggerTimestamp
property EventLine0AnyEdge
property EventLine0AnyEdgeFrameID
property EventLine0AnyEdgeTimestamp
property EventLine0FallingEdge
property EventLine0FallingEdgeFrameID
property EventLine0FallingEdgeTimestamp
property EventLine0RisingEdge
property EventLine0RisingEdgeFrameID
property EventLine0RisingEdgeTimestamp
property EventLine1AnyEdge
property EventLine1AnyEdgeFrameID
property EventLine1AnyEdgeTimestamp
property EventLine1FallingEdge
property EventLine1FallingEdgeFrameID
property EventLine1FallingEdgeTimestamp
property EventLine1RisingEdge
property EventLine1RisingEdgeFrameID
property EventLine1RisingEdgeTimestamp
property EventLinkSpeedChange
property EventLinkSpeedChangeFrameID
property EventLinkSpeedChangeTimestamp
property EventLinkTrigger0
property EventLinkTrigger0FrameID
property EventLinkTrigger0Timestamp
property EventLinkTrigger1
property EventLinkTrigger1FrameID
property EventLinkTrigger1Timestamp
property EventNotification
property EventSelector
property EventSequencerSetChange

property EventSequencerSetChangeFrameID
property EventSequencerSetChangeTimestamp
property EventSerialData
property EventSerialDataLength
property EventSerialPortReceive
property EventSerialPortReceiveTimestamp
property EventSerialReceiveOverflow
property EventStream@TransferBlockEnd
property EventStream@TransferBlockEndFrameID
property EventStream@TransferBlockEndTimestamp
property EventStream@TransferBlockStart
property EventStream@TransferBlockStartFrameID
property EventStream@TransferBlockStartTimestamp
property EventStream@TransferBlockTrigger
property EventStream@TransferBlockTriggerFrameID
property EventStream@TransferBlockTriggerTimestamp
property EventStream@TransferBurstEnd
property EventStream@TransferBurstEndFrameID
property EventStream@TransferBurstEndTimestamp
property EventStream@TransferBurstStart
property EventStream@TransferBurstStartFrameID
property EventStream@TransferBurstStartTimestamp
property EventStream@TransferEnd
property EventStream@TransferEndFrameID
property EventStream@TransferEndTimestamp
property EventStream@TransferOverflow
property EventStream@TransferOverflowFrameID
property EventStream@TransferOverflowTimestamp
property EventStream@TransferPause
property EventStream@TransferPauseFrameID
property EventStream@TransferPauseTimestamp

property EventStream0TransferResume
property EventStream0TransferResumeFrameID
property EventStream0TransferResumeTimestamp
property EventStream0TransferStart
property EventStream0TransferStartFrameID
property EventStream0TransferStartTimestamp
property EventTest
property EventTestTimestamp
property EventTimer0End
property EventTimer0EndFrameID
property EventTimer0EndTimestamp
property EventTimer0Start
property EventTimer0StartFrameID
property EventTimer0StartTimestamp
property EventTimer1End
property EventTimer1EndFrameID
property EventTimer1EndTimestamp
property EventTimer1Start
property EventTimer1StartFrameID
property EventTimer1StartTimestamp
property ExposureActiveMode
property ExposureAuto
property ExposureMode
property ExposureTime
property ExposureTimeMode
property ExposureTimeSelector
property ExternalVoltageEnable
property ExternalVoltageSelector
property ExternalVoltageValue
property FactoryReset
property FfcEnable

property FfcMode

property FfcUserGain

property FfcUserOffset

property FfcUserTableReset

property FfcUserTableSave

property FfcUserTableXCoordinate

property FileAccessBuffer

property FileAccessLength

property FileAccessOffset

property FileOpenMode

property FileOperationExecute

property FileOperationResult

property FileOperationSelector

property FileOperationStatus

property FileSelector

property FileSize

property Gain

property GainAuto

property GainAutoBalance

property GainConversion

property GainSelector

property Gamma

property GammaEnable

property GevActiveLinkCount

property GevCCP

property GevCurrentDefaultGateway

property GevCurrentIPAddress

property GevCurrentIPConfigurationDHCP

property GevCurrentIPConfigurationLLA

property GevCurrentIPConfigurationPersistentIP

property GevCurrentPhysicalLinkConfiguration

property `GevCurrentSubnetMask`
property `GevDiscoveryAckDelay`
property `GevFirstURL`
property `GevGVCPExtendedStatusCodes`
property `GevGVCPExtendedStatusCodesSelector`
property `GevGVCPHeartbeatDisable`
property `GevGVCPPendingAck`
property `GevGVCPPendingTimeout`
property `GevGVSPExtendedIDMode`
property `GevHeartbeatTimeout`
property `GevIEEE1588`
property `GevIEEE1588ClockAccuracy`
property `GevIEEE1588ClockId`
property `GevIEEE1588DataSetLatch`
property `GevIEEE1588Mode`
property `GevIEEE1588OffsetFromMasterLatched`
property `GevIEEE1588ParentClockIdLatched`
property `GevIEEE1588Status`
property `GevIEEE1588StatusLatched`
property `GevIPConfigurationStatus`
property `GevInterfaceSelector`
property `GevMACAddress`
property `GevMCDA`
property `GevMCPHostPort`
property `GevMCRC`
property `GevMCSP`
property `GevMCTT`
property `GevNumberOfActiveLinks`
property `GevNumberOfInterfaces`
property `GevPAUSEFrameReception`
property `GevPAUSEFrameTransmission`

property `GevPersistentDefaultGateway`
property `GevPersistentIPAddress`
property `GevPersistentSubnetMask`
property `GevPhysicalLinkConfiguration`
property `GevPhysicalLinkConfigurationCapability`
property `GevPrimaryApplicationIPAddress`
property `GevPrimaryApplicationSocket`
property `GevPrimaryApplicationSwitchoverKey`
property `GevSCCFGAllInTransmission`
property `GevSCCFGExtendedChunkData`
property `GevSCCFGPacketResendDestination`
property `GevSCCFGUnconditionalStreaming`
property `GevSCDA`
property `GevSCPD`
property `GevSCPDirection`
property `GevSCPHostPort`
property `GevSCPInterfaceIndex`
property `GevSCPSBigEndian`
property `GevSCPSToNotFragment`
property `GevSCPSFireTestPacket`
property `GevSCPSPacketSize`
property `GevSCSP`
property `GevSCZoneConfigurationLock`
property `GevSCZoneCount`
property `GevSCZoneDirectionAll`
property `GevSecondURL`
property `GevStreamChannelSelector`
property `GevSupportedOption`
property `GevSupportedOptionSelector`
property `GevTimestampTickFrequency`
property `GuiXmlManifestAddress`

property Height
property HeightMax
property ImageComponentEnable
property ImageComponentSelector
property ImageCompressionBitrate
property ImageCompressionJPEGFormatOption
property ImageCompressionMode
property ImageCompressionQuality
property ImageCompressionRateOption
Init(*self*)
 void Spinnaker::Camera::Init()
property IspEnable
property LUTEnable
property LUTIndex
property LUTSelector
property LUTValue
property LUTValueAll
property LargePenalty
property LensShadingCoefficientActiveSet
property LensShadingCorrectionCalibration
property LensShadingCorrectionCalibrationGainLimit
property LensShadingCorrectionCalibrationSetup
property LensShadingCorrectionCalibrationStatus
property LensShadingCorrectionMode
property LensShadingCorrectionStepSize
property LensShadingCorrectionVersion
property LineFilterWidth
property LineFormat
property LineInputFilterSelector
property LineInverter
property LineMode

property LinePitch
property LineSelector
property LineSource
property LineStatus
property LineStatusAll
property LinkErrorCount
property LinkRecoveryCount
property LinkUptime
property LogicBlockLUTInputActivation
property LogicBlockLUTInputSelector
property LogicBlockLUTInputSource
property LogicBlockLUTOutputValue
property LogicBlockLUTOutputValueAll
property LogicBlockLUTRowIndex
property LogicBlockLUTSelector
property LogicBlockSelector
property MaxDataRateThreshold
property MaxDeviceResetTime
property MultiRoiConfigurationInvalidReason
property MultiRoiConfigurationInvalidReasonAll
property MultiRoiEnable
property MultiRoiFeatureEnable
property MultiRoiHeight
property MultiRoiOffsetX
property MultiRoiOffsetY
property MultiRoiSelector
property MultiRoiWidth
property MultiRoiWindows
property NumDirections
property OffsetX
property OffsetY

property PacketResendRequestCount

property PacketResendRequestsDroppedCount

property PauseFrameCount

property PayloadSize

property PixelColorFilter

property PixelDynamicRangeMax

property PixelDynamicRangeMin

property PixelFormat

property PixelFormatInfoID

property PixelFormatInfoSelector

property PixelSize

property PowerSupplyCurrent

property PowerSupplyVoltage

property RegionDestination

property RegionMode

property RegionSelector

property ReverseX

property ReverseY

property RgbTransformLightSource

property Saturation

property SaturationEnable

property Scan3dAxisMax

property Scan3dAxisMin

property Scan3dBaseline

property Scan3dCoordinateOffset

property Scan3dCoordinateReferenceSelector

property Scan3dCoordinateReferenceValue

property Scan3dCoordinateScale

property Scan3dCoordinateSelector

property Scan3dCoordinateSystem

property Scan3dCoordinateSystemReference

property Scan3dCoordinateTransformSelector
property Scan3dDistanceUnit
property Scan3dFocalLength
property Scan3dInvalidDataFlag
property Scan3dInvalidDataValue
property Scan3dOutputMode
property Scan3dPrincipalPointU
property Scan3dPrincipalPointV
property Scan3dTransformValue
property SensorDescription
property SensorDigitizationTaps
property SensorHeight
property SensorShutterMode
property SensorTaps
property SensorWidth
property SequencerConfigurationMode
property SequencerConfigurationReset
property SequencerConfigurationValid
property SequencerFeatureEnable
property SequencerMode
property SequencerPathSelector
property SequencerSetActive
property SequencerSetLoad
property SequencerSetNext
property SequencerSetSave
property SequencerSetSelector
property SequencerSetStart
property SequencerSetValid
property SequencerTriggerActivation
property SequencerTriggerSource
property SerialPortBaudRate

property SerialPortDataBits
property SerialPortParity
property SerialPortSelector
property SerialPortSource
property SerialPortStopBits
property SerialReceiveFramingErrorCount
property SerialReceiveParityErrorCount
property SerialReceiveQueueClear
property SerialReceiveQueueCurrentCharacterCount
property SerialReceiveQueueMaxCharacterCount
property SerialTransmitQueueCurrentCharacterCount
property SerialTransmitQueueMaxCharacterCount
property Sharpening
property SharpeningAuto
property SharpeningEnable
property SharpeningThreshold
property SmallPenalty
property SoftwareSignalPulse
property SoftwareSignalSelector
property SourceCount
property SourceSelector
property StereoHeight
property StereoResolution
property StereoWidth
property TLParamsLocked
property Test0001
property TestEventGenerate
property TestPattern
property TestPatternGeneratorSelector
property TestPendingAck
property TimerDelay

property TimerDuration
property TimerReset
property TimerSelector
property TimerStatus
property TimerTriggerActivation
property TimerTriggerSource
property TimerValue
property Timestamp
property TimestampIncrement
property TimestampLatch
property TimestampLatchValue
property TimestampReset
property TotalDisparity
property TransferAbort
property TransferBlockCount
property TransferBurstCount
property TransferComponentSelector
property TransferControlMode
property TransferOperationMode
property TransferPause
property TransferQueueCurrentBlockCount
property TransferQueueMaxBlockCount
property TransferQueueMode
property TransferQueueOverflowCount
property TransferResume
property TransferSelector
property TransferStart
property TransferStatus
property TransferStatusSelector
property TransferStop
property TransferStreamChannel

property TransferTriggerActivation
property TransferTriggerMode
property TransferTriggerSelector
property TransferTriggerSource
property TransmissionDelay
property TransmissionDelayAverage
property TransmissionDelayMax
property TriggerActivation
property TriggerDelay
property TriggerDivider
property TriggerEventTest
property TriggerMode
property TriggerMultiplier
property TriggerOverlap
property TriggerSelector
property TriggerSoftware
property TriggerSource
property U3VAccessPrivilege
property U3VCPCapability
property U3VCPEIRMAvailable
property U3VCPIIDC2Available
property U3VCPSIRMAvailable
property U3VCurrentSpeed
property U3VMaxAcknowledgeTransferLength
property U3VMaxCommandTransferLength
property U3VMaxDeviceResponseTime
property U3VMessageChannelID
property U3VNumberOfStreamChannels
property U3VVersionMajor
property U3VVersionMinor
property UniquenessRatio

`property UserOutputSelector`
`property UserOutputValue`
`property UserOutputValueAll`
`property UserOutputValueAllMask`
`property UserSetDefault`
`property UserSetFeatureEnable`
`property UserSetLoad`
`property UserSetSave`
`property UserSetSelector`
`property V3_3Enable`
`property WhiteClip`
`property WhiteClipSelector`
`property Width`
`property WidthMax`
`property WindowSizeH`
`property WindowSizeW`
`property aPAUSEMACtrlFramesReceived`
`property aPAUSEMACtrlFramesTransmitted`
`property thisown`

The membership flag

class `PySpin.PySpin.CameraBase(*args, **kwargs)`

Bases: [*ICameraBase*](#)

The base class for the camera object.

C++ includes: CameraBase.h

BeginAcquisition(*self*)

void Spinnaker::CameraBase::BeginAcquisition()

Starts the image acquisition engine. The camera must be initialized via a call to `Init()` before starting an acquisition.

See: `Init()`

DeInit(*self*)

void Spinnaker::CameraBase::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: Init()

See: UnregisterEvent(Event & evtToUnregister)

DiscoverMaxPacketSize(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::DiscoverMaxPacketSize()

Returns the largest packet size that can be safely used on the interface that device is connected to

The maximum packet size returned.

EndAcquisition(*self*)

void Spinnaker::CameraBase::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: Init()

See: BeginAcquisition()

See: GetNextImage(grabTimeout)

See: Image::Release()

ForceIP(*self*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GenApi::EAccessMode Spinnaker::CameraBase::GetAccessMode() const

Returns the access mode that the software has on the Camera. The camera does not need to be initialized before calling this function.

See: Init()

An enumeration value indicating the access mode

GetActiveNumDataStreams(*self*) → unsigned int

GetBufferOwnership(*self*) → Spinnaker::BufferOwnership

GetDeviceID(*self*) → *gcstring*

GetGuiXml(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetGuiXml() const

Returns the GUI XML that can be passed into the Spinnaker GUI framework

GenICam::gcstring that represents the uncompressed GUI XML file

GetNextImage(*self*, grabTimeout=EVENT_TIMEOUT_INFINITE, streamIndex=0) → *ImagePtr*

Parameters

- **grabTimeout** (a 64bit value that represents a timeout in milliseconds)
- **streamIndex** (uint64_t)
- **ImagePtr**
- **Spinnaker::CameraBase::GetNextImage(uint64_t**
- **grabTimeout=EVENT_TIMEOUT_INFINITE**

- **streamID=0** (*uint64_t*)
- **This** (*Gets the next image that was received by the transport layer.*)
- **cameras** (*function will block indefinitely until an image arrives. Most*)
- **camera** (*support one stream so the default streamID is 0 but if a*)
- **select** (*supports multiple streams the user can input the streamID to*)
- **images** (*from which stream to grab*)
- **See** (*EndAcquisition()*)
- **See**
- **See**
- **Parameters**
- -----
- **grabTimeout**
- **streamID** (*The stream to grab the image.*)
- **object** (*pointer to an Image*)

GetNextImageSync(*self*, *grabTimeout=EVENT_TIMEOUT_INFINITE*) → *ImageList*

Parameters

grabTimeout (*uint64_t*)

GetNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetNodeMap() const

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: Init()

A reference to the INodeMap.

GetNumDataStreams(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::GetNumDataStreams()

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse(*self*) → unsigned int

unsigned int Spinnaker::CameraBase::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().

The number of images that needs to be cleaned up.

GetTLDeviceNodeMap(*self*) → *INodeMap*

GenApi::INodeMap& Spinnaker::CameraBase::GetTLDeviceNodeMap() const

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.

A reference to the INodeMap.

GetTLStreamNodeMap(*self*, *streamIndex*=0) → *INodeMap*

Parameters

- **streamIndex** (*uint64_t*)
- **const** (*GenApi::INodeMap& Spinnaker::CameraBase::GetTLStreamNodeMap()*)
- **XML** (Gets a reference to the node map that is generated from a *GenICam*)
- **be** (file for the *GenTL Stream* module. The camera does not need to)
- **map.** (initialized before acquiring this node)
- **INodeMap.** (A reference to the)

GetUniqueID(*self*) → *gcstring*

GenICam::gcstring Spinnaker::CameraBase::GetUniqueID()

This returns a unique id string that identifies the camera. This is the camera serial number.
string that uniquely identifies the camera (serial number)

GetUserBufferCount(*self*) → *uint64_t*

GetUserBufferSize(*self*) → *uint64_t*

GetUserBufferTotalSize(*self*) → *uint64_t*

Init(*self*)

void Spinnaker::CameraBase::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: *BeginAcquisition()*

See: *EndAcquisition()*

See: *GetNodeMap()*

See: *GetNextImage()*

IsInitialized(*self*) → *bool*

bool Spinnaker::CameraBase::IsInitialized()

Checks if camera is initialized. This function needs to return true in order to retrieve a valid *NodeMap* from the *GetNodeMap()* call.

See: *GetNodeMap()*

If camera is initialized or not

IsStreaming(*self*) → *bool*

bool Spinnaker::CameraBase::IsStreaming() const

Returns true if the camera is currently streaming or false if it is not.

See: *Init()*

returns true if camera is streaming and false otherwise.

IsValid(*self*) → bool

bool Spinnaker::CameraBase::IsValid()

Checks a flag to determine if camera is still valid for use.

If camera is valid or not

Note that CameraPtr and CameraBase both define an IsValid() function. In order to determine the validity of the camera using a CameraPtr, user must first call get() to retrieve the CameraBase object.

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::ImageEventHandler &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (*Spinnaker::GenICam::gcstring const &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**
- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self*, *mode*)

Parameters

mode (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self*, *pMemBuffers*, *totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

property thisown

The membership flag

class PySpin.PySpin.CameraList(*args)

Bases: *ICameraList*

Used to hold a list of camera objects.

C++ includes: CameraList.h

Add(self, camera)

Parameters

camera (Spinnaker::CameraPtr)

Append(self, list)

Parameters

- **list** (Spinnaker::CameraList const &)
- **void**
- **&otherList** (Spinnaker::CameraList::Append(*CameraList*)
- **list.** (*Appends a camera list to the current*)
- **Parameters**
- -----
- **otherList** (*The other list to append to this list*)

Clear(self)

void Spinnaker::CameraList::Clear()

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: System:ReleaseInstance()

GetByDeviceID(self, deviceID) → *CameraPtr*

Parameters

deviceID (std::string)

GetByIndex(self, index) → *CameraPtr*

Parameters

- **index** (*The index at which to retrieve the camera object*)
- **CameraPtr**
- **const** (Spinnaker::CameraList::GetByIndex(int index))
- **"index".** (*Returns a pointer to a camera object at the*)
- **Parameters**
- -----
- **index**
- **object.** (*A pointer to an camera*)

GetBySerial(*self*, *serialNumber*) → *CameraPtr*

Parameters

- **serialNumber** (*The serial number of the camera object to retrieve*)
- **CameraPtr**
- **const** (*Spinnaker::CameraList::GetBySerial(std::string serialNumber)*)
- **number.** (*Returns a pointer to a camera object with the specified serial*)
- **Parameters**
- -----
- **serialNumber**
- **object.** (*A pointer to an camera*)

GetSize(*self*) → unsigned int

int Spinnaker::CameraList::GetSize() const

Returns the size of the camera list. The size is the number of Camera objects stored in the list.

An integer that represents the list size.

Remove(*self*, *camera*)

Parameters

camera (*Spinnaker::CameraPtr*)

RemoveByDeviceID(*self*, *deviceID*)

Parameters

deviceID (*std::string*)

RemoveByIndex(*self*, *index*)

Parameters

- **index** (*The index at which to remove the Camera object*)
- **void**
- **index** (*Spinnaker::CameraList::RemoveByIndex(int)*)
- **reference** (*Removes a camera at "index" and destroys its corresponding*)
- **object.** (*counted*)
- **Parameters**
- -----
- **index**

RemoveBySerial(*self*, *serialNumber*)

Parameters

- **serialNumber** (*The serial number of the Camera object to remove*)
- **void**
- **serialNumber** (*Spinnaker::CameraList::RemoveBySerial(std::string)*)

- **its** (*Removes a camera using its serial number and destroys*)
- **object.** (*corresponding reference counted*)
- **Parameters**
- -----
- **serialNumber**

property thisown

The membership flag

class PySpin.PySpin.CameraPtr(*args)

Bases: _SWIG_CamPtr

A reference tracked pointer to a camera object.

C++ includes: CameraPtr.h

property thisown

The membership flag

class PySpin.PySpin.CategoryNode(*args, **kwargs)

Bases: [ICategory](#), [ValueNode](#)

Interface for string properties.

C++ includes: CategoryNode.h

GetFeatures(self)

virtual void Spinnaker::GenApi::CategoryNode::GetFeatures(FeatureList_t &Features) const

Get all features of the category (including sub-categories)

SetReference(self, pBase)**Parameters**

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::CategoryNode::SetReference(INode** (*virtual void*)
- ***pBase)**
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.PySpin.ChannelStatistics(image, channel)

Bases: object

Class used to store statistics (as properties) for one channel of an image. Properties:

- **channel**: The image channel that the statistics are based on (as an int).
- **range_min**: The smallest possible pixel value.
- **range_max**: The largest possible pixel value.
- **pixel_value_min**: The smallest pixel value in the current channel.
- **pixel_value_max**: The largest pixel value in the current channel.
- **num_pixel_values**: The total number of pixel values in the current channel.

- `pixel_value_mean`: The average pixel value in the current channel.
- `histogram`: NumPy array representing the histogram of the current channel.

property `channel`

property `histogram`

property `num_pixel_values`

property `pixel_value_max`

property `pixel_value_mean`

property `pixel_value_min`

property `range_max`

property `range_min`

property `thisown`

The membership flag

class `PySpin.PySpin.ChunkData(*args)`

Bases: [*IChunkData*](#)

The chunk data which contains additional information about an image.

C++ includes: `ChunkData.h`

GetBlackLevel(*self*) → `float64_t`

`float64_t Spinnaker::ChunkData::GetBlackLevel() const`

Description: Returns the black level used to capture the image included in the payload. Visibility: Expert

GetCRC(*self*) → `int64_t`

GetCompressionMode(*self*) → `int64_t`

GetCompressionRatio(*self*) → `float64_t`

GetCounterValue(*self*) → `int64_t`

`int64_t Spinnaker::ChunkData::GetCounterValue() const`

Description: Returns the value of the selected Chunk counter at the time of the FrameStart event. Visibility: Expert

GetCurrentDataRate(*self*) → `int64_t`

GetEnable(*self*) → `bool`

GetEncoderValue(*self*) → `int64_t`

`int64_t Spinnaker::ChunkData::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. Visibility: Expert

GetExposureEndLineStatusAll(*self*) → `int64_t`

GetExposureTime(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetExposureTime() const

Description: Returns the exposure time used to capture the image. Visibility: Expert

GetFrameID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetFrameID() const

Description: Returns the unique Identifier of the frame (or image) included in the payload. Visibility: Expert

GetGain(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetGain() const

Description: Returns the gain used to capture the image. Visibility: Expert

GetHeight(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetHeight() const

Description: Returns the Height of the image included in the payload. Visibility: Expert

GetImage(*self*) → int64_t

GetInferenceBoundingBoxResult(*self*) → *InferenceBoundingBoxResult*

GetInferenceConfidence(*self*) → float64_t

GetInferenceFrameId(*self*) → int64_t

GetInferenceResult(*self*) → int64_t

GetLinePitch(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLinePitch() const

Description: Returns the LinePitch of the image included in the payload. Visibility: Expert

GetLineStatusAll(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetLineStatusAll() const

Description: Returns the status of all the I/O lines at the time of the FrameStart internal event. Visibility: Expert

GetModeActive(*self*) → bool

GetOffsetX(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetX() const

Description: Returns the OffsetX of the image included in the payload. Visibility: Expert

GetOffsetY(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetOffsetY() const

Description: Returns the OffsetY of the image included in the payload. Visibility: Expert

GetPartSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPartSelector() const

Description: Selects the part to access in chunk data in a multipart transmission. Visibility: Expert

GetPixelDynamicRangeMax(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMax() const

Description: Returns the maximum value of dynamic range of the image included in the payload. Visibility: Expert

GetPixelDynamicRangeMin(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetPixelDynamicRangeMin() const

Description: Returns the minimum value of dynamic range of the image included in the payload. Visibility: Expert

GetScan3dAxisMax(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMax() const

Description: Returns the Maximum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dAxisMin(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dAxisMin() const

Description: Returns the Minimum Axis value for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateOffset(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateOffset() const

Description: Returns the Offset for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dCoordinateReferenceValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateReferenceValue() const

Description: Reads the value of a position or pose coordinate for the anchor or transformed coordinate systems relative to the reference point. Visibility: Expert

GetScan3dCoordinateScale(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

Description: Returns the Scale for the selected coordinate axis of the image included in the payload. Visibility: Expert

GetScan3dInvalidDataFlag(*self*) → bool

GetScan3dInvalidDataValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dInvalidDataValue() const

Description: Returns the Invalid Data Value used for the image included in the payload. Visibility: Expert

GetScan3dTransformValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const

Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetScanLineSelector() const

Description: Index for vector representation of one chunk value per line in an image. Visibility: Expert

GetSequencerSetActive(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetSequencerSetActive() const

Description: Return the index of the active set of the running sequencer included in the payload. Visibility: Expert

GetSerialData(*self*) → uint8_t *

GetSerialDataLength(*self*) → int64_t

GetSerialReceiveOverflow(*self*) → bool

GetStreamChannelID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetStreamChannelID() const

Description: Returns identifier of the stream channel used to carry the block. Visibility: Expert

GetTimerValue(*self*) → float64_t

float64_t Spinnaker::ChunkData::GetTimerValue() const

Description: Returns the value of the selected Timer at the time of the FrameStart internal event. Visibility: Expert

GetTimestamp(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTimestamp() const

Description: Returns the Timestamp of the image included in the payload at the time of the FrameStart internal event. Visibility: Expert

GetTimestampLatchValue(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const

Description: Returns the last Timestamp latched with the TimestampLatch command. Visibility: Expert

GetTransferBlockID(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTransferBlockID() const

Description: Returns the unique identifier of the transfer block used to transport the payload. Visibility: Expert

GetTransferQueueCurrentBlockCount(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const

Description: Returns the current number of blocks in the transfer queue. Visibility: Expert

GetWidth(*self*) → int64_t

int64_t Spinnaker::ChunkData::GetWidth() const

Description: Returns the Width of the image included in the payload. Visibility: Expert

SetChunks(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap &*)
- **void**
- **&pNodeMap** (*Spinnaker::ChunkData::SetChunks(GenApi::INodeMap)*)

property thisown

The membership flag

PySpin.PySpin.**Combine**(*Peter, Paul*) → Spinnaker::GenApi::EAccessMode

Parameters

- **Peter** (*Spinnaker::GenApi::Combine(ECachingMode)*)
- **Paul** (*enum Spinnaker::GenApi::ECachingMode*)
- **Combine**(**Peter**
- **Spinnaker::GenApi::EVisibility** (*Paul*) ->)
- **Peter**
- **Paul**
- **Combine**(**Peter**
- **Spinnaker::GenApi::ECachingMode** (*Paul*) ->)
- **Peter**
- **Paul**
- **ECachingMode**
- **Peter**
- **Paul**) (*ECachingMode*)
- **combination** (*Computes which CachingMode results from a*)

class PySpin.PySpin.**CommandNode**(*args, **kwargs)

Bases: *ICommand, ValueNode*

Interface for string properties.

C++ includes: CommandNode.h

Execute(*self, Verify=True*)

Parameters

- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Verify=true** (*virtual void Spinnaker::GenApi::CommandNode::Execute(bool)*)
- **command** (*Execute the*)
- **Parameters**
- -----
- **Verify**

IsDone(*self, Verify=True*) → bool

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **Verify=true** (*virtual bool Spinnaker::GenApi::CommandNode::IsDone(bool)*)
- **executed** (*Query whether the command is*)
- **Parameters**
- -----
- **Verify**

- **checked** (*is always*)
- **otherwise** (*True if the Execute command has finished; false*)

SetReference(*self, pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::CommandNode::SetReference**(*INode (virtual void)*)
- ***pBase**
- **Value** (*overload SetReference for*)

property thisown

The membership flag

PySpin.PySpin.DeregisterNodeCallback(*f*)

Parameters

f (*NodeCallback &*)

class PySpin.PySpin.DeviceArrivalEventHandler

Bases: *IDeviceArrivalEventHandler*

Proxy of C++ Spinnaker::DeviceArrivalEventHandler class.

OnDeviceArrival(*self, pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.PySpin.DeviceEventExposureEndData

Bases: *object*

Proxy of C++ Spinnaker::DeviceEventExposureEndData class.

property frameID

property thisown

The membership flag

class PySpin.PySpin.DeviceEventHandler

Bases: *IDeviceEventHandler*

Proxy of C++ Spinnaker::DeviceEventHandler class.

GetDeviceEventId(*self*) → *uint64_t*

GetDeviceEventName(*self*) → *gcstring*

OnDeviceEvent(*self, eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

class PySpin.PySpin.DeviceEventInferenceData

Bases: object

Proxy of C++ Spinnaker::DeviceEventInferenceData class.

property confidence

property frameID

property result

property thisown

The membership flag

class PySpin.PySpin.DeviceRemovalEventHandler

Bases: *IDeviceRemovalEventHandler*

Proxy of C++ Spinnaker::DeviceRemovalEventHandler class.

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

PySpin.PySpin.DoesEnvironmentVariableExist(*VariableName*) → bool

Parameters

- **VariableName** (*Spinnaker::GenICam::gcstring const &*)
- **bool** (*SPINNAKER_API*)
- **Spinnaker::GenICam::DoesEnvironmentVariableExist(const**
- **&VariableName)** (*Spinnaker::GenICam::gcstring*)
- **exists** (*Returns true if an environment variable*)

class PySpin.PySpin.EAccessModeClass

Bases: object

Holds conversion methods for the access mode enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EAccessMode **)
- **gcstring** (*ToString(Value) ->*)

- **Value** (*enum Spinnaker::GenApi::EAccessMode*)

property thisown

The membership flag

class PySpin.PySpin.ECachingModeClass

Bases: object

Holds conversion methods for the caching mode enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::ECachingMode **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ECachingMode*)

property thisown

The membership flag

class PySpin.PySpin.EDisplayNotationClass

Bases: object

Holds conversion methods for the notation type of floats.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EDisplayNotation **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EDisplayNotation*)

property thisown

The membership flag

class PySpin.PySpin.EEndianessClass

Bases: object

Holds conversion methods for the endianess enumeration.

C++ includes: EnumClasses.h

static **FromString**(ValueStr, pValue) → bool

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::EEndianess *)

static **ToString**(ValueStr, pValue)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::EEndianess *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::EEndianess)

property thisown

The membership flag

class PySpin.PySpin.ESchemaVersionClass

Bases: object

helper class converting ESchemaVersion from and to string

C++ includes: EnumClasses.h

static **FromString**(ValueStr, pValue) → bool

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::ESchemaVersion *)

static **ToString**(ValueStr, pValue)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::ESchemaVersion *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::ESchemaVersion)

property thisown

The membership flag

class PySpin.PySpin.EInputDirectionClass

Bases: object

Holds conversion methods for the notation type of floats.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::EInputDirection *)

static ToString(ValueStr, pValue)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::EInputDirection *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::EInputDirection)

property thisown

The membership flag

class PySpin.PySpin.ENamespaceClass

Bases: object

Holds conversion methods for the namespace enumeration.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::ENamespace *)

static ToString(ValueStr, pValue)

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring &)
- **pValue** (Spinnaker::GenApi::ENamespace *)
- **gcstring** (ToString(Value) ->)
- **Value** (enum Spinnaker::GenApi::ENamespace)

property thisown

The membership flag

class PySpin.PySpin.ERepresentationClass

Bases: object

Holds conversion methods for the representation enumeration.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- **ValueStr** (Spinnaker::GenICam::gcstring const &)
- **pValue** (Spinnaker::GenApi::ERepresentation *)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::ERepresentation* *)
- **gcstring** (*ToString(Value)* ->)
- **Value** (*enum Spinnaker::GenApi::ERepresentation*)

property thisown

The membership flag

class PySpin.PySpin.**ESignClass**

Bases: object

Holds conversion methods for the sign enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* const &)
- **pValue** (*Spinnaker::GenApi::ESign* *)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::ESign* *)
- **gcstring** (*ToString(Value)* ->)
- **Value** (*enum Spinnaker::GenApi::ESign*)

property thisown

The membership flag

class PySpin.PySpin.**ESlopeClass**

Bases: object

Holds conversion methods for the converter formulas.

C++ includes: EnumClasses.h

static FromString(*ValueStr*, *pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* const &)
- **pValue** (*Spinnaker::GenApi::ESlope* *)

static ToString(*ValueStr*, *pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)
- **pValue** (*Spinnaker::GenApi::ESlope* *)

- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::ESlope*)

property thisown

The membership flag

class PySpin.PySpin.EStandardNameSpaceClass

Bases: object

Holds conversion methods for the standard namespace enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EStandardNameSpace **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EStandardNameSpace **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EStandardNameSpace*)

property thisown

The membership flag

class PySpin.PySpin.EVisibilityClass

Bases: object

Holds conversion methods for the visibility enumeration.

C++ includes: EnumClasses.h

static FromString(*ValueStr, pValue*) → bool

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **pValue** (*Spinnaker::GenApi::EVisibility **)

static ToString(*ValueStr, pValue*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **pValue** (*Spinnaker::GenApi::EVisibility **)
- **gcstring** (*ToString(Value) ->*)
- **Value** (*enum Spinnaker::GenApi::EVisibility*)

property thisown

The membership flag

class PySpin.PySpin.EYesNoClass

Bases: object

Holds conversion methods for the standard namespace enumeration.

C++ includes: EnumClasses.h

static FromString(ValueStr, pValue) → bool

Parameters

- ValueStr (Spinnaker::GenICam::gcstring const &)
- pValue (Spinnaker::GenApi::EYesNo *)

static ToString(ValueStr, pValue)

Parameters

- ValueStr (Spinnaker::GenICam::gcstring &)
- pValue (Spinnaker::GenApi::EYesNo *)
- gcstring (ToString(Value) ->)
- Value (enum Spinnaker::GenApi::EYesNo)

property thisown

The membership flag

PySpin.PySpin.EatComments(_is) → std::istream &

Parameters

- is (std::istream &)
- SPINNAKER_API
- &is) (std::istream& Spinnaker::GenApi::EatComments(std::istream)
- '#'. (Helper function ignoring lines starting with comment character)

class PySpin.PySpin.EnumEntryNode(*args, **kwargs)

Bases: IEnumEntry, ValueNode

Interface for string properties.

C++ includes: EnumEntryNode.h

GetNumericValue(self) → double

virtual double Spinnaker::GenApi::EnumEntryNode::GetNumericValue()

Get double number associated with the entry

GetSymbolic(self) → gcstring

virtual GenICam::gcstring Spinnaker::GenApi::EnumEntryNode::GetSymbolic() const

Get symbolic enum value

GetValue(self) → int64_t

virtual int64_t Spinnaker::GenApi::EnumEntryNode::GetValue()

Get numeric enum value

IsSelfClearing(*self*) → bool

virtual bool Spinnaker::GenApi::EnumEntryNode::IsSelfClearing()

Indicates if the corresponding EnumEntry is self clearing

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::EnumEntryNode::SetReference**(*INode (virtual void)*)
- ***pBase**)
- **EnumEntry** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.PySpin.**EnumNode**(*args, **kwargs)

Bases: [IEnumeration](#), [ValueNode](#)

Interface for string properties.

C++ includes: EnumNode.h

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)
- **Spinnaker::GenApi::EnumNode::GetCurrentEntry**(*bool (virtual IEnumEntry*)*)
- **Verify=false**
- **IgnoreCache=false**) (*bool*)
- **entry** (*Get the current*)

GetEntries(*self*)

virtual void Spinnaker::GenApi::EnumNode::GetEntries(NodeList_t &Entries)

Get list of entry nodes

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*Get an entry node by its*)
- **virtual**
- **int64_t** (*IEnumEntry* Spinnaker::GenApi::EnumNode::GetEntry(const)*)
- **IntValue**)
- **IntValue**

GetEntryByName(*self*, *Symbolic*) → *IEnumEntry*

Parameters

- **Symbolic** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenApi::EnumNode::GetEntryByName**(*const IEnumEntry**) (virtual)
- **&Symbolic** (*GenICam::gcstring*)
- **name** (*Get an entry node by*)

GetIntValue(*self*, *Verify=False*, *IgnoreCache=False*) → *int64_t*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::EnumNode::GetIntValue**(*bool (virtual int64_t)*)
- **Verify=false**
- **IgnoreCache=false** (*bool*)
- **value** (*Get integer node*)
- **Parameters**
- **-----**
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

GetSymbolics(*self*, *Symbolics*)

Parameters

- **Symbolics** (*Spinnaker::GenApi::StringList_t &*)
- **Spinnaker::GenApi::EnumNode::GetSymbolics**(*StringList_t (virtual void)*)
- **&Symbolics**)
- **Values** (*Get list of symbolic*)

SetIntValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*virtual void Spinnaker::GenApi::EnumNode::SetIntValue(int64_t)*)
- **Verify** (*bool*)
- **Value**

:param : :param bool Verify=true): :param Set integer node value: :param Parameters: :param -----:
:param Value: :type Value: The value to set :param Verify: :type Verify: Enables AccessMode and Range
verification (default = true)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::EnumNode::SetReference(INode)*
—
- **Enumeration** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.PySpin.**EventHandler**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::EventHandler class.

GetEventPayloadData(*self*) → PyObject *

GetEventPayloadDataSize(*self*) → size_t const

GetEventType(*self*) → Spinnaker::EventType

SetEventType(*self*, *eventType*)

Parameters

eventType (*enum Spinnaker::EventType*)

property thisown

The membership flag

class PySpin.PySpin.**FloatNode**(*args, **kwargs)

Bases: *IFloat*, *ValueNode*

Interface for string properties.

C++ includes: FloatNode.h

GetDisplayNotation(*self*) → Spinnaker::GenApi::EDisplayNotation

virtual EDisplayNotation Spinnaker::GenApi::FloatNode::GetDisplayNotation() const

Get the way the float should be converted to a string

GetDisplayPrecision(*self*) → int64_t

virtual int64_t Spinnaker::GenApi::FloatNode::GetDisplayPrecision() const

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

GetEnumAlias(*self*) → *IEnumeration*

IEnumeration* Spinnaker::GenApi::FloatNode::GetEnumAlias()

gets the interface of an alias node.

GetInc(*self*) → double

virtual double Spinnaker::GenApi::FloatNode::GetInc()

Get the constant increment if there is any

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

virtual EIncMode Spinnaker::GenApi::FloatNode::GetIncMode()

Get increment mode

GetIntAlias(*self*) → *Integer*

Integer* Spinnaker::GenApi::FloatNode::GetIntAlias()

gets the interface of an alias node.

GetListOfValidValues(*self*, *bounded=True*) → *double_autovector_t*

Parameters

- **bounded** (*bool*)
- **virtual**
- **double_autovector_t**
- **bounded=true**) (*Spinnaker::GenApi::FloatNode::GetListOfValidValues*(*bool*)
- **value** (*Get list of valid*)

GetMax(*self*) → *double*

virtual double Spinnaker::GenApi::FloatNode::GetMax()

Get maximum value allowed

GetMin(*self*) → *double*

virtual double Spinnaker::GenApi::FloatNode::GetMin()

Get minimum value allowed

GetRepresentation(*self*) → *Spinnaker::GenApi::ERepresentation*

virtual ERepresentation Spinnaker::GenApi::FloatNode::GetRepresentation()

Get recommended representation

GetUnit(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::FloatNode::GetUnit() const

Get the physical unit name

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *double*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::FloatNode::GetValue**(*bool* (*virtual double*)
- **Verify=false**
- **IgnoreCache=false**) (*bool*)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)

- **read** (*The value*)

HasInc(*self*) → bool

virtual bool Spinnaker::GenApi::FloatNode::HasInc()

True if the float has a constant increment

ImposeMax(*self*, *Value*)

Parameters

- **Value** (*double*)
- **Value** (*virtual void Spinnaker::GenApi::FloatNode::ImposeMax(double)*)
- **value** (*Restrict maximum*)

ImposeMin(*self*, *Value*)

Parameters

- **Value** (*double*)
- **Value** (*virtual void Spinnaker::GenApi::FloatNode::ImposeMin(double)*)
- **value** (*Restrict minimum*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::FloatNode::SetReference(INode)*
—
- **Float** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*The value to set*)
- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Value**
- **bool**
- **Verify=true)**
- **value** (*Set node*)
- **Parameters**
- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.PySpin.**FloatRegNode**(*args, **kwargs)

Bases: [FloatNode](#), [RegisterNode](#)

Interface for string properties.

C++ includes: FloatRegNode.h

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::FloatRegNode::SetReference(INode** (*virtual void*)
- ***pBase**)
- **Value** (*overload SetReference for*)

property thisown

The membership flag

PySpin.PySpin.**GetErrorMessage**() → char const *

PySpin.PySpin.**GetFiles**(FileTemplate, DirectoriesOnly=False)

Parameters

- **FileTemplate** (*Spinnaker::GenICam::gcstring const &*)
- **DirectoriesOnly** (*bool const*)
- **SPINNAKER_API**
- **&FileTemplate** (*void Spinnaker::GenICam::GetFiles(const gcstring*)

:param : :param gcstring_vector &FileNames: :param const bool DirectoriesOnly=false): :param Gets a list of files or directories matching a given FileTemplate:

PySpin.PySpin.**GetGenICamCLProtocolFolder**() → *gcstring*

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.

PySpin.PySpin.**GetGenICamCacheFolder**() → *gcstring*

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.

PySpin.PySpin.**GetGenICamLogConfig**() → *gcstring*

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.

PySpin.PySpin.**GetInterfaceName**(pBase) → *gcstring*

Parameters

- **pBase** (*Spinnaker::GenApi::IBase **)
- ***pBase** (*GenICam::gcstring Spinnaker::GenApi::GetInterfaceName(**IBase**) –*
- **DEPRICATED** (*Returns the name of the main interface as string*)
- **use**
- **instead** (*IBase::GetPrincipalInterfaceType()*)

PySpin.PySpin.**GetModulePathFromFunction**(pFunction) → *gcstring*

Parameters

- **pFunction** (*void **)
- **gcstring** (*SPINNAKER_API*)
- ***pFunction** (*Spinnaker::GenICam::GetModulePathFromFunction(void) –*
- **only** (*true = only subdirectories (ex . and ..) are retrieved; false =>*
- **retrieved** (*files are*)
- **given** (*Gets the full path to the module (DLL/SO) containing the*
- **found.** (*pFunction; empty string if not*)

PySpin.PySpin.**GetValueOfEnvironmentVariable**(VariableName) → *gcstring*

Parameters

- **VariableName** (*Spinnaker::GenICam::gcstring const &*)
- **GetValueOfEnvironmentVariable(VariableName**
- **bool** (*SPINNAKER_API*)
- **VariableName**
- **VariableContent** (*Spinnaker::GenICam::gcstring &*)
- **bool**
- **gcstring** (*Spinnaker::GenICam::GetValueOfEnvironmentVariable(const)*
- **&VariableName**
- **&VariableContent**) (*gcstring*)
- **environment** (*Retrieve the value of an environment variable true if*
- **found** (*variable was*)
- **false** (*otherwise*)

class PySpin.PySpin.H264Option

Bases: object

Options for saving H264 files.

C++ includes: SpinVideoDefs.h

property bitrate

property `crf`

property `frameRate`

property `height`

property `reserved`

property `thisown`

The membership flag

property `useMP4`

property `width`

class `PySpin.PySpin.IBase(*args, **kwargs)`

Bases: `object`

Proxy of C++ Spinnaker::GenApi::IBase class.

GetAccessMode(*self*) → `Spinnaker::GenApi::EAccessMode`

property `thisown`

The membership flag

class `PySpin.PySpin.IBoolean(*args, **kwargs)`

Bases: `IValue`

Proxy of C++ Spinnaker::GenApi::IBoolean class.

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → `bool`

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*bool*)
- **Verify** (*bool*)

property `thisown`

The membership flag

class `PySpin.PySpin.ICameraBase(*args, **kwargs)`

Bases: `object`

Proxy of C++ Spinnaker::ICameraBase class.

BeginAcquisition(*self*)

DeInit(*self*)

DiscoverMaxPacketSize(*self*) → `unsigned int`

EndAcquisition(*self*)

ForceIP(*self*)

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

GetActiveNumDataStreams(*self*) → unsigned int

GetBufferOwnership(*self*) → Spinnaker::BufferOwnership

GetDeviceID(*self*) → *gcstring*

GetGuiXml(*self*) → *gcstring*

GetNextImage(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE, *streamIndex*=0) → *ImagePtr*

Parameters

- **grabTimeout** (uint64_t)
- **streamIndex** (uint64_t)

GetNextImageSync(*self*, *grabTimeout*=EVENT_TIMEOUT_INFINITE) → *ImageList*

Parameters

grabTimeout (uint64_t)

GetNodeMap(*self*) → *INodeMap*

GetNumDataStreams(*self*) → unsigned int

GetNumImagesInUse(*self*) → unsigned int

GetTLDeviceNodeMap(*self*) → *INodeMap*

GetTLStreamNodeMap(*self*, *streamIndex*) → *INodeMap*

Parameters

streamIndex (uint64_t)

GetUniqueID(*self*) → *gcstring*

GetUserBufferCount(*self*) → uint64_t

GetUserBufferSize(*self*) → uint64_t

GetUserBufferTotalSize(*self*) → uint64_t

Init(*self*)

IsInitialized(*self*) → bool

IsStreaming(*self*) → bool

IsValid(*self*) → bool

ReadPort(*self*, *iAddress*) → PyObject *

Parameters

iAddress (uint64_t)

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::ImageEventHandler &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **eventName**)
- **evtHandlerToRegister**
- **eventName** (*Spinnaker::GenICam::gcstring const &*)
- **RegisterEventHandler**(*self*
- **evtHandlerToRegister**
- **streamIndex**)
- **evtHandlerToRegister**
- **streamIndex** (*uint64_t*)

SetBufferOwnership(*self*, *mode*)

Parameters

- **mode** (*enum Spinnaker::BufferOwnership const*)

SetUserBuffers(*self*, *pMemBuffers*, *totalSize*)

Parameters

- **pMemBuffers** (*void *const*)
- **totalSize** (*uint64_t*)
- **SetUserBuffers**(*self*
- **ppMemBuffers** (*void **const*)
- **bufferCount** (*uint64_t const*)
- **bufferSize**)
- **ppMemBuffers**
- **bufferCount**
- **bufferSize** (*uint64_t const*)

property TLDevice

property TLStream

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

- **evtHandlerToUnregister** (*Spinnaker::EventHandler &*)

WritePort(*self*, *iAddress*, *pBuffer*)

Parameters

- **iAddress** (*uint64_t*)
- **pBuffer** (*uint32_t*)

property thisown

The membership flag

class PySpin.PySpin.ICameraList(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::ICameraList class.

Add(self, camera)**Parameters****camera** (Spinnaker::CameraPtr)**Append**(self, list)**Parameters****list** (Spinnaker::CameraList const &)**Clear**(self)**GetByDeviceID**(self, deviceID) → CameraPtr**Parameters****deviceID** (std::string)**GetByIndex**(self, index) → CameraPtr**Parameters****index** (unsigned int)**GetBySerial**(self, serialNumber) → CameraPtr**Parameters****serialNumber** (std::string)**GetSize**(self) → unsigned int**Remove**(self, camera)**Parameters****camera** (Spinnaker::CameraPtr)**RemoveByDeviceID**(self, deviceID)**Parameters****deviceID** (std::string)**RemoveByIndex**(self, index)**Parameters****index** (unsigned int)**RemoveBySerial**(self, serialNumber)**Parameters****serialNumber** (std::string)**property thisown**

The membership flag

```
class PySpin.PySpin.ICategory(*args, **kwargs)
    Bases: IValue
    Proxy of C++ Spinnaker::GenApi::ICategory class.
    GetFeatures(self)

    property thisown
        The membership flag

class PySpin.PySpin.IChunkData(*args, **kwargs)
    Bases: object
    Proxy of C++ Spinnaker::IChunkData class.
    GetBlackLevel(self) → float64_t
    GetCRC(self) → int64_t
    GetCompressionMode(self) → int64_t
    GetCompressionRatio(self) → float64_t
    GetCounterValue(self) → int64_t
    GetCurrentDataRate(self) → int64_t
    GetEnable(self) → bool
    GetEncoderValue(self) → int64_t
    GetExposureEndLineStatusAll(self) → int64_t
    GetExposureTime(self) → float64_t
    GetFrameID(self) → int64_t
    GetGain(self) → float64_t
    GetHeight(self) → int64_t
    GetImage(self) → int64_t
    GetInferenceBoundingBoxResult(self) → InferenceBoundingBoxResult
    GetInferenceConfidence(self) → float64_t
    GetInferenceFrameId(self) → int64_t
    GetInferenceResult(self) → int64_t
    GetLinePitch(self) → int64_t
    GetLineStatusAll(self) → int64_t
    GetModeActive(self) → bool
    GetOffsetX(self) → int64_t
    GetOffsetY(self) → int64_t
```



```

GetPartSelector(self) → int64_t
GetPixelDynamicRangeMax(self) → int64_t
GetPixelDynamicRangeMin(self) → int64_t
GetScan3dAxisMax(self) → float64_t
GetScan3dAxisMin(self) → float64_t
GetScan3dCoordinateOffset(self) → float64_t
GetScan3dCoordinateReferenceValue(self) → float64_t
GetScan3dCoordinateScale(self) → float64_t
GetScan3dInvalidDataFlag(self) → bool
GetScan3dInvalidDataValue(self) → float64_t
GetScan3dTransformValue(self) → float64_t
GetScanLineSelector(self) → int64_t
GetSequencerSetActive(self) → int64_t
GetSerialData(self) → uint8_t *
GetSerialDataLength(self) → int64_t
GetSerialReceiveOverflow(self) → bool
GetStreamChannelID(self) → int64_t
GetTimerValue(self) → float64_t
GetTimestamp(self) → int64_t
GetTimestampLatchValue(self) → int64_t
GetTransferBlockID(self) → int64_t
GetTransferQueueCurrentBlockCount(self) → int64_t
GetWidth(self) → int64_t
SetChunks(self, pNodeMap)

```

Parameters

pNodeMap (*Spinnaker::GenApi::INodeMap &*)

property thisown

The membership flag

```
class PySpin.PySpin.ICommand(*args, **kwargs)
```

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::ICommand class.

Execute(*self*, *Verify*=*True*)

Parameters

Verify (*bool*)

IsDone(*self*, *Verify*=*True*) → *bool*

Parameters

Verify (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IDestroy**(*args, **kwargs)

Bases: *object*

Proxy of C++ Spinnaker::GenApi::IDestroy class.

Destroy(*self*)

property thisown

The membership flag

class PySpin.PySpin.**IDeviceArrivalEventHandler**(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::IDeviceArrivalEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.PySpin.**IDeviceEventHandler**(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::IDeviceEventHandler class.

GetDeviceEventId(*self*) → *uint64_t*

GetDeviceEventName(*self*) → *gcstring*

OnDeviceEvent(*self*, *eventName*)

Parameters

eventName (*Spinnaker::GenICam::gcstring*)

property thisown

The membership flag

class PySpin.PySpin.**IDeviceInfo**(*args, **kwargs)

Bases: *object*

Proxy of C++ Spinnaker::GenApi::IDeviceInfo class.

GetDeviceVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t &*)

GetGenApiVersion(*self*, *Version*, *Build*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t* &)
- **Build** (*uint16_t* &)

GetModelName(*self*) → *gcstring*

GetProductGuid(*self*) → *gcstring*

GetSchemaVersion(*self*, *Version*)

Parameters

Version (*Spinnaker::GenICam::Version_t* &)

GetStandardNameSpace(*self*) → *gcstring*

GetToolTip(*self*) → *gcstring*

GetVendorName(*self*) → *gcstring*

GetVersionGuid(*self*) → *gcstring*

property thisown

The membership flag

class *PySpin.PySpin.IDeviceRemovalEventHandler*(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ *Spinnaker::IDeviceRemovalEventHandler* class.

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumEntry*(*args, **kwargs)

Bases: *IValue*

Proxy of C++ *Spinnaker::GenApi::IEnumEntry* class.

GetNumericValue(*self*) → double

GetSymbolic(*self*) → *gcstring*

GetValue(*self*) → *int64_t*

IsSelfClearing(*self*) → bool

property thisown

The membership flag

class *PySpin.PySpin.IEnumReference*(*args, **kwargs)

Bases: object

Proxy of C++ *Spinnaker::GenApi::IEnumReference* class.

SetEnumReference(*self*, *Index*, *Name*)

Parameters

- **Index** (*int*)
- **Name** (*Spinnaker::GenICam::gcstring*)

SetNumEnums(*self*, *NumEnums*)

Parameters

NumEnums (*int*)

property **thisown**

The membership flag

class PySpin.PySpin.IEnumeration(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IEnumeration class.

GetCurrentEntry(*self*, *Verify*=False, *IgnoreCache*=False) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntries(*self*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

IntValue (*int64_t const*)

GetEntryByName(*self*, *Symbolic*) → *IEnumEntry*

Parameters

Symbolic (*Spinnaker::GenICam::gcstring const &*)

GetIntValue(*self*, *Verify*=False, *IgnoreCache*=False) → *int64_t*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetSymbolics(*self*, *Symbolics*)

Parameters

Symbolics (*Spinnaker::GenApi::StringList_t &*)

SetIntValue(*self*, *Value*, *Verify*=True)

Parameters

- **Value** (*int64_t*)
- **Verify** (*bool*)

property **thisown**

The membership flag

class PySpin.PySpin.IEnumerationT_AcquisitionModeEnums(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AcquisitionModeEnums > class.

GetCurrentEntry(self, Verify=False, IgnoreCache=False) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(self, IntValue) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(self
- **IEnumEntry** (Value) ->)
- **Value** (enum Spinnaker::AcquisitionModeEnums const)

GetValue(self, Verify=False, IgnoreCache=False) → Spinnaker::AcquisitionModeEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(self, Value, Verify=True)

Parameters

- **Value** (enum Spinnaker::AcquisitionModeEnums)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.IEnumerationT_AcquisitionStatusSelectorEnums(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AcquisitionStatusSelectorEnums > class.

GetCurrentEntry(self, Verify=False, IgnoreCache=False) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(self, IntValue) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(self
- **IEnumEntry** (Value) ->)
- **Value** (enum Spinnaker::AcquisitionStatusSelectorEnums const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AcquisitionStatusSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AcquisitionStatusSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ActionSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ActionSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ActionSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ActionSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ActionSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ActionUnconditionalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ActionUnconditionalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ActionUnconditionalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ActionUnconditionalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ActionUnconditionalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_AdcBitDepthEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< AdcBitDepthEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AdcBitDepthEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AdcBitDepthEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AdcBitDepthEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_AutoAlgorithmSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoAlgorithmSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoAlgorithmSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_AutoExposureControlPriorityEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureControlPriorityEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureControlPriorityEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AutoExposureControlPriorityEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureControlPriorityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_AutoExposureLightingModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< AutoExposureLightingModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureLightingModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureLightingModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureLightingModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_AutoExposureMeteringModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureMeteringModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureMeteringModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureMeteringModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureMeteringModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_AutoExposureTargetGreyValueAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< AutoExposureTargetGreyValueAutoEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::AutoExposureTargetGreyValueAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_BalanceRatioSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BalanceRatioSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BalanceRatioSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BalanceRatioSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BalanceRatioSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BalanceWhiteAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BalanceWhiteAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BalanceWhiteAutoEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BalanceWhiteAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BalanceWhiteAutoProfileEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoProfileEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BalanceWhiteAutoProfileEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_BinningHorizontalModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BinningHorizontalModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BinningHorizontalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BinningHorizontalModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningHorizontalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BinningSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BinningSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BinningSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BinningSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BinningVerticalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BinningVerticalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BinningVerticalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BinningVerticalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BinningVerticalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_BlackLevelAutoBalanceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BlackLevelAutoBalanceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BlackLevelAutoBalanceEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BlackLevelAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BlackLevelAutoEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BlackLevelAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BlackLevelAutoEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BlackLevelSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BlackLevelSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BlackLevelSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::BlackLevelSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BlackLevelSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionAutoEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BsiFlatFieldCorrectionAutoEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionGainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkBlackLevelSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkBlackLevelSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkCounterSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkCounterSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkCounterSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkCounterSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkCounterSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkEncoderSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkEncoderSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkEncoderSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkEncoderSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkEncoderSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkEncoderStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkEncoderStatusEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkEncoderStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkEncoderStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkEncoderStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkExposureTimeSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkExposureTimeSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkExposureTimeSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkExposureTimeSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkGainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkGainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkGainSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkGainSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkGainSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkImageComponentEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkImageComponentEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkImageComponentEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkImageComponentEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkImageComponentEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkPixelFormatEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkPixelFormatEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkPixelFormatEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkPixelFormatEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkPixelFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkRegionIDEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkRegionIDEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkRegionIDEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkRegionIDEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkRegionIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag


```
class PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums(*args,
                                                                              **kwargs)
```

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateReferenceSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

```
class PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSelectorEnums(*args, **kwargs)
```

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*

- **IEnumEntry** (*Value*) ->
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dCoordinateSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystemEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dCoordinateSystemEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

```
class PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums(*args,
                                                                            **kwargs)
```

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

```
class PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums(*args,
                                                                                **kwargs)
```

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkScan3dDistanceUnitEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkScan3dDistanceUnitEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ChunkScan3dDistanceUnitEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dDistanceUnitEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dDistanceUnitEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkScan3dOutputModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkScan3dOutputModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkScan3dOutputModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkScan3dOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkScan3dOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ChunkSourceIDEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ChunkSourceIDEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkSourceIDEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkSourceIDEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkSourceIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkTimerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkTimerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkTimerSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkTimerSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkTimerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ChunkTransferStreamIDEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ChunkTransferStreamIDEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ChunkTransferStreamIDEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ChunkTransferStreamIDEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ChunkTransferStreamIDEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ClConfigurationEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ClConfigurationEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ClConfigurationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ClConfigurationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ClConfigurationEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ClTimeSlotsCountEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ClTimeSlotsCountEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ClTimeSlotsCountEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ClTimeSlotsCountEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ClTimeSlotsCountEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ColorTransformationSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ColorTransformationSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ColorTransformationSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ColorTransformationSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ColorTransformationSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ColorTransformationValueSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ColorTransformationValueSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ColorTransformationValueSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ComponentDestinationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ComponentDestinationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ComponentDestinationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::ComponentDestinationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ComponentDestinationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ComponentSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ComponentSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ComponentSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ComponentSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ComponentSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CompressionSaturationPriorityEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CompressionSaturationPriorityEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CompressionSaturationPriorityEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CompressionSaturationPriorityEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CompressionSaturationPriorityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterEventActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterEventActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterEventActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterEventActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterEventActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterEventSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterEventSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterEventSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterEventSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterEventSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CounterResetActivationEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CounterResetActivationEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterResetActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterResetActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterResetActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterResetSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterResetSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterResetSourceEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CounterResetSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterResetSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CounterStatusEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CounterStatusEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterTriggerActivationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CounterTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::CounterTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CounterTriggerSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CounterTriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CounterTriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CounterTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CounterTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CxpConnectionTestModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CxpConnectionTestModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpConnectionTestModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpConnectionTestModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::CxpConnectionTestModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CxpLinkConfigurationEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationEnums >* class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::CxpLinkConfigurationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::CxpLinkConfigurationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::CxpLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CxpLinkConfigurationPreferredEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationPreferredEnums >* class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpLinkConfigurationPreferredEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationStatusEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::CxpLinkConfigurationStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpLinkConfigurationStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::CxpLinkConfigurationStatusEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_CxpPoCxpStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< CxpPoCxpStatusEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::CxpPoCxpStatusEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::CxpPoCxpStatusEnums*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::CxpPoCxpStatusEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DecimationHorizontalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DecimationHorizontalModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DecimationHorizontalModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DecimationHorizontalModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DecimationHorizontalModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DecimationSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DecimationSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DecimationSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DecimationSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::DecimationSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DecimationVerticalModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DecimationVerticalModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::DecimationVerticalModeEnums* const)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::DecimationVerticalModeEnums*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::DecimationVerticalModeEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DefectCorrectionModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DefectCorrectionModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DefectCorrectionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DefectCorrectionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DefectCorrectionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeinterlacingEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeinterlacingEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeinterlacingEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeinterlacingEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeinterlacingEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceAccessStatusEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceAccessStatusEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceAccessStatusEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceAccessStatusEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceAccessStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceCharacterSetEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceCharacterSetEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceCharacterSetEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceCharacterSetEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceCharacterSetEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceClockSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceClockSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceClockSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceClockSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceClockSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceConnectionStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceConnectionStatusEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceConnectionStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceConnectionStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceConnectionStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceCurrentSpeedEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceCurrentSpeedEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceCurrentSpeedEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceCurrentSpeedEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceCurrentSpeedEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceEndiannessMechanismEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceEndiannessMechanismEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceEndiannessMechanismEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceEndiannessMechanismEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceEndiannessMechanismEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceIndicatorModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceIndicatorModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceIndicatorModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceIndicatorModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceIndicatorModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceLinkHeartbeatModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceLinkHeartbeatModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceLinkHeartbeatModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceLinkThroughputLimitModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceLinkThroughputLimitModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceLinkThroughputLimitModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DevicePowerSupplySelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DevicePowerSupplySelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DevicePowerSupplySelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceRegistersEndiannessEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceRegistersEndiannessEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceRegistersEndiannessEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceScanTypeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceScanTypeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceScanTypeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceScanTypeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceScanTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceSensorChromaEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceSensorChromaEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSensorChromaEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceSensorChromaEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSensorChromaEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceSerialPortBaudRateEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceSerialPortBaudRateEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceSerialPortSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceSerialPortSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceSerialPortSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceStreamChannelEndiannessEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceStreamChannelEndiannessEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceStreamChannelEndiannessEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::DeviceStreamChannelEndiannessEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceStreamChannelEndiannessEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceStreamChannelTypeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceStreamChannelTypeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceStreamChannelTypeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceStreamChannelTypeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceStreamChannelTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceTLTypeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceTLTypeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTLTypeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTLTypeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTLTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceTapGeometryEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTapGeometryEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTapGeometryEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceTapGeometryEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTapGeometryEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceTemperatureSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTemperatureSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_DeviceTypeEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< DeviceTypeEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTypeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::DeviceTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_DeviceTypeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< DeviceTypeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::DeviceTypeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::DeviceTypeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::DeviceTypeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EncoderModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_EncoderOutputModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< EncoderOutputModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderOutputModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EncoderResetActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderResetActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderResetActivationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::EncoderResetActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::EncoderResetActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EncoderResetSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderResetSourceEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderResetSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderResetSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderResetSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_EncoderSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< EncoderSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EncoderSourceAEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderSourceAEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSourceAEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderSourceAEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSourceAEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EncoderSourceBEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EncoderSourceBEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderSourceBEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderSourceBEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderSourceBEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_EncoderStatusEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< EncoderStatusEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EncoderStatusEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EncoderStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EncoderStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EventNotificationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EventNotificationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EventNotificationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EventNotificationEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EventNotificationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_EventSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< EventSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::EventSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::EventSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::EventSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ExposureActiveModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ExposureActiveModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureActiveModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureActiveModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ExposureActiveModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ExposureAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureAutoEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureAutoEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::ExposureAutoEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::ExposureAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ExposureModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExposureModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class `PySpin.PySpin.IEnumerationT_ExposureTimeModeEnums(*args, **kwargs)`

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ExposureTimeModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureTimeModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExposureTimeModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExposureTimeModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ExposureTimeSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExposureTimeSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExposureTimeSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureTimeSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExposureTimeSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ExternalVoltageSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ExternalVoltageSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ExternalVoltageSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ExternalVoltageSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ExternalVoltageSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_FLIRFilterDriverStatusEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< FLIRFilterDriverStatusEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FLIRFilterDriverStatusEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FLIRFilterDriverStatusEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::FLIRFilterDriverStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_FfcModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FfcModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FfcModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::FfcModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::FfcModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_FileOpenModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileOpenModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileOpenModeEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileOpenModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::FileOpenModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_FileOperationSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< FileOperationSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileOperationSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileOperationSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::FileOperationSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_FileOperationStatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileOperationStatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::FileOperationStatusEnums* const)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::FileOperationStatusEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::FileOperationStatusEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_FileSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< FileSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::FileSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::FileSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::FileSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GUIXMLLocationEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GUIXMLLocationEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GUIXMLLocationEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GUIXMLLocationEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GUIXMLLocationEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GainAutoBalanceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainAutoBalanceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainAutoBalanceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GainAutoBalanceEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GainAutoBalanceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GainAutoEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainAutoEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainAutoEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GainAutoEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GainAutoEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GainConversionEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GainConversionEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GainConversionEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GainConversionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::GainConversionEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GainSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GainSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::GainSelectorEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GainSelectorEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::GainSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GenICamXMLLocationEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GenICamXMLLocationEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GenICamXMLLocationEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GenICamXMLLocationEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GenICamXMLLocationEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GevCCPEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCCPEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCCPEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevCCPEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevCCPEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevCCPEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCCPEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCCPEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::GevCCPEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevCCPEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevCurrentPhysicalLinkConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::GevCurrentPhysicalLinkConfigurationEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GevGVCPExtendedStatusCodesSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevGVSPExtendedIDModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::GevGVSPExtendedIDModeEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevGVSPExtendedIDModeEnums*

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::GevGVSPExtendedIDModeEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevIEEE1588ClockAccuracyEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIEEE1588ClockAccuracyEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588ClockAccuracyEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GevIEEE1588ModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GevIEEE1588ModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) →)
- **Value** (*enum Spinnaker::GevIEEE1588ModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588ModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588ModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevIEEE1588StatusEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIEEE1588StatusEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIEEE1588StatusEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::GevIEEE1588StatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588StatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevIEEE1588StatusLatchedEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevIEEE1588StatusLatchedEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIEEE1588StatusLatchedEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIEEE1588StatusLatchedEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevIEEE1588StatusLatchedEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GevIPConfigurationStatusEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GevIPConfigurationStatusEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevIPConfigurationStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevIPConfigurationStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::GevIPConfigurationStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevPhysicalLinkConfigurationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevPhysicalLinkConfigurationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::GevPhysicalLinkConfigurationEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::GevPhysicalLinkConfigurationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::GevPhysicalLinkConfigurationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_GevSCPDDirectionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< GevSCPDDirectionEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevSCPDirectionEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevSCPDirectionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::GevSCPDirectionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_GevSupportedOptionSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< GevSupportedOptionSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::GevSupportedOptionSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::GevSupportedOptionSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::GevSupportedOptionSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ImageComponentSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageComponentSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ImageComponentSelectorEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageComponentSelectorEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ImageComponentSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ImageCompressionJPEGFormatOptionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::ImageCompressionJPEGFormatOptionEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_ImageCompressionModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< ImageCompressionModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::ImageCompressionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::ImageCompressionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ImageCompressionModeEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_ImageCompressionRateOptionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< ImageCompressionRateOptionEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::ImageCompressionRateOptionEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageCompressionRateOptionEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::ImageCompressionRateOptionEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_InterfaceTypeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< InterfaceTypeEnum > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::InterfaceTypeEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::InterfaceTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::InterfaceTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_LUTSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LUTSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LUTSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LUTSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LUTSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LensShadingCoefficientActiveSetEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LensShadingCoefficientActiveSetEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LensShadingCoefficientActiveSetEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::LensShadingCoefficientActiveSetEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::LensShadingCoefficientActiveSetEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LensShadingCorrectionModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LensShadingCorrectionModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LensShadingCorrectionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LensShadingCorrectionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LensShadingCorrectionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_LineFormatEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LineFormatEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineFormatEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineFormatEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LineInputFilterSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineInputFilterSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineInputFilterSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineInputFilterSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineInputFilterSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LineModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_LineSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LineSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LineSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LineSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LineSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LineSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LineSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LogicBlockLUTInputActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTInputActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_LogicBlockLUTInputSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTInputSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LogicBlockLUTInputSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTInputSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_LogicBlockLUTSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< LogicBlockLUTSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockLUTSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_LogicBlockSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< LogicBlockSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::LogicBlockSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::LogicBlockSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::LogicBlockSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_MultiRoiConfigurationInvalidReasonEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< MultiRoiConfigurationInvalidReasonEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::MultiRoiConfigurationInvalidReasonEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →

Spinnaker::MultiRoiConfigurationInvalidReasonEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::MultiRoiConfigurationInvalidReasonEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_MultiRoiSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< MultiRoiSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)

- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::MultiRoiSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::MultiRoiSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::MultiRoiSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_POEStatusEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< POEStatusEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::POEStatusEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::POEStatusEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::POEStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_PixelColorFilterEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelColorFilterEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelColorFilterEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::PixelColorFilterEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::PixelColorFilterEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_PixelFormatEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelFormatEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelFormatEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::PixelFormatEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::PixelFormatEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_PixelFormatInfoSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< PixelFormatInfoSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::PixelFormatInfoSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_PixelSizeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< PixelSizeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::PixelSizeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::PixelSizeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::PixelSizeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_RegionDestinationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< RegionDestinationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionDestinationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RegionDestinationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::RegionDestinationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_RegionModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< RegionModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RegionModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::RegionModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_RegionSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< RegionSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RegionSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::RegionSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::RegionSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_RgbTransformLightSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< RgbTransformLightSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::RgbTransformLightSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::RgbTransformLightSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::RgbTransformLightSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_Scan3dCoordinateReferenceSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateReferenceSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateReferenceSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_Scan3dCoordinateSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_Scan3dCoordinateSystemEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateSystemEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemReferenceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dCoordinateSystemReferenceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dCoordinateSystemReferenceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::Scan3dCoordinateSystemReferenceEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_Scan3dCoordinateTransformSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateTransformSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::Scan3dCoordinateTransformSelectorEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) →
Spinnaker::Scan3dCoordinateTransformSelectorEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::Scan3dCoordinateTransformSelectorEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_Scan3dDistanceUnitEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< Scan3dDistanceUnitEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)

- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dDistanceUnitEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dDistanceUnitEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dDistanceUnitEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_Scan3dOutputModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< Scan3dOutputModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::Scan3dOutputModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::Scan3dOutputModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::Scan3dOutputModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SensorDigitizationTapsEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SensorDigitizationTapsEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorDigitizationTapsEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::SensorDigitizationTapsEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SensorDigitizationTapsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SensorShutterModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SensorShutterModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorShutterModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SensorShutterModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SensorShutterModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_SensorTapsEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SensorTapsEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SensorTapsEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SensorTapsEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SensorTapsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SequencerConfigurationModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerConfigurationModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerConfigurationModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::SequencerConfigurationModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SequencerConfigurationModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SequencerConfigurationValidEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerConfigurationValidEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerConfigurationValidEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerConfigurationValidEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerConfigurationValidEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_SequencerModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SequencerModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SequencerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SequencerSetValidEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerSetValidEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerSetValidEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::SequencerSetValidEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SequencerSetValidEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SequencerTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SequencerTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerTriggerActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_SequencerTriggerSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SequencerTriggerSourceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SequencerTriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SequencerTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SequencerTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SerialPortBaudRateEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortBaudRateEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortBaudRateEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortBaudRateEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortBaudRateEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SerialPortParityEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortParityEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortParityEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortParityEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortParityEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_SerialPortSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SerialPortSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SerialPortSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SerialPortStopBitsEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SerialPortStopBitsEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SerialPortStopBitsEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SerialPortStopBitsEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::SerialPortStopBitsEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_SoftwareSignalSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< SoftwareSignalSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SoftwareSignalSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::SoftwareSignalSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SoftwareSignalSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_SourceSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< SourceSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::SourceSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::SourceSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::SourceSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_StereoResolutionEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StereoResolutionEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StereoResolutionEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StereoResolutionEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::StereoResolutionEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_StreamBufferCountModeEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< StreamBufferCountModeEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StreamBufferCountModeEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StreamBufferCountModeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::StreamBufferCountModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_StreamBufferHandlingModeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StreamBufferHandlingModeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::StreamBufferHandlingModeEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::StreamBufferHandlingModeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (enum *Spinnaker::StreamBufferHandlingModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_StreamModeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< StreamModeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StreamModeEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StreamModeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::StreamModeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_StreamTypeEnum*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< StreamTypeEnum >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::StreamTypeEnum const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::StreamTypeEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::StreamTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TLTypeEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TLTypeEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TLTypeEnum const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::TLTypeEnum

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TLTypeEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnum**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TeledyneGigeVisionFilterDriverStatusEnum > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TeledyneGigeVisionFilterDriverStatusEnum*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TestPatternEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TestPatternEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TestPatternEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TestPatternEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TestPatternEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TestPatternGeneratorSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TestPatternGeneratorSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TestPatternGeneratorSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → *Spinnaker::TestPatternGeneratorSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TestPatternGeneratorSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TimerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TimerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TimerStatusEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TimerStatusEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerStatusEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerStatusEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TimerStatusEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TimerTriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerTriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerTriggerActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TimerTriggerActivationEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TimerTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TimerTriggerSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TimerTriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TimerTriggerSourceEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TimerTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TimerTriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TransferComponentSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferComponentSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t* *const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferComponentSelectorEnums* *const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferComponentSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferComponentSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferControlModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferControlModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferControlModeEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::TransferControlModeEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::TransferControlModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferOperationModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferOperationModeEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferOperationModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferOperationModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferOperationModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TransferQueueModeEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferQueueModeEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferQueueModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferQueueModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferQueueModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferStatusSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferStatusSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferStatusSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferStatusSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferStatusSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TransferTriggerActivationEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferTriggerActivationEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerActivationEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerActivationEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerActivationEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferTriggerModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferTriggerModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TransferTriggerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TransferTriggerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TransferTriggerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TransferTriggerSourceEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TransferTriggerSourceEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TransferTriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TransferTriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::TransferTriggerSourceEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TriggerActivationEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerActivationEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (int64_t const)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (enum *Spinnaker::TriggerActivationEnums* const)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerActivationEnums

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (enum *Spinnaker::TriggerActivationEnums*)
- **Verify** (bool)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TriggerModeEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerModeEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (bool)
- **IgnoreCache** (bool)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerModeEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerModeEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerModeEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_TriggerOverlapEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< TriggerOverlapEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerOverlapEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerOverlapEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerOverlapEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TriggerSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerSelectorEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_TriggerSourceEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< TriggerSourceEnums > class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::TriggerSourceEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::TriggerSourceEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerSourceEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_U3VCurrentSpeedEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< U3VCurrentSpeedEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::U3VCurrentSpeedEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::U3VCurrentSpeedEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::U3VCurrentSpeedEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_UserOutputSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< UserOutputSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserOutputSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::UserOutputSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::UserOutputSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_UserSetDefaultEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< UserSetDefaultEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserSetDefaultEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::UserSetDefaultEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::UserSetDefaultEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class *PySpin.PySpin.IEnumerationT_UserSetSelectorEnums*(*args, **kwargs)

Bases: *IEnumeration*, *IEnumReference*

Proxy of C++ *Spinnaker::GenApi::IEnumerationT< UserSetSelectorEnums >* class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → *IEnumEntry*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → *IEnumEntry*

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::UserSetSelectorEnums const*)

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *Spinnaker::UserSetSelectorEnums*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::UserSetSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IEnumerationT_WhiteClipSelectorEnums**(*args, **kwargs)

Bases: [IEnumeration](#), [IEnumReference](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT< WhiteClipSelectorEnums > class.

GetCurrentEntry(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → [IEnumEntry](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

GetEntry(*self*, *IntValue*) → [IEnumEntry](#)

Parameters

- **IntValue** (*int64_t const*)
- **GetEntry**(*self*
- **IEnumEntry** (*Value*) ->)
- **Value** (*enum Spinnaker::WhiteClipSelectorEnums const*)

GetValue(*self*, *Verify*=*False*, *IgnoreCache*=*False*) → Spinnaker::WhiteClipSelectorEnums

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*enum Spinnaker::WhiteClipSelectorEnums*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**IFloat**(*args, **kwargs)

Bases: [IValue](#)

Proxy of C++ Spinnaker::GenApi::IFloat class.

GetDisplayNotation(*self*) → Spinnaker::GenApi::EDisplayNotation

GetDisplayPrecision(*self*) → int64_t

GetInc(*self*) → double

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

GetListOfValidValues(*self*, *bounded=True*) → *double_autovector_t*

Parameters

bounded (*bool*)

GetMax(*self*) → double

GetMin(*self*) → double

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

GetUnit(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → double

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

HasInc(*self*) → bool

ImposeMax(*self*, *Value*)

Parameters

Value (*double*)

ImposeMin(*self*, *Value*)

Parameters

Value (*double*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*double*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.IImage(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IImage class.

CalculateChannelStatistics(*self*, *channel*) → *ChannelStatistics*

Parameters

channel (*enum Spinnaker::StatisticsChannel*)

CalculateStatistics(*self*, *pStatistics*)

Parameters

pStatistics (*Spinnaker::ImageStatistics &*)

CheckCRC(*self*) → bool

DeepCopy(*self*, *pSrcImage*)

Parameters

pSrcImage (*Spinnaker::ImagePtr* *const*)

GetBitsPerPixel(*self*) → *size_t*

GetBufferSize(*self*) → *size_t*

GetChunkData(*self*) → *ChunkData*

GetChunkLayoutId(*self*) → *uint64_t*

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetData(*self*)

GetData(*self*) → *PyObject **

GetDataAbsoluteMax(*self*) → *float*

GetDataAbsoluteMin(*self*) → *float*

GetFrameID(*self*) → *uint64_t*

GetHeight(*self*) → *size_t*

GetID(*self*) → *uint64_t*

GetImagePayloadType(*self*) → *Spinnaker::ImagePayloadType*

GetImageSize(*self*) → *size_t*

GetImageStatus(*self*) → *Spinnaker::ImageStatus*

GetNDArray(*self*) → *PyObject **

GetNumChannels(*self*) → *size_t*

GetPayloadType(*self*) → *size_t*

GetPixelFormat(*self*) → *Spinnaker::PixelFormatEnums*

GetPixelFormatIntType(*self*) → *Spinnaker::PixelFormatIntType*

GetPixelFormatName(*self*) → *gcstring*

GetPrivateData(*self*) → *void **

GetStreamIndex(*self*) → *uint64_t*

GetStride(*self*) → *size_t*

GetTLPayloadType(*self*) → *Spinnaker::TLPayloadType*

GetTLPixelFormat(*self*) → *uint64_t*

GetTLPixelFormatNamespace(*self*) → *Spinnaker::TLPixelFormatNamespace*

GetTimeStamp(*self*) → *uint64_t*

GetValidPayloadSize(*self*) → *size_t*

GetWidth(*self*) → size_t

GetXOffset(*self*) → size_t

GetXPadding(*self*) → size_t

GetYOffset(*self*) → size_t

GetYPadding(*self*) → size_t

HasCRC(*self*) → bool

HasChunkData(*self*) → bool

IsCompressed(*self*) → bool

IsInUse(*self*) → bool

IsIncomplete(*self*) → bool

Release(*self*)

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

Parameters

- **width** (size_t)
- **height** (size_t)
- **offsetX** (size_t)
- **offsetY** (size_t)
- **pixelFormat** (enum *Spinnaker::PixelFormatEnums*)
- **ResetImage**(self
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**)
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (void *)
- **ResetImage**(self
- **width**
- **height**

- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`
- `dataPayloadType` (*enum Spinnaker::TLPayloadType*)
- `dataSize`
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`
- `dataPayloadType`
- `dataSize` (*size_t*)

Save(*self*, *pFilename*, *format*=*SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT*)

Parameters

- `pFilename` (*char const **)
- `format` (*enum Spinnaker::ImageFileFormat*)
- `Save`(*self*
- `pFilename`
- `pOption`)
- `pFilename`
- `pOption` (*Spinnaker::BMPOption &*)
- `Save`(*self*
- `pFilename`
- `pOption`)
- `pFilename`
- `pOption`
- `Save`(*self*
- `pFilename`
- `pOption`)
- `pFilename`
- `pOption`
- `Save`(*self*
- `pFilename`
- `pOption`)

- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `option)`
- `pFilename`
- `option (Spinnaker::SIOption &)`

property thisown

The membership flag

class `PySpin.PySpin.IImageEventHandler(*args, **kwargs)`

Bases: [*EventHandler*](#)

Proxy of C++ `Spinnaker::IImageEventHandler` class.

property thisown

The membership flag

class `PySpin.PySpin.IImageList(*args, **kwargs)`

Bases: `object`

Proxy of C++ `Spinnaker::ImageList` class.

Add(*self*, *image*)

Parameters

image (*Spinnaker::ImagePtr*)

Append(*self*, *list*)

Parameters

list (Spinnaker::ImageList const &)

Clear(*self*)

GetByIndex(*self*, *index*) → *ImagePtr*

Parameters

index (unsigned int)

GetByPayloadType(*self*, *payloadType*) → *ImagePtr*

Parameters

payloadType (enum Spinnaker::ImagePayloadType const)

GetByPixelFormat(*self*, *pixelFormat*) → *ImagePtr*

Parameters

pixelFormat (enum Spinnaker::PixelFormatEnums)

GetByStreamIndex(*self*, *streamIndex*) → *ImagePtr*

Parameters

streamIndex (uint64_t const)

GetSize(*self*) → unsigned int

Release(*self*)

RemoveByIndex(*self*, *index*)

Parameters

index (unsigned int)

RemoveByPayloadType(*self*, *payloadType*)

Parameters

payloadType (enum Spinnaker::ImagePayloadType const)

RemoveByPixelFormat(*self*, *pixelFormat*)

Parameters

pixelFormat (enum Spinnaker::PixelFormatEnums)

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (uint64_t const)

Save(*self*, *filename*)

Parameters

filename (char const *)

property thisown

The membership flag

class PySpin.PySpin.IImageListEventHandler(*args, **kwargs)

Bases: *EventHandler*

Proxy of C++ Spinnaker::IImageListEventHandler class.

property thisown

The membership flag

class PySpin.PySpin.IImageProcessor(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IImageProcessor class.

ApplyGamma(self, srcImage, gamma, isGammaInverse=False) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **gamma** (*float*)
- **isGammaInverse** (*bool*)
- **ApplyGamma**(self
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **gamma**
- **isGammaInverse=False**)
- **srcImage**
- **destImage**
- **gamma**
- **isGammaInverse**

Convert(self, srcImage, destFormat) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **destFormat** (*enum Spinnaker::PixelFormatEnums*)
- **Convert**(self
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **destFormat**)
- **srcImage**
- **destImage**
- **destFormat**
- **Convert**(self
- **srcImageList** (*Spinnaker::ImageList const &*)
- **ImagePtr** (*destFormat*) ->)
- **srcImageList**
- **destFormat**
- **Convert**(self

- `srcImageList`
- `destImage`
- `destFormat`
- `srcImageList`
- `destImage`
- `destFormat`

GetColorProcessing(*self*) → Spinnaker::ColorProcessingAlgorithm

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

colorAlgorithm (enum Spinnaker::ColorProcessingAlgorithm)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

numThreads (unsigned int)

property **thisown**

The membership flag

class PySpin.PySpin.**IIInteger**(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IIInteger class.

GetInc(*self*) → int64_t

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

GetListOfValidValues(*self*, *bounded=True*) → *int64_autovector_t*

Parameters

bounded (bool)

GetMax(*self*) → int64_t

GetMin(*self*) → int64_t

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

GetUnit(*self*) → *gcstring*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → int64_t

Parameters

- **Verify** (bool)

- **IgnoreCache** (bool)

ImposeMax(*self*, *Value*)

Parameters

Value (int64_t)

ImposeMin(*self*, *Value*)

Parameters

Value (*int64_t*)

SetValue(*self*, *Value*, *Verify*=*True*)

Parameters

- **Value** (*int64_t*)

- **Verify** (*bool*)

property **thisown**

The membership flag

class PySpin.PySpin.**IInterface**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IInterface class.

GetCameras(*self*, *updateCameras*=*True*) → *CameraList*

Parameters

updateCameras (*bool*)

GetTLNodeMap(*self*) → *INodeMap*

IsCameraInUse(*self*) → *bool*

IsValid(*self*) → *bool*

RegisterEventHandler(*self*, *evtHandlerToRegister*)

Parameters

evtHandlerToRegister (*Spinnaker::EventHandler &*)

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime*=0, *requestAck*=*False*,
pResultSize=*None*, *results*=0)

Parameters

- **deviceKey** (*unsigned int*)

- **groupKey** (*unsigned int*)

- **groupMask** (*unsigned int*)

- **actionTime** (*unsigned long long*)

- **requestAck** (*bool*)

- **pResultSize** (*unsigned int **)

- **results** (*Spinnaker::ActionCommandResult []*)

property **TLInterface**

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

UpdateCameras(*self*) → *bool*

property thisown

The membership flag

class PySpin.PySpin.IInterfaceArrivalEventHandler(*args, **kwargs)Bases: [EventHandler](#)

Proxy of C++ Spinnaker::IInterfaceArrivalEventHandler class.

OnInterfaceArrival(self, pInterface)**Parameters****pInterface** (Spinnaker::InterfacePtr)**property thisown**

The membership flag

class PySpin.PySpin.IInterfaceEventHandler(*args, **kwargs)Bases: [IDeviceArrivalEventHandler](#), [IDeviceRemovalEventHandler](#)

Proxy of C++ Spinnaker::IInterfaceEventHandler class.

OnDeviceArrival(self, pCamera)**Parameters****pCamera** (Spinnaker::CameraPtr)**OnDeviceRemoval**(self, pCamera)**Parameters****pCamera** (Spinnaker::CameraPtr)**property thisown**

The membership flag

class PySpin.PySpin.IInterfaceList(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IInterfaceList class.

Add(self, iface)**Parameters****iface** (Spinnaker::InterfacePtr)**Append**(self, list)**Parameters****list** (Spinnaker::InterfaceList const *)**Clear**(self)**GetByIndex**(self, index) → [InterfacePtr](#)**Parameters****index** (unsigned int)**GetSize**(self) → unsigned int**Remove**(self, iface)**Parameters****iface** (Spinnaker::InterfacePtr)

property thisown

The membership flag

class PySpin.PySpin.IInterfaceRemovalEventHandler(*args, **kwargs)

Bases: [EventHandler](#)

Proxy of C++ Spinnaker::IInterfaceRemovalEventHandler class.

OnInterfaceRemoval(self, pInterface)

Parameters

pInterface (Spinnaker::InterfacePtr)

property thisown

The membership flag

class PySpin.PySpin.ILoggingEventHandler(*args, **kwargs)

Bases: [EventHandler](#)

Proxy of C++ Spinnaker::ILoggingEventHandler class.

OnLogEvent(self, eventPtr)

Parameters

eventPtr (Spinnaker::LoggingEventDataPtr)

property thisown

The membership flag

class PySpin.PySpin.INode(*args, **kwargs)

Bases: [ISelector](#), [IReference](#)

Proxy of C++ Spinnaker::GenApi::INode class.

DeregisterCallback(self, hCallback) → bool

Parameters

hCallback (Spinnaker::GenApi::CallbackHandleType)

GetAlias(self) → [INode](#)

GetCachingMode(self) → Spinnaker::GenApi::ECachingMode

GetCastAlias(self) → [INode](#)

GetChildren(self, Children, LinkType=ctReadingChildren)

Parameters

- **Children** (Spinnaker::GenApi::NodeList_t &)
- **LinkType** (enum Spinnaker::GenApi::ELinkType)

GetDescription(self) → *gcstring*

GetDeviceName(self) → *gcstring*

GetDisplayName(self) → *gcstring*

GetDocuURL(self) → *gcstring*

GetEventID(self) → *gcstring*

GetLockNodes(*self*, *LockNodes*)

Parameters

LockNodes (*Spinnaker::GenApi::NodeList_t* &)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

FullQualified (*bool*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

GetNodeMap(*self*) → *INodeMap*

GetParents(*self*, *Parents*)

Parameters

Parents (*Spinnaker::GenApi::NodeList_t* &)

GetPollingTime(*self*) → *int64_t*

GetPrincipalInterfaceType(*self*) → *Spinnaker::GenApi::EInterfaceType*

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → *bool*

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const* &)

- **ValueStr** (*Spinnaker::GenICam::gcstring* &)

- **AttributeStr** (*Spinnaker::GenICam::gcstring* &)

GetPropertyNames(*self*)

GetToolTip(*self*) → *gcstring*

GetVisibility(*self*) → *Spinnaker::GenApi::EVisibility*

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

ImposedAccessMode (*enum Spinnaker::GenApi::EAccessMode*)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*)

InvalidateNode(*self*)

IsAccessModeCacheable(*self*) → *Spinnaker::GenApi::EYesNo*

IsCachable(*self*) → *bool*

IsDeprecated(*self*) → *bool*

IsFeature(*self*) → *bool*

IsStreamable(*self*) → *bool*

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

pCallback (*Spinnaker::GenApi::CNodeCallback **)

property thisown

The membership flag

class PySpin.PySpin.**INodeMap**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::INodeMap class.

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)
- **Connect**(*self*
- **bool** (*pPort*) →)
- **pPort**

GetDeviceName(*self*) → *gcstring*

GetNode(*self*, *Name*) → *INode*

Parameters

Name (*Spinnaker::GenICam::gcstring const &*)

GetNodes(*self*)

GetNumNodes(*self*) → uint64_t

InvalidateNodes(*self*)

Poll(*self*, *ElapsedTime*)

Parameters

ElapsedTime (*int64_t*)

property thisown

The membership flag

class PySpin.PySpin.**INodeMapDyn**(*args, **kwargs)

Bases: *INodeMap*

Proxy of C++ Spinnaker::GenApi::INodeMapDyn class.

ClearAllNodes(*self*)

ExtractIndependentSubtree(*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *ExtractedSubtree*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **SubTreeRootNodeName** (*Spinnaker::GenICam::gcstring const &*)

- **ExtractedSubtree** (*Spinnaker::GenICam::gcstring &*)

GetSupportedSchemaVersions(*self*)

LoadXMLFromFile(*self, FileName*)

Parameters

- **FileName** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromFileInject(*self, TargetFileName, InjectFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromString(*self, XMLData*)

Parameters

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromStringInject(*self, TargetXMLData, InjectXMLData*)

Parameters

- **TargetXMLData** (*Spinnaker::GenICam::gcstring const &*)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*)

LoadXMLFromZIPData(*self, zipData, zipSize*)

Parameters

- **zipData** (*void const **)
- **zipSize** (*size_t*)

LoadXMLFromZIPFile(*self, ZipFileName*)

Parameters

- **ZipFileName** (*Spinnaker::GenICam::gcstring const &*)

MergeXMLFiles(*self, TargetFileName, InjectedFileName, OutputFileName*)

Parameters

- **TargetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **InjectedFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)

PreprocessXMLFromFile(*self, XMLFileName, StyleSheetFileName, OutputFileName, XMLValidation=xvDefault*)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

PreprocessXMLFromZIPFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*,
XMLValidation=xvDefault)

Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*)
- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*)
- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*)
- **XMLValidation** (*uint32_t const*)

property thisown

The membership flag

class PySpin.PySpin.**IPersistScript**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::IPersistScript class.

PersistFeature(*self*, *item*)

Parameters

item (*Spinnaker::GenApi::IValue &*)

SetInfo(*self*, *Info*)

Parameters

Info (*Spinnaker::GenICam::gcstring &*)

property thisown

The membership flag

class PySpin.PySpin.**IPointCloud**(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::IPointCloud class.

AddPoint(*self*, *point*)

Parameters

point (*Spinnaker::Stereo3DPoint const*)

GetNumPoints(*self*) → *size_t*

GetPoint(*self*, *index*) → *Stereo3DPoint*

Parameters

index (*unsigned int const*)

GetPointCloudData(*self*) → *Spinnaker::IPointCloud::PointCloudData **

LoadPointCloudFromPly(*self*, *filename*)

Parameters

filename (*std::string const &*)

PrintPoints(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)

SavePointCloudAsPly(*self*, *arg2*)

Parameters

arg2 (*std::string const &*)

property thisown

The membership flag

class PySpin.PySpin.IReference(**args*, ***kwargs*)

Bases: object

Proxy of C++ Spinnaker::GenApi::IReference class.

SetReference(*self*, *pBase*)

Parameters

pBase (*INode **)

property thisown

The membership flag

class PySpin.PySpin.IRegister(**args*, ***kwargs*)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IRegister class.

Get(*self*, *pBuffer*, *Length*, *Verify=False*, *IgnoreCache=False*)

Parameters

- **pBuffer** (*uint8_t **)
- **Length** (*int64_t*)
- **Verify** (*bool*)
- **IgnoreCache** (*bool*)
- **Get**(*self*
- **>** (*ignore_cache*) -> *std::vector< uint8_t*)
- **size_read** (*int64_t*)
- **Get**(*self*
- **size_read**
- **>**
- **size_read**
- **verify_range** (*bool*)
- **Get**(*self*
- **size_read**
- **verify_range**
- **>**
- **size_read**
- **verify_range**
- **ignore_cache** (*bool*)

Gets a NumPy array representing the contents of the register, as 8-bit unsigned ints.

6.3 Parameters:

pBuffer: The number of bytes to retrieve

Verify: Enables Range verification (default = false). The AccessMode is always checked

IgnoreCache: If true the value is read ignoring any caches (default = false)

GetAddress(*self*) → int64_t

GetLength(*self*) → int64_t

Set(*self*, *pBuffer*, *Verify*=True)

Parameters

- **pBuffer** (uint8_t const *)
- **Verify** (bool)

Set the register's contents with the contents (as 8-bit unsigned ints) of the given array.

6.4 Parameters:

pBuffer: The NumPy array containing the data to set

Verify: Enables AccessMode and Range verification (default = true)

property thisown

The membership flag

class PySpin.PySpin.ISelector(*args, **kwargs)

Bases: [IBase](#)

Proxy of C++ Spinnaker::GenApi::ISelector class.

GetSelectedFeatures(*self*, *arg2*)

Parameters

arg2 (FeatureList_t &)

GetSelectingFeatures(*self*, *arg2*)

Parameters

arg2 (FeatureList_t &)

IsSelector(*self*) → bool

property thisown

The membership flag

class PySpin.PySpin.ISelectorDigit(*args, **kwargs)

Bases: object

Proxy of C++ Spinnaker::GenApi::ISelectorDigit class.

GetSelectorList(*self*, *Incremental=False*)

Parameters

Incremental (*bool*)

Restore(*self*)

SetFirst(*self*) → *bool*

SetNext(*self*, *Tick=True*) → *bool*

Parameters

Tick (*bool*)

ToString(*self*) → *gcstring*

property thisown

The membership flag

class PySpin.PySpin.**IString**(*args, **kwargs)

Bases: *IValue*

Proxy of C++ Spinnaker::GenApi::IString class.

GetMaxLength(*self*) → *int64_t*

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**ISystem**(*args, **kwargs)

Bases: *object*

Proxy of C++ Spinnaker::ISystem class.

GetCameras(*self*, *updateInterfaces=True*, *updateCameras=True*) → *CameraList*

Parameters

- **updateInterfaces** (*bool*)
- **updateCameras** (*bool*)

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

updateInterface (*bool*)

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → bool

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (Spinnaker::EventHandler &)
- **updateInterface** (bool)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (Spinnaker::LoggingEventHandler &)

ReleaseInstance(*self*)

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*,
pResultSize=None, *results=0*)

Parameters

- **deviceKey** (unsigned int)
- **groupKey** (unsigned int)
- **groupMask** (unsigned int)
- **actionTime** (unsigned long long)
- **requestAck** (bool)
- **pResultSize** (unsigned int *)
- **results** (Spinnaker::ActionCommandResult [])

SetLoggingEventPriorityLevel(*self*, *level*)

Parameters

level (enum Spinnaker::SpinnakerLogLevel)

property **TLSystem**

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (Spinnaker::EventHandler &)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (Spinnaker::LoggingEventHandler &)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

updateInterfaces (bool)

UpdateInterfaceList(*self*)

property thisown

The membership flag

class PySpin.PySpin.**ISystemEventHandler**(*args, **kwargs)

Bases: [*IInterfaceArrivalEventHandler*](#), [*IInterfaceRemovalEventHandler*](#)

Proxy of C++ Spinnaker::ISystemEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.PySpin.**IValue**(*args, **kwargs)

Bases: [*INode*](#)

Proxy of C++ Spinnaker::GenApi::IValue class.

FromString(*self*, *ValueStr*, *Verify=True*)

Parameters

- **ValueStr** (*Spinnaker::GenICam::gcstring const &*)
- **Verify** (*bool*)

GetNode(*self*) → [*INode*](#)

IsValueCacheValid(*self*) → bool

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → [*gcstring*](#)

Parameters

- **Verify** (*bool*)
- **IgnoreCache** (*bool*)

property thisown

The membership flag

class PySpin.PySpin.**Image**(*args, **kwargs)

Bases: [*IImage*](#)

The image object class.

C++ includes: Image.h

CheckCRC(*self*) → bool

bool *Spinnaker::Image::CheckCRC()* const

Checks if the computed checksum matches with chunk data's ImageCRC

Returns true if computed checksum matches with the chunk data's CRC and false otherwise.

static **Create()** \rightarrow *ImagePtr*

static **Create**(*image*) \rightarrow *ImagePtr*

Parameters

- **image** (*Spinnaker::ImagePtr const*)
- **Create**(**width**
- **height** (*size_t*)
- **offsetX** (*size_t*)
- **offsetY** (*size_t*)
- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*)
- **ImagePtr** (*copied from another*)
- **width** (*or using*)
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData** (*void **)
- **Create**(**width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType** (*enum Spinnaker::TLPayloadType*)
- **ImagePtr**
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**
- **dataPayloadType**
- **dataSize** (*size_t*)
- **object** (*Creates a new Image*)
- **constructor** (*either using a default*)
- **ImagePtr**
- **width**

- **height**

:param : :param offset_x: :param offset_y: :param pixel format: :param and a NumPy array containing 8-bit unsigned ints representing the image data: :param (replaces the void* pData argument).:

DeepCopy(*self*, *pSrcImage*)

Parameters

- **pSrcImage** (*The Image to copy the data from.*)
- **void**
- **pSrcImage** (*Spinnaker::Image::DeepCopy(const ImagePtr)*)
- **operation** (*Performs a deep copy of the Image. After this*)
- **image** (*the*)
- **not** (*contents and member variables will be the same. The Images will*)
- **released.** (*share a buffer. The Image's current buffer will not be*)
- **Parameters**
- -----
- **pSrcImage**

GetBitsPerPixel(*self*) → size_t

size_t Spinnaker::Image::GetBitsPerPixel() const

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.

The number of bits used per pixel.

GetBufferSize(*self*) → size_t

size_t Spinnaker::Image::GetBufferSize() const

Gets the size of the buffer associated with the image in bytes.

The size of the buffer, in bytes.

GetChunkData(*self*) → *ChunkData*

const ChunkData& Spinnaker::Image::GetChunkData() const

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.

ChunkData interface that provides access to image chunks.

GetChunkLayoutId(*self*) → uint64_t

uint64_t Spinnaker::Image::GetChunkLayoutId() const

Returns the id of the chunk data layout.

uint64_t value representing the id of the chunk data layout.

GetColorProcessing(*self*) → Spinnaker::ColorProcessingAlgorithm

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

GetDataAbsoluteMax(*self*) → float

GetDataAbsoluteMin(*self*) → float

GetFrameID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight(*self*) → size_t

size_t Spinnaker::Image::GetHeight() const

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.

The height in pixels.

GetID(*self*) → uint64_t

uint64_t Spinnaker::Image::GetID() const

Gets a unique ID for this image. Each image in a stream will have a unique ID to help identify it.

The 64 bit unique id for this image.

GetImagePayloadType(*self*) → Spinnaker::ImagePayloadType

GetImageSize(*self*) → size_t

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus(*self*) → Spinnaker::ImageStatus

ImageStatus Spinnaker::Image::GetImageStatus() const

Returns data integrity status of the image returned from GetNextImage()

Returns whether image has any data integrity issues.

static GetImageStatusDescription(*status*) → char const *

Parameters

status (enum Spinnaker::ImageStatus)

GetNumChannels(*self*) → size_t

GetPayloadType(*self*) → size_t

size_t Spinnaker::Image::GetPayloadType() const

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.

Device types specific payload type.

GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums

Spinnaker::PixelFormatEnums Spinnaker::Image::GetPixelFormat() const

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: Convert()

enum value representing the PixelFormat.

GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType

GetPixelFormatName(*self*) → *gcstring*

GenICam::gcstring Spinnaker::Image::GetPixelFormatName() const

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.

string value representing the PixelFormat.

GetPrivateData(*self*) → void *

void* Spinnaker::Image::GetPrivateData() const

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.

A pointer to the user passed data pointer.

GetStreamIndex(*self*) → uint64_t

GetStride(*self*) → size_t

size_t Spinnaker::Image::GetStride() const

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.

The stride in bytes.

GetTLPayloadType(*self*) → Spinnaker::TLPayloadType

PayloadTypeInfoIDs Spinnaker::Image::GetTLPayloadType() const

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.

Transport Layer specific payload type.

GetTLPixelFormat(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTLPixelFormat() const

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: GetTLPixelFormatNamespace()

Transport Layer specific pixel format.

GetTLPixelFormatNamespace(*self*) → Spinnaker::TLPixelFormatNamespace

PixelFormatNamespaceID Spinnaker::Image::GetTLPixelFormatNamespace() const

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: GetTLPixelFormat()

enum value representing the PixelFormatNamespace.

GetTimeStamp(*self*) → uint64_t

uint64_t Spinnaker::Image::GetTimeStamp() const

Gets the time stamp for the image in nanoseconds.

The time stamp of the image.

GetValidPayloadSize(*self*) → size_t

size_t Spinnaker::Image::GetValidPayloadSize() const

Returns the size of valid data in the image payload. This is the actual amount of data read from the device. A user created image has a payload size of zero. GetBufferSize() returns the total size of bytes allocated for the image.

See: GetBufferSize()

size_t value representing valid payload.

GetWidth(*self*) → size_t

size_t Spinnaker::Image::GetWidth() const

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.

The width in pixels.

GetXOffset(*self*) → size_t

size_t Spinnaker::Image::GetXOffset() const

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.

The x offset in pixels.

GetXPadding(*self*) → size_t

size_t Spinnaker::Image::GetXPadding() const

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.

The x padding in bytes.

GetYOffset(*self*) → size_t

size_t Spinnaker::Image::GetYOffset() const

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.

The y offset in pixels.

GetYPadding(*self*) → size_t

size_t Spinnaker::Image::GetYPadding() const

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.

The y padding in bytes.

HasCRC(*self*) → bool

bool Spinnaker::Image::HasCRC() const

Checks if the image contains ImageCRC checksum from chunk data

Returns true if image contains ImageCRC checksum from chunk data and false otherwise.

HasChunkData(*self*) → bool

IsCompressed(*self*) → bool

IsInUse(*self*) → bool

bool Spinnaker::Image::IsInUse()

Returns true if the image is still in use by the stream

Returns true if the image is in use and false otherwise.

IsIncomplete(*self*) → bool

bool Spinnaker::Image::IsIncomplete() const

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 true if image is incomplete, false otherwise.

static Load(*pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT) → *ImagePtr*

Parameters

- **pFilename** (*char const **)
- **format** (*enum Spinnaker::ImageFileFormat*)

Release(*self*)

void Spinnaker::Image::Release()

ResetImage(*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

Parameters

- **width** (*The width of image in pixels to set.*)
- **height** (*The height of image in pixels to set.*)
- **offsetX** (*The x offset in pixels to set.*)
- **offsetY** (*The y offset in pixels to set.*)
- **pixelFormat** (*Pixel format to set.*)
- **ResetImage**(*self*
- **width**
- **height**

- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`)
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData` (*Pointer to the image buffer.*)
- `ResetImage(self`
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`
- `dataPayloadType` (*enum Spinnaker::TLPayloadType*)
- `dataSize`)
- `width`
- `height`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `pData`
- `dataPayloadType`
- `dataSize` (*size_t*)
- `void`
- `width`
- `height`
- `size_t`
- `offsetX`
- `offsetY`
- `pixelFormat`
- `void`
- `*pData)`

- **object.** (*Sets new dimensions of the image*)
- **Parameters**
- -----
- **width**
- **height**
- **offsetX**
- **offsetY**
- **pixelFormat**
- **pData**

Save(*self*, *pFilename*, *format*=SPINNAKER_IMAGE_FILE_FORMAT_FROM_FILE_EXT)

Parameters

- **pFilename** (*Filename to save image with.*)
- **format** (*enum Spinnaker::ImageFileFormat*)
- **Save**(*self*
- **pFilename**
- **pOption**)
- **pFilename**
- **pOption** (*Options to use while saving image.*)
- **Save**(*self*
- **pFilename**
- **pOption**)
- **pFilename**
- **pOption**
- **Save**(*self*
- **pFilename**
- **pOption**)
- **pFilename**
- **pOption**
- **Save**(*self*
- **pFilename**
- **pOption**)
- **pFilename**
- **pOption**
- **Save**(*self*
- **pFilename**
- **pOption**)

- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `Save(self`
- `pFilename`
- `pOption)`
- `pFilename`
- `pOption`
- `void`
- `*pFilename` (*Spinnaker::Image::Save(const char) –*
- `&pOption)` ([BMPOption](#))
- `specified.` (*Saves the image to the specified file name with the*
options)
- `Parameters`
- `-----`
- `pFilename`
- `pOption`

property thisown

The membership flag

class `PySpin.PySpin.ImageEventHandler`

Bases: [IImageEventHandler](#)

Proxy of C++ `Spinnaker::ImageEventHandler` class.

OnImageEvent(*self, image*)

Parameters

image (*Spinnaker::ImagePtr*)

property thisown

The membership flag


```

class PySpin.PySpin.ImageList(*args)
    Bases: IImageList
    Proxy of C++ Spinnaker::ImageList class.

    Add(self, image)

        Parameters
        image (Spinnaker::ImagePtr)

    Append(self, list)

        Parameters
        list (Spinnaker::ImageList const &)

    Clear(self)

    GetByIndex(self, index) → ImagePtr

        Parameters
        index (unsigned int)

    GetByPayloadType(self, payloadType) → ImagePtr

        Parameters
        payloadType (enum Spinnaker::ImagePayloadType const)

    GetByPixelFormat(self, pixelFormat) → ImagePtr

        Parameters
        pixelFormat (enum Spinnaker::PixelFormatEnums)

    GetByStreamIndex(self, streamIndex) → ImagePtr

        Parameters
        streamIndex (uint64_t const)

    GetSize(self) → unsigned int

    static Load(filename) → ImageList

        Parameters
        filename (char const *)

    Release(self)

    RemoveByIndex(self, index)

        Parameters
        index (unsigned int)

    RemoveByPayloadType(self, payloadType)

        Parameters
        payloadType (enum Spinnaker::ImagePayloadType const)

    RemoveByPixelFormat(self, pixelFormat)

        Parameters
        pixelFormat (enum Spinnaker::PixelFormatEnums)

```

RemoveByStreamIndex(*self*, *streamIndex*)

Parameters

streamIndex (*uint64_t* *const*)

Save(*self*, *filename*)

Parameters

filename (*char const **)

property thisown

The membership flag

class PySpin.PySpin.**ImageListEventHandler**

Bases: *IImageListEventHandler*

Proxy of C++ Spinnaker::ImageListEventHandler class.

OnImageListEvent(*self*, *imageList*)

Parameters

imageList (*Spinnaker::ImageList*)

property thisown

The membership flag

class PySpin.PySpin.**ImagePixel**

Bases: *object*

Proxy of C++ Spinnaker::ImagePixel class.

property thisown

The membership flag

property u

property v

class PySpin.PySpin.**ImageProcessor**(**args*)

Bases: *IImageProcessor*

Proxy of C++ Spinnaker::ImageProcessor class.

ApplyGamma(*self*, *srcImage*, *gamma*, *applyGammaInverse=False*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **gamma** (*float*)
- **applyGammaInverse** (*bool*)
- **ApplyGamma**(*self*
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **gamma**
- **applyGammaInverse=False**)
- **srcImage**

- `destImage`
- `gamma`
- `applyGammaInverse`

Convert(*self*, *srcImage*, *destFormat*) → *ImagePtr*

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `destFormat` (*enum Spinnaker::PixelFormatEnums*)
- **Convert**(*self*
- `srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)
- `destFormat`)
- `srcImage`
- `destImage`
- `destFormat`
- **Convert**(*self*
- `srcImageList` (*Spinnaker::ImageList const &*)
- `ImagePtr` (*destFormat*) ->)
- `srcImageList`
- `destFormat`
- **Convert**(*self*
- `srcImageList`
- `destImage`
- `destFormat`)
- `srcImageList`
- `destImage`
- `destFormat`

GetColorProcessing(*self*) → *Spinnaker::ColorProcessingAlgorithm*

GetNumDecompressionThreads(*self*) → unsigned int

SetColorProcessing(*self*, *colorAlgorithm*)

Parameters

- `colorAlgorithm` (*enum Spinnaker::ColorProcessingAlgorithm*)

SetNumDecompressionThreads(*self*, *numThreads*)

Parameters

- `numThreads` (*unsigned int*)

property thisown

The membership flag

class PySpin.PySpin.**ImagePtr**(*args)

Bases: `_SWIG_ImgPtr`

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.

C++ includes: ImagePtr.h

property thisown

The membership flag

class PySpin.PySpin.**ImageUtility**

Bases: `object`

Proxy of C++ Spinnaker::ImageUtility class.

static CreateNormalized(srcImage, destPixelFormat, src-
DataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)
→ *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **destPixelFormat** (*enum Spinnaker::PixelFormatEnums const*)
- **srcDataRange** (*enum Spinnaker::SourceDataRange*)
- **CreateNormalized**(srcImage
- **min** (*double const*)
- **max** (*double const*)
- **ImagePtr** (*srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE*)
→)
- **srcImage**
- **min**
- **max**
- **srcDataRange**
- **CreateNormalized**(srcImage
- **min**
- **max**
- **destPixelFormat**
- **ImagePtr**
- **srcImage**
- **min**
- **max**
- **destPixelFormat**
- **srcDataRange**
- **CreateNormalized**(srcImage
- **destImage** (*Spinnaker::ImagePtr &*)

- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `srcDataRange`
- `CreateNormalized(srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange=SPINNAKER_SOURCE_DATA_RANGE_IMAGE_DATA_RANGE)`
- `srcImage`
- `destImage`
- `min`
- `max`
- `srcDataRange`

static `CreateScaled(srcImage, scalingAlg, scalingFactor) → ImagePtr`

Parameters

- `srcImage` (*Spinnaker::ImagePtr const &*)
- `scalingAlg` (*enum Spinnaker::ImageScalingAlgorithm*)
- `scalingFactor` (*double*)
- `CreateScaled(srcImage`
- `destImage` (*Spinnaker::ImagePtr &*)
- `scalingAlg`
- `scalingFactor`)
- `srcImage`
- `destImage`
- `scalingAlg`
- `scalingFactor`

property `thisown`

The membership flag

class `PySpin.PySpin.ImageUtilityCCM`

Bases: `object`

Proxy of C++ `Spinnaker::ImageUtilityCCM` class.

static `ApplicationToString(application) → std::string`

Parameters

- `application` (*Spinnaker::CCMApplication const &*)

static ColorSpaceToString(*colorSpace*) → std::string

Parameters

colorSpace (*Spinnaker::CCMColorSpace const &*)

static ColorTemperatureToString(*colorTemperature*) → std::string

Parameters

colorTemperature (*Spinnaker::CCMColorTemperature const &*)

static CreateColorCorrected(*srcImage, settings*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **settings** (*Spinnaker::CCMSettings const &*)
- **CreateColorCorrected**(**srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)
- **settings**)
- **srcImage**
- **destImage**
- **settings**

static EncryptColorCorrectionMatrix(*ccmMatrixEntries*) → std::string

Parameters

ccmMatrixEntries (*std::string*)

static SensorToString(*sensor*) → std::string

Parameters

sensor (*Spinnaker::CCMSensor const &*)

static TypeToString(*type*) → std::string

Parameters

type (*Spinnaker::CCMType const &*)

property thisown

The membership flag

class PySpin.PySpin.**ImageUtilityHeatmap**

Bases: object

Proxy of C++ Spinnaker::ImageUtilityHeatmap class.

static CreateHeatmap(*srcImage*) → *ImagePtr*

Parameters

- **srcImage** (*Spinnaker::ImagePtr const &*)
- **CreateHeatmap**(**srcImage**
- **min** (*float const*)
- **max** (*float const*)
- **lowColor** (*enum Spinnaker::HeatmapColor const*)

- **highColor** (*enum Spinnaker::HeatmapColor const*)
- **doCheckInvalidVal** (*bool const*)
- **ImagePtr** (*invalidVal*) ->)
- **srcImage**
- **min**
- **max**
- **lowColor**
- **highColor**
- **doCheckInvalidVal**
- **invalidVal** (*unsigned int const*)
- **CreateHeatmap**(**srcImage**
- **destImage**)
- **srcImage**
- **destImage** (*Spinnaker::ImagePtr &*)

static GetHeatmapColorGradient(*currentLowColor, currentHighColor*)

Parameters

- **currentLowColor** (*Spinnaker::HeatmapColor &*)
- **currentHighColor** (*Spinnaker::HeatmapColor &*)

static GetHeatmapRange(*currentLowValue, currentHighValue*)

Parameters

- **currentLowValue** (*unsigned int &*)
- **currentHighValue** (*unsigned int &*)

static SetHeatmapColorGradient(*newLowColor, newHighColor*)

Parameters

- **newLowColor** (*enum Spinnaker::HeatmapColor const*)
- **newHighColor** (*enum Spinnaker::HeatmapColor const*)

static SetHeatmapRange(*newLowValue, newHighValue*)

Parameters

- **newLowValue** (*unsigned int const*)
- **newHighValue** (*unsigned int const*)

property thisown

The membership flag

class PySpin.PySpin.ImageUtilityPolarization

Bases: object

Proxy of C++ Spinnaker::ImageUtilityPolarization class.

```
static CreateAolp(srcImage, colorProcessin-  
                  gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                  → ImagePtr
```

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateAolp**(srcImage
- **destAolpImg** (Spinnaker::ImagePtr &)
- **colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)**
- **srcImage**
- **destAolpImg**
- **colorProcessingAlg**

```
static CreateDolp(srcImage, colorProcessin-  
                  gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                  → ImagePtr
```

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateDolp**(srcImage
- **destDolpImage** (Spinnaker::ImagePtr &)
- **colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)**
- **srcImage**
- **destDolpImage**
- **colorProcessingAlg**

```
static CreateGlareReduced(srcImage) → ImagePtr
```

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **CreateGlareReduced**(srcImage
- **destGlareReducedImage**)
- **srcImage**
- **destGlareReducedImage** (Spinnaker::ImagePtr &)

```
static CreateStokesS0(srcImage, colorProcessin-  
                      gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)  
                      → ImagePtr
```

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)

- **CreateStokesS0**(srcImage
- destStokesS0Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS0Image
- colorProcessingAlg

static CreateStokesS1(srcImage, colorProcessin-
gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateStokesS1**(srcImage
- destStokesS1Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS1Image
- colorProcessingAlg

static CreateStokesS2(srcImage, colorProcessin-
gAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
→ ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **colorProcessingAlg** (enum Spinnaker::ColorProcessingAlgorithm const)
- **CreateStokesS2**(srcImage
- destStokesS2Image (Spinnaker::ImagePtr &)
- colorProcessingAlg=SPINNAKER_COLOR_PROCESSING_ALGORITHM_NEAREST_NEIGHBOR)
- srcImage
- destStokesS2Image
- colorProcessingAlg

static ExtractPolarQuadrant(srcImage, desiredQuadrant) → ImagePtr

Parameters

- **srcImage** (Spinnaker::ImagePtr const &)
- **desiredQuadrant** (enum Spinnaker::PolarizationQuadrant const)
- **ExtractPolarQuadrant**(srcImage
- destQuadImage (Spinnaker::ImagePtr &)
- desiredQuadrant)

- `srcImage`
- `destQuadImage`
- `desiredQuadrant`

property thisown

The membership flag

class `PySpin.PySpin.ImageUtilityStereo`

Bases: `object`

Proxy of C++ `Spinnaker::ImageUtilityStereo` class.

static `Compute3DPointFromPixel`(*disparity, stereoCameraParameters, stereo3DPoint*) → `bool`

Parameters

- `disparity` (`uint16_t const`)
- `stereoCameraParameters` (`Spinnaker::StereoCameraParameters const &`)
- `stereo3DPoint` (`Spinnaker::Stereo3DPoint &`)

static `ComputeDistanceBetweenPoints`(*disparityImage, stereoParam, imagePixel1, imagePixel2*) → `PyObject *`

Parameters

- `disparityImage` (`Spinnaker::ImagePtr const &`)
- `stereoParam` (`Spinnaker::StereoCameraParameters const &`)
- `imagePixel1` (`Spinnaker::ImagePixel const &`)
- `imagePixel2` (`Spinnaker::ImagePixel const &`)

static `ComputeDistanceToPoint`(*disparityImage, stereoParam, imagePixel*) → `PyObject *`

Parameters

- `disparityImage` (`Spinnaker::ImagePtr const &`)
- `stereoParam` (`Spinnaker::StereoCameraParameters const &`)
- `imagePixel` (`Spinnaker::ImagePixel const &`)

static `ComputePointCloud`(*disparityImage, rectifiedImage, pointCloudParameters, stereoCameraParameters*) → `PointCloud`

Parameters

- `disparityImage` (`Spinnaker::ImagePtr const &`)
- `rectifiedImage` (`Spinnaker::ImagePtr const &`)
- `pointCloudParameters` (`Spinnaker::PointCloudParameters const &`)
- `stereoCameraParameters` (`Spinnaker::StereoCameraParameters const &`)
- `ComputePointCloud`(`disparityImage`
- `rectifiedImage`
- `pointCloudParameters`
- `stereoCameraParameters`

- `pointCloud`)
- `disparityImage`
- `rectifiedImage`
- `pointCloudParameters`
- `stereoCameraParameters`
- `pointCloud` (*Spinnaker::PointCloud &*)

static `CreateDepthImage`(*disparityImage, stereoCameraParameters, invalidDepthVal, depth_range_list*)
 → *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `stereoCameraParameters` (*Spinnaker::StereoCameraParameters const &*)
- `invalidDepthVal` (*uint16_t const*)
- `depth_range_list` (*PyObject **)
- `CreateDepthImage`(*disparityImage*
- `stereoCameraParameters`
- `invalidDepthVal`
- `depthImage` (*Spinnaker::ImagePtr &*)
- `depth_range_list`)
- `disparityImage`
- `stereoCameraParameters`
- `invalidDepthVal`
- `depthImage`
- `depth_range_list`

static `FilterSpeckles`(*disparityImage, maxSpeckleSize, speckleThreshold, disparityScaleFactor,*
 invalidDataValue) → *ImagePtr*

Parameters

- `disparityImage` (*Spinnaker::ImagePtr const &*)
- `maxSpeckleSize` (*int const*)
- `speckleThreshold` (*int const*)
- `disparityScaleFactor` (*float const*)
- `invalidDataValue` (*float const*)

static `FilterSpecklesFromImage`(*disparityImage, maxSpeckleSize, speckleThreshold,*
 disparityScaleFactor, invalidDataValue)

Parameters

- `disparityImage` (*Spinnaker::ImagePtr &*)
- `maxSpeckleSize` (*int const*)

- **speckleThreshold**(*int const*)
- **disparityScaleFactor**(*float const*)
- **invalidDataValue**(*float const*)

static IsStereoCamera(*pCamera*) → bool

Parameters

pCamera (*Spinnaker::CameraPtr*)

property maxDepthThresholdInMeter

property maxDepthThresholdInMm

property thisown

The membership flag

class PySpin.PySpin.**InferenceBoundingBox**

Bases: object

Proxy of C++ Spinnaker::InferenceBoundingBox class.

property boxType

property circle

property classId

property confidence

property rect

property rotatedRect

property thisown

The membership flag

class PySpin.PySpin.**InferenceBoundingBoxResult**(*args)

Bases: object

Proxy of C++ Spinnaker::InferenceBoundingBoxResult class.

GetBoxAt(*self, index*) → *InferenceBoundingBox*

Parameters

index (*uint16_t const*)

GetBoxCount(*self*) → int16_t

GetBoxSize(*self*) → int8_t

GetVersion(*self*) → int8_t

property thisown

The membership flag

class PySpin.PySpin.**InferenceBoxCircle**

Bases: object

Proxy of C++ Spinnaker::InferenceBoxCircle class.

property centerXCoord

property centerYCoord

property radius

property thisown

The membership flag

class PySpin.PySpin.InferenceBoxRect

Bases: object

Proxy of C++ Spinnaker::InferenceBoxRect class.

property bottomRightXCoord

property bottomRightYCoord

property thisown

The membership flag

property topLeftXCoord

property topLeftYCoord

class PySpin.PySpin.InferenceBoxRotatedRect

Bases: object

Proxy of C++ Spinnaker::InferenceBoxRotatedRect class.

property bottomRightXCoord

property bottomRightYCoord

property rotationAngle

property thisown

The membership flag

property topLeftXCoord

property topLeftYCoord

class PySpin.PySpin.IntRegNode(*args, **kwargs)

Bases: *IntegerNode*, *RegisterNode*

Interface for string properties.

C++ includes: IntRegNode.h

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase**) (*virtual void Spinnaker::GenApi::IntRegNode::SetReference(INode)*
—
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.PySpin.IntegerNode(*args, **kwargs)

Bases: *IInteger*, *ValueNode*

Interface for string properties.

C++ includes: IntegerNode.h

GetFloatAlias(self) → *IFloat*

virtual IFloat* Spinnaker::GenApi::IntegerNode::GetFloatAlias()

gets the interface of an alias node.

GetInc(self) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetInc()

Get increment

GetIncMode(self) → Spinnaker::GenApi::EIncMode

virtual EIncMode Spinnaker::GenApi::IntegerNode::GetIncMode()

Get increment mode

GetListOfValidValues(self, bounded=True) → *int64_autovector_t*

Parameters

- **bounded** (*bool*)
- **virtual**
- **int64_autovector_t**
- **Spinnaker::GenApi::IntegerNode::GetListOfValidValues**(bool
- **bounded=true)**
- **value** (*Get list of valid*)

GetMax(self) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetMax()

Get maximum value allowed

GetMin(self) → int64_t

virtual int64_t Spinnaker::GenApi::IntegerNode::GetMin()

Get minimum value allowed

GetRepresentation(self) → Spinnaker::GenApi::ERepresentation

virtual ERepresentation Spinnaker::GenApi::IntegerNode::GetRepresentation()

Get recommended representation

GetUnit(self) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::IntegerNode::GetUnit()

Get the physical unit name

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → int64_t

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::IntegerNode::GetValue**(bool (virtual int64_t))
- **Verify=false**
- **IgnoreCache=false**) (bool)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

ImposeMax(*self*, *Value*)

Parameters

- **Value** (int64_t)
- **Value**) (virtual void Spinnaker::GenApi::IntegerNode::ImposeMax(int64_t))
- **value** (*Restrict maximum*)

ImposeMin(*self*, *Value*)

Parameters

- **Value** (int64_t)
- **Value**) (virtual void Spinnaker::GenApi::IntegerNode::ImposeMin(int64_t))
- **value** (*Restrict minimum*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::IntegerNode::SetReference**(INode (virtual void))
- ***pBase**)
- **Integer** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (virtual void Spinnaker::GenApi::IntegerNode::SetValue(int64_t))
- **Verify** (bool)

- Value

:param : :param bool Verify=true): :param Set node value: :param Parameters: :param ———: :param Value: :type Value: The value to set :param Verify: :type Verify: Enables AccessMode and Range verification (default = true)

property thisown

The membership flag

class PySpin.PySpin.InterfaceArrivalEventHandler

Bases: [*IInterfaceArrivalEventHandler*](#)

Proxy of C++ Spinnaker::InterfaceArrivalEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.PySpin.InterfaceEventHandler

Bases: [*IInterfaceEventHandler*](#)

Proxy of C++ Spinnaker::InterfaceEventHandler class.

OnDeviceArrival(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

OnDeviceRemoval(*self*, *pCamera*)

Parameters

pCamera (*Spinnaker::CameraPtr*)

property thisown

The membership flag

class PySpin.PySpin.InterfaceList(*args)

Bases: [*IInterfaceList*](#)

A list of the available interfaces on the system.

C++ includes: InterfaceList.h

Add(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

Append(*self*, *list*)

Parameters

list (*Spinnaker::InterfaceList const **)

Clear(*self*)

void Spinnaker::InterfaceList::Clear()

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.

GetByIndex(*self*, *index*) → *InterfacePtr*

Parameters

- **index** (*The index at which to retrieve the Interface object*)
- **const** (*InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index)*)
- **"index"**. (*Returns a pointer to an Interface object at the*)
- **Parameters**
- -----
- **index**
- **object.** (*A pointer to an Interface*)

GetByInterfaceID(*self*, *interfaceID*) → *InterfacePtr*

Parameters

interfaceID (*std::string*)

GetSize(*self*) → unsigned int

int Spinnaker::InterfaceList::GetSize() const

Returns the size of the interface list. The size is the number of Interface objects stored in the list.

An integer that represents the list size.

Remove(*self*, *iface*)

Parameters

iface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.PySpin.**InterfacePtr**(*args)

Bases: `_SWIG_IFacePtr`

A reference tracked pointer to the interface object.

C++ includes: `InterfacePtr.h`

property thisown

The membership flag

class PySpin.PySpin.**InterfaceRemovalEventHandler**

Bases: `IInterfaceRemovalEventHandler`

Proxy of C++ Spinnaker::InterfaceRemovalEventHandler class.

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

PySpin.PySpin.IsAvailable(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsAvailable(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **T** (*Spinnaker::GenApi::IsAvailable(const Spinnaker::GenApi::CPointer<*)

- **B**
- **&ptr** (>)
- **Available** (*Checks if a node is*)

PySpin.PySpin.**IsCacheable**(*CachingMode*) → bool

Parameters

- **CachingMode** (*enum Spinnaker::GenApi::ECachingMode*)
- **bool**
- **CachingMode** (*Spinnaker::GenApi::IsCacheable(ECachingMode)*)
- **Cacheability** (*Tests*)

PySpin.PySpin.**IsImplemented**(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsImplemented(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**

- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **T**(*Spinnaker::GenApi::IsImplemented*(const *Spinnaker::GenApi::CPointer*<)

:param : :param B > &ptr): :param Checks if a node is Implemented:

PySpin.PySpin.**IsReadable**(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsReadable(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**

- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **ptr**
- **bool**
- **T** (*Spinnaker::GenApi::IsReadable(const Spinnaker::GenApi::CPointer<*)
- **B**
- **&ptr** (*>*)
- **readable** (*Checks if a node is*

PySpin.PySpin.**IsVisible**(*Visibility, MaxVisiblity*) → bool

Parameters

- **Visibility** (*Spinnaker::GenApi::IsVisible(EVisibility)*
- **MaxVisiblity** (*enum Spinnaker::GenApi::EVisibility*)
- **bool**
- **Visibility**
- **EVisibility**
- **MaxVisiblity**
- **CAVE** (*Tests Visibility*)

PySpin.PySpin.**IsWritable**(*AccessMode*) → bool

Parameters

- **AccessMode** (*enum Spinnaker::GenApi::EAccessMode*)
- **bool** (*IsWritable(ptr) ->*)
- **p** (*Spinnaker::GenApi::IBase const **)
- **bool**
- **r** (*Spinnaker::GenApi::IBase const &*)
- **bool**
- **ptr** (*Spinnaker::GenApi::CPointer< Spinnaker::GenApi::IFloat, Spinnaker::GenApi::IBase > const &*)
- **bool**
- **ptr**
- **bool**
- **ptr**

- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- ptr
- bool
- T(*Spinnaker::GenApi::IsWritable(const Spinnaker::GenApi::CPointer<*
- B
- &ptr) (>)
- Writable(*Checks if a node is*)

```
class PySpin.PySpin.JPEGOption
```

Bases: object

Options for saving JPEG image.

C++ includes: SpinnakerDefs.h

property progressive

property quality

property reserved

property thisown

The membership flag

class PySpin.PySpin.JPG20Option

Bases: object

Options for saving JPEG2000 image.

C++ includes: SpinnakerDefs.h

property quality**property** reserved**property** thisown

The membership flag

class PySpin.PySpin.LibraryVersion

Bases: object

Proxy of C++ Spinnaker::LibraryVersion class.

property build**property** major**property** minor**property** thisown

The membership flag

property type**class** PySpin.PySpin.LoggingEventData(*args, **kwargs)

Bases: object

The LoggingEventData object.

C++ includes: LoggingEventData.h

GetCategoryName(self) → char const *

const char* Spinnaker::LoggingEventData::GetCategoryName()

Gets the logging event category name.

The category name

GetLogMessage(self) → char const *

const char* Spinnaker::LoggingEventData::GetLogMessage()

Gets the logging event message.

The log message

GetNDC(self) → char const *

const char* Spinnaker::LoggingEventData::GetNDC()

Gets the logging event's Nested Diagnostic Context (NDC).

The log event's NDC

GetPriority(self) → int const

const int Spinnaker::LoggingEventData::GetPriority()

Gets the logging event priority.

The log priority

GetPriorityName(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetPriorityName()

Gets the logging event priority name.

The priority name of the log

GetThreadName(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetThreadName()

Gets the logging event thread name.

The thread name

GetTimestamp(*self*) → char const *

const char* Spinnaker::LoggingEventData::GetTimestamp()

Gets the logging event time stamp.

The time stamp of the log

property thisown

The membership flag

class PySpin.PySpin.**LoggingEventDataPtr**(*args)

Bases: `_SWIG_LogPtr`

A reference tracked pointer to the LoggingEvent object.

C++ includes: `LoggingEventDataPtr.h`

property thisown

The membership flag

class PySpin.PySpin.**LoggingEventHandler**

Bases: [*ILoggingEventHandler*](#)

Proxy of C++ Spinnaker::LoggingEventHandler class.

OnLogEvent(*self*, *eventPtr*)

Parameters

eventPtr (*Spinnaker::LoggingEventDataPtr*)

property thisown

The membership flag

class PySpin.PySpin.**MJPGOption**

Bases: `object`

Options for saving MJPG files.

C++ includes: `SpinVideoDefs.h`

property frameRate

property height

property quality

property reserved

property thisown

The membership flag

property width

class PySpin.PySpin.Node(*args, **kwargs)

Bases: *INode*

class common to all nodes

C++ includes: Node.h

DeregisterCallback(self, hCallback) → bool

Parameters

- **hCallback** (*Spinnaker::GenApi::CallbackHandleType*)
- **bool** (*virtual*)
- **Spinnaker::GenApi::Node::DeregisterCallback**(*CallbackHandleType*
- **hCallback**)
- **the** (*De register change callback Destroys CNodeCallback object true if*)
- **valid** (*callback handle was*)

GetAccessMode(self) → *Spinnaker::GenApi::EAccessMode*

virtual *EAccessMode* *Spinnaker::GenApi::Node::GetAccessMode()* const

Base interface overrides.

Get the access mode of the node

GetAlias(self) → *INode*

virtual *INode** *Spinnaker::GenApi::Node::GetAlias()* const

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

GetCachingMode(self) → *Spinnaker::GenApi::ECachingMode*

virtual *ECachingMode* *Spinnaker::GenApi::Node::GetCachingMode()* const

Get Caching Mode

GetCastAlias(self) → *INode*

virtual *INode** *Spinnaker::GenApi::Node::GetCastAlias()* const

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

GetChildren(self, *LinkType=ctReadingChildren*)

Parameters

- **LinkType** (*The link type*)
- **virtual**
- **Spinnaker::GenApi::Node::GetChildren**(*GenApi::NodeList_t* (*void*)
- **&Children**
- **const** (*ELinkType LinkType=ctReadingChildren*)
- **on.** (*Get all nodes this node directly depends*)

- **Parameters**
- -----
- **Children** (*List of children nodes*)
- **LinkType**

GetDescription(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDescription() const

Get a long description of the node

GetDeviceName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDeviceName() const

Get a name of the device

GetDisplayName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDisplayName() const

Get a name string for display

GetDocuURL(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDocuURL() const

Gets a URL pointing to the documentation of that feature

GetEventID(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetEventID() const

Get the EventId of the node

GetLockNodes(*self*)

GetName(*self*, *FullQualified=False*) → *gcstring*

Parameters

- **FullQualified** (*bool*)
- **virtual**
- **Spinnaker::GenApi::Node::GetName**(*bool* (*GenICam::gcstring*))
- **const** (*FullQualified=false*)
- **name** (*Get node*)

GetNameSpace(*self*) → *Spinnaker::GenApi::ENamespace*

virtual GenApi::ENamespace Spinnaker::GenApi::Node::GetNameSpace() const

Get name space

GetNodeHandle(*self*) → *std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >*

std::shared_ptr<Node::NodeImpl> Spinnaker::GenApi::Node::GetNodeHandle() const

Get Node handle

GetNodeMap(*self*) → *INodeMap*

virtual INodeMap* Spinnaker::GenApi::Node::GetNodeMap() const

Retrieves the central node map

GetParents(*self*)

virtual void Spinnaker::GenApi::Node::GetParents(GenApi::NodeList_t &Parents) const

Gets all nodes this node is directly depending on.

6.5 Parameters:

Parents: List of parent nodes

GetPollingTime(*self*) → int64_t

virtual int64_t Spinnaker::GenApi::Node::GetPollingTime() const

recommended polling time (for not cacheable nodes)

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

virtual EInterfaceType Spinnaker::GenApi::Node::GetPrincipalInterfaceType() const

Get the type of the main interface of a node

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

Parameters

- **PropertyName** (*Spinnaker::GenICam::gcstring const &*)
- **ValueStr** (*Spinnaker::GenICam::gcstring &*)
- **AttributeStr** (*Spinnaker::GenICam::gcstring &*)
- **virtual**
- **GenICam::gcstring** (*bool Spinnaker::GenApi::Node::GetProperty(const)*)
- **&PropertyName**
- **&ValueStr** (*GenICam::gcstring*)
- **GenICam::gcstring**
- **&AttributeStr**
- **a** (*Retrieves a property plus an additional attribute by name If*)
- **as** (*property has multiple values/attribute they come with Tabs*)
- **delimiters**

GetPropertyNames(*self*)

virtual void Spinnaker::GenApi::Node::GetPropertyNames(GenICam::gcstring_vector &PropertyNames) const

Returns a list of the names all properties set during initialization

GetSelectedFeatures(*self*)

virtual void Spinnaker::GenApi::Node::GetSelectedFeatures(FeatureList_t &) const

retrieve the group of selected features

GetSelectingFeatures(*self*)

virtual void Spinnaker::GenApi::Node::GetSelectingFeatures(FeatureList_t &) const

retrieve the group of features selecting this node

GetToolTip(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetToolTip() const

Get a short description of the node

GetVisibility(*self*) → Spinnaker::GenApi::EVisibility

virtual EVisibility Spinnaker::GenApi::Node::GetVisibility() const

Get the recommended visibility of the node

ImposeAccessMode(*self*, *ImposedAccessMode*)

Parameters

- **ImposedAccessMode** (enum Spinnaker::GenApi::EAccessMode)
- **Spinnaker::GenApi::Node::ImposeAccessMode**(EAccessMode (virtual void)
- **ImposedAccessMode**)
- **node** (Imposes an access mode to the natural access mode of the)

ImposeVisibility(*self*, *ImposedVisibility*)

Parameters

- **ImposedVisibility** (enum Spinnaker::GenApi::EVisibility)
- **Spinnaker::GenApi::Node::ImposeVisibility**(EVisibility (virtual void)
- **ImposedVisibility**)
- **node** (Imposes a visibility to the natural visibility of the)

InvalidateNode(*self*)

virtual void Spinnaker::GenApi::Node::InvalidateNode()

Indicates that the node's value may have changed. Fires the callback on this and all dependent nodes

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

virtual EYesNo Spinnaker::GenApi::Node::IsAccessModeCacheable() const

True if the AccessMode can be cached

IsCachable(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsCachable() const

Is the node value cacheable

IsDeprecated(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsDeprecated() const

True if the node should not be used any more

IsFeature(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsFeature() const

True if the node can be reached via category nodes from a category node named "Root"

IsSelector(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsSelector() const

Selector interface overrides.

true if this feature selects a group of features

IsStreamable(*self*) → bool

virtual bool Spinnaker::GenApi::Node::IsStreamable() const

True if the node is streamable

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

Parameters

- **pCallback** (*Spinnaker::GenApi::CNodeCallback **)
- **CallbackHandleType** (*virtual*)
- ***pCallback** (*Spinnaker::GenApi::Node::RegisterCallback(CNodeCallback)* –
–
- **object** (*Register change callback Takes ownership of the CNodeCallback*)

SetNodeHandle(*self*, *pNodeHandle*)

Parameters

- **pNodeHandle** (*std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >*)
- **void**
- **Node::NodeImpl** (*Spinnaker::GenApi::Node::SetNodeHandle(std::shared_ptr<)*
- **pNodeHandle**) (*>*)
- **handle** (*Set Node*)

SetNodeMap(*self*, *pNodeMap*)

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap **)
- **void**
- ***pNodeMap**) (*Spinnaker::GenApi::Node::SetNodeMap(INodeMap) –*

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::ISelector **)
- **SetReference(self**
- **pBase)**
- **pBase**
- **virtual**
- ***pBase**) (*void Spinnaker::GenApi::Node::SetReference(ISelector) –*

property thisown

The membership flag

class PySpin.PySpin.**NodeCallback**

Bases: object

Proxy of C++ NodeCallback class.

CallbackFunction(*self*, *node*)

Parameters

node (*Spinnaker::GenApi::INode **)

Callback function used in node callbacks (see NodeMapCallback example for more details). Users should override this function when using node callbacks.

6.6 Parameters:

node: *INode* passed to the function during the callback.

property thisown

The membership flag

class PySpin.PySpin.**NodeMap**(*args)

Bases: *INodeMap*, *IDeviceInfo*

Smart pointer template for NodeMaps with create function.

6.7 Parameters:

TCameraParams: The camera specific parameter class (auto generated from camera xml file)

C++ includes: *NodeMap.h*

static ClearXMLCache() → bool

Connect(*self*, *pPort*, *PortName*) → bool

Parameters

- **pPort** (*IPort **)
- **PortName** (*Spinnaker::GenICam::gcstring const &*)
- **Connect**(*self*
- **bool** (*pPort*) →)
- **pPort**
- **virtual**
- **const** (*bool Spinnaker::GenApi::NodeMap::Connect(IPort *pPort)*) –
- **"Device"** (*Connects a port to the standard port*)

Destroy(*self*)

void *Spinnaker::GenApi::NodeMap::Destroy*()

Destroys the node map

GetDeviceName(*self*) → *gcstring*

virtual *GenICam::gcstring Spinnaker::GenApi::NodeMap::GetDeviceName*()

Get device name

GetDeviceVersion(*self*, *Version*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t &*)

- **void** (*virtual*)
- **Spinnaker::GenApi::NodeMap::GetDeviceVersion**(GenICam::Version_t &Version)
- **file** (*Get the version of the device description*)

GetGenApiVersion(*self*, *Version*, *Build*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t &*)
- **Build** (*uint16_t &*)
- **void** (*virtual*)
- **Spinnaker::GenApi::NodeMap::GetGenApiVersion**(GenICam::Version_t &Version, uint16_t &Build)
- **implementation** (*Get the version of the DLL's GenApi*)

GetModelName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetModelName()

Get the model name

GetNode(*self*, *key*) → *INode*

Parameters

- **key** (*Spinnaker::GenICam::gcstring const &*)
- **virtual**
- **GenICam::gcstring** (*INode* Spinnaker::GenApi::NodeMap::GetNode(const)*)
- **const** (*&key*)
- **name** (*Retrieves the node from the central map by*)

GetNodeMapHandle(*self*) → void *

void* Spinnaker::GenApi::NodeMap::GetNodeMapHandle() const

GetNodes(*self*)

virtual void Spinnaker::GenApi::NodeMap::GetNodes(NodeList_t &Nodes) const

Retrieves all nodes in the node map

GetNumNodes(*self*) → uint64_t

virtual uint64_t Spinnaker::GenApi::NodeMap::GetNumNodes() const

Get the number of nodes in the map

GetProductGuid(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetProductGuid()

Get the GUID describing the product

GetSchemaVersion(*self*, *Version*)

Parameters

- **Version** (*Spinnaker::GenICam::Version_t* &)
- **void** (*virtual*)
- **Spinnaker::GenApi::NodeMap::GetSchemaVersion**(*GenICam::Version_t* &*Version*)
- **number** (*Get the schema version*)

GetStandardNameSpace(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetStandardNameSpace()

Get the standard name space

GetSupportedSchemaVersions(*self*)

virtual void Spinnaker::GenApi::NodeMap::GetSupportedSchemaVersions(GenICam::gcstring &ng_vector &SchemaVersions)

! 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.); Gets a list of supported schema versionsEach 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

GetToolTip(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetToolTip()

Get tool tip

GetVendorName(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVendorName()

Get the vendor name

GetVersionGuid(*self*) → *gcstring*

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVersionGuid()

Get the GUID describing the product version

InvalidateNodes(*self*)

virtual void Spinnaker::GenApi::NodeMap::InvalidateNodes() const

Invalidates all nodes

LoadXMLFromFile(*self*, *FileName*)**Parameters**

- **FileName** (*Spinnaker::GenICam::gcstring*)
- **Spinnaker::GenApi::NodeMap::LoadXMLFromFile**(*GenICam::gcstring* (*void*))
- **FileName**)
- **if** (! Creates the object from the default DLL ! note Can only be used)
- **xml** (the class *TCameraParams* was auto generated from a specific camera)
- **LoadDLL(void);** (*file void*)
- **and** (! Creates the object from a DLL whose name is deduced from vendor)
- **VendorName** (*model name void LoadDLL*(*GenICam::gcstring*))

:param : :param *GenICam::gcstring* *ModelName*);: :param ! Creates the object from a DLL with given file name void: :param *LoadDLL*(*GenICam::gcstring* *FileName*); Creates the object from a XML: :param file with given file name:

LoadXMLFromFileInject(*self*, *TargetFileName*, *InjectFileName*)**Parameters**

- **TargetFileName** (*Spinnaker::GenICam::gcstring*)
- **InjectFileName** (*Spinnaker::GenICam::gcstring*)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromFileInject**(*GenICam::gcstring*
- **TargetFileName**
- **InjectFileName**) (*GenICam::gcstring*)
- **given** (*Creates the object from a XML target and an inject file with*)
- **name** (*file*)

LoadXMLFromString(*self*, *XMLData*)**Parameters**

- **XMLData** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenApi::NodeMap::LoadXMLFromString**(*const* (*void*))
- **&XMLData**) (*GenICam::gcstring*)
- **string** (*Creates the object from XML data given in a*)

LoadXMLFromStringInject (*self*, *TargetXMLData*const, *InjectXMLData*)

Parameters

- **TargetXMLData**const (*Spinnaker::GenICam::gcstring* const &)
- **InjectXMLData** (*Spinnaker::GenICam::gcstring* const &)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromStringInject**(const
- **&TargetXMLData**const (*GenICam::gcstring*)
- **GenICam::gcstring** (*const*)
- **&InjectXMLData**)
- **injection** (*Creates the object from XML data given in a string with*)

LoadXMLFromZIPData (*self*, *zipData*, *zipSize*)

Parameters

- **zipData** (*void* const *)
- **zipSize** (*size_t*)
- **void** (*void Spinnaker::GenApi::NodeMap::LoadXMLFromZIPData* (*const*)
- ***zipData**
- **zipSize**) (*size_t*)
- **string** (*Creates the object from a ZIP'd XML file given in a*)

LoadXMLFromZIPFile (*self*, *ZipFileName*)

Parameters

- **ZipFileName** (*Spinnaker::GenICam::gcstring*)
- **void**
- **Spinnaker::GenApi::NodeMap::LoadXMLFromZIPFile** (*GenICam::gcstring*
- **ZipFileName**)
- **name** (*Creates the object from a ZIP'd XML file with given file*)

Poll (*self*, *ElapsedTime*)

Parameters

- **ElapsedTime** (*int64_t*)
- **void** (*virtual*)
- **ElapsedTime**) (*Spinnaker::GenApi::NodeMap::Poll* (*int64_t*)
- **time** (*Fires nodes which have a polling*)

property thisown

The membership flag

class PySpin.PySpin.PGMOption

Bases: object

Options for saving PGM images.

C++ includes: SpinnakerDefs.h

property binaryFile

property reserved

property thisown

The membership flag

class PySpin.PySpin.PNGOption

Bases: object

Options for saving PNG images.

C++ includes: SpinnakerDefs.h

property compressionLevel

property interlaced

property reserved

property thisown

The membership flag

class PySpin.PySpin.PPMOption

Bases: object

Options for saving PPM images.

C++ includes: SpinnakerDefs.h

property binaryFile

property reserved

property thisown

The membership flag

class PySpin.PySpin.PointCloud(*args)

Bases: *IPointCloud*

Proxy of C++ Spinnaker::PointCloud class.

AddPoint(self, point)

Parameters

point (Spinnaker::Stereo3DPoint const)

GetNumPoints(self) → size_t

GetPoint(self, index) → *Stereo3DPoint*

Parameters

index (unsigned int const)

GetPointCloudData(self) → Spinnaker::IPointCloud::PointCloudData *

LoadPointCloudFromPly(*self*, *filename*)

Parameters

filename (*std::string const &*)

PrintPoints(*self*, *numPointsToPrint*)

Parameters

numPointsToPrint (*unsigned int*)

SavePointCloudAsPly(*self*, *arg0*)

Parameters

arg0 (*std::string const &*)

property thisown

The membership flag

class PySpin.PySpin.PointCloudParameters

Bases: object

Proxy of C++ Spinnaker::PointCloudParameters class.

property ROIImageBottom

property ROIImageLeft

property ROIImageRight

property ROIImageTop

property ROIWorldCoordinatesXMax

property ROIWorldCoordinatesXMin

property ROIWorldCoordinatesYMax

property ROIWorldCoordinatesYMin

property ROIWorldCoordinatesZMax

property ROIWorldCoordinatesZMin

property decimationFactor

property thisown

The membership flag

class PySpin.PySpin.RegisterNode(*args, **kwargs)

Bases: [IRegister](#), [ValueNode](#)

Interface for string properties.

C++ includes: RegisterNode.h

Get(*self*, *pBuffer*, *Length*, *Verify=False*, *IgnoreCache=False*)

Parameters

- **pBuffer** (*The buffer receiving the data to read*)
- **Length** (*The number of bytes to retrieve*)
- **Verify** (*Enables Range verification (default = false). The AccessMode*)

- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **virtual**
- ***pBuffer** (*void Spinnaker::GenApi::RegisterNode::Get(uint8_t) –*
- **int64_t**
- **Length**
- **Verify=false** (*bool*)
- **IgnoreCache=false** (*bool*)
- **contents** (*Fills a buffer with the register's*
- **Parameters**
- **-----**
- **pBuffer**
- **Length**
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

GetAddress(*self*) → int64_t

virtual int64_t Spinnaker::GenApi::RegisterNode::GetAddress()

Retrieves the Address of the register

GetLength(*self*) → int64_t

virtual int64_t Spinnaker::GenApi::RegisterNode::GetLength()

Retrieves the Length of the register [Bytes]

Set(*self, pBuffer, Verify=True*)

Parameters

- **pBuffer** (*uint8_t const **)
- **Verify** (*bool*)
- **virtual**
- ***pBuffer** (*void Spinnaker::GenApi::RegisterNode::Set(const uint8_t) –*

:param : :param int64_t Length: :param bool Verify=true): :param Set the register's contents: :param Parameters: :param ———: :param pBuffer: :type pBuffer: The buffer containing the data to set :param Length: :type Length: The number of bytes in pBuffer :param Verify: :type Verify: Enables AccessMode and Range verification (default = true)

SetReference(*self, pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)

- **Spinnaker::GenApi::RegisterNode::SetReference(INode** (*virtual void*)
- ***pBase**)
- **Register** (*overload SetReference for*)

property thisown

The membership flag

`PySpin.PySpin.RegisterNodeCallback(pNode,f)`

Parameters

- **pNode** (*Spinnaker::GenApi::INode **)
- **f** (*NodeCallback &*)

`PySpin.PySpin.ReplaceEnvironmentVariables(Buffer, ReplaceBlankBy20=False)`

Parameters

- **Buffer** (*Spinnaker::GenICam::gcstring &*)
- **ReplaceBlankBy20** (*bool*)
- **void** (*SPINNAKER_API*)
- **&Buffer** (*Spinnaker::GenICam::ReplaceEnvironmentVariables(gcstring)*)
- **bool**
- **ReplaceBlankBy20=false**)
- **%20** (*Replaces in a string and replace ' ' with*)

class `PySpin.PySpin.SIOption`

Bases: object

Proxy of C++ Spinnaker::SIOption class.

property reserved

property thisown

The membership flag

`PySpin.PySpin.SetGenICamCLProtocolFolder(path)`

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **void** (*SPINNAKER_API*)
- **&path** (*Spinnaker::GenICam::SetGenICamCLProtocolFolder(const gcstring)*)
- **folder** (*Stores the path of the CLProtocol*)

`PySpin.PySpin.SetGenICamCacheFolder(path)`

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenICam::SetGenICamCacheFolder(const** (*SPINNAKER_API void*)
- **&path**) (*gcstring*)
- **folder** (*Stores the path of the GenICam cache*)

PySpin.PySpin.SetGenICamLogConfig(*path*)

Parameters

- **path** (*Spinnaker::GenICam::gcstring const &*)
- **Spinnaker::GenICam::SetGenICamLogConfig(const (SPINNAKER_API void)**
- **&path)** (*gcstring*)
- **file** (*Stores the path of the GenICam logging properties*)

PySpin.PySpin.SetMessageCallback(*cb*)

Adds a callback to the updatator to handle messages from the updatator. Only gets called if the -P switch is present in the arguments passed to UpdateFirmware[Console]!

Parameters

cb – Function to use as callback; this function must take exactly 1 argument.

PySpin.PySpin.SetProgressCallback(*cb*)

Adds a callback to the updatator to represent update progress. Only gets called if the -P switch is present in the arguments passed to UpdateFirmware[Console]!

Parameters

cb – Function to use as callback; this function must take exactly 4 arguments.

PySpin.PySpin.SpinUpdate_SetMsgCallback(*messageCallbackFunction*)

Parameters

messageCallbackFunction (*SpinUpdate::UpdatatorMessageCallback*)

PySpin.PySpin.SpinUpdate_SetProgCallback(*progressCallbackFunction*)

Parameters

progressCallbackFunction (*SpinUpdate::UpdatatorProgressCallback*)

class PySpin.PySpin.SpinVideo

Bases: object

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

C++ includes: SpinVideo.h

Append(*self, pImage*)

Parameters

- **pImage** (*The image to append.*)
- **virtual**
- **pImage)** (*void Spinnaker::Video::SpinVideo::Append(ImagePtr)*
- **file.** (*Append an image to the AVI/MP4*)
- **Parameters**
- **-----**
- **pImage**

Close(*self*)

virtual void Spinnaker::Video::SpinVideo::Close()

Close the AVI/MP4 file.

See: Open()

Open(*self*, *pFileName*, *pOption*)**Parameters**

- **pFileName** (*The filename of the MP4 file.*)
- **pOption** (*H264 options to apply to the MP4 file.*)
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **Open**(*self*
- **pFileName**
- **pOption**)
- **pFileName**
- **pOption**
- **void** (*virtual*)
- ***pFileName** (*Spinnaker::Video::SpinVideo::Open(const char) –*
- **Video::H264Option**
- **&pOption**)
- **The** (*Open an H264 MP4 file in preparation for writing Images to disk.*
)
- **automatically** (*size of MP4 files is limited to 2GB. The filenames are*)
- **specified.** (*generated using the filename*)
- **Parameters**
- **-----**
- **pFileName**
- **pOption**
- **See** ([H264Option](#))
- **See**

SetMaximumFileSize(*self*, *size*)**Parameters**

size (*unsigned int*)

property thisown

The membership flag

class PySpin.PySpin.Stereo3DPoint

Bases: object

Proxy of C++ Spinnaker::Stereo3DPoint class.

property b**property g****property pixel****property r****property thisown**

The membership flag

property x**property y****property z****class** PySpin.PySpin.StereoCameraParameters

Bases: object

Proxy of C++ Spinnaker::StereoCameraParameters class.

property baseline**property coordinateOffset****property disparityScaleFactor****property focalLength****property invalidDataFlag****property invalidDataValue****property principalPointU****property principalPointV****property thisown**

The membership flag

class PySpin.PySpin.StringNode(*args, **kwargs)Bases: *IString*, *ValueNode*

Interface for string properties.

C++ includes: StringNode.h

GetMaxLength(self) → int64_t

virtual int64_t Spinnaker::GenApi::StringNode::GetMaxLength()

Retrieves the maximum length of the string in bytes

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::StringNode::GetValue**(*bool* *GenICam::gcstring*) (virtual)
- **Verify=false**
- **IgnoreCache=false** (*bool*)
- **value** (*Get node*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

SetReference(*self*, *pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::StringNode::SetReference(INode)*
—
- **Value** (*overload SetReference for*)

SetValue(*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*The value to set*)
- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Spinnaker::GenApi::StringNode::SetValue**(*const* *virtual void*)
- **&Value** (*GenICam::gcstring*)
- **Verify=true** (*bool*)
- **value** (*Set node*)
- **Parameters**
- -----
- **Value**
- **Verify**

property thisown

The membership flag

class PySpin.PySpin.StringRegNode(*args, **kwargs)

Bases: [StringNode](#), [RegisterNode](#)

Interface for string properties.

C++ includes: StringRegNode.h

SetReference(self, pBase)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- **Spinnaker::GenApi::StringRegNode::SetReference**(INode (virtual void)
- ***pBase**)
- **Value** (*overload SetReference for*)

property thisown

The membership flag

class PySpin.PySpin.System(*args, **kwargs)

Bases: [ISystem](#)

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

C++ includes: System.h

GetCameras(self, updateInterfaces=True, updateCameras=True) → [CameraList](#)

Parameters

- **updateInterfaces** (*Determines whether or not updateInterfaceList() is*
- **updateCameras** (*Determines whether or not UpdateCameras() is called*)
- **CameraList**
- **updateInterfaces=true** (*Spinnaker::System::GetCameras(bool)*
- **bool**
- **updateCameras=true**)
- **call** (*Returns a list of cameras that are available on the system. This*)
- **interfaces.** (*returns both GigE Vision and Usb3 Vision cameras from all*)
- **It** (*The camera list object will reference count the cameras it returns.*)
- **before** (*is important that the camera list is destroyed or is cleared*)
- **system->** (*calling system-> ReleaseInstance() or else the call to*
- **a** (*ReleaseInstance() will result in an error message thrown that*)
- **held.** (*reference to the camera is still*)
- **See** (*CameraList::Clear()*)
- **See**

- **Parameters**
- -----
- **updateInterfaces**
- **system** (*before getting cameras from available interfaces on the*)
- **updateCameras**
- **system**
- **cameras.** (*An CameraList object that contains a list of all*)

static **GetInstance()** → *SystemPtr*

GetInterfaces(*self*, *updateInterface=True*) → *InterfaceList*

Parameters

- **updateInterface** (*Determines whether or not UpdateInterfaceList() is*)
- **Spinnaker::System::GetInterfaces**(**bool** (*InterfaceList*)
- **updateInterface=true**)
- **call** (*Returns a list of interfaces available on the system. This*)
- **interfaces.** (*An InterfaceList object that contains a list of all*)
- **Parameters**
- -----
- **updateInterface**
- **interfaces** (*called before getting available*)
- **interfaces.**

GetLibraryVersion(*self*) → *LibraryVersion*

GetLoggingEventPriorityLevel(*self*) → *Spinnaker::SpinnakerLogLevel*

SpinnakerLogLevel *Spinnaker::System::GetLoggingEventPriorityLevel()*

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: *SpinnakerLogLevel*

Level The threshold level

GetTLNodeMap(*self*) → *INodeMap*

IsInUse(*self*) → bool

bool Spinnaker::System::IsInUse()

Checks if the system is in use by any interface or camera objects.

Returns true if the system is in use and false otherwise.

RegisterEventHandler(*self*, *evtHandlerToRegister*, *updateInterface=False*)

Parameters

- **evtHandlerToRegister** (*Spinnaker::EventHandler &*)
- **updateInterface** (*bool*)

RegisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

ReleaseInstance(*self*)

void Spinnaker::System::ReleaseInstance()

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: Error

See: GetInstance()

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *requestAck=False*, *pResultSize=None*, *results=0*)

Parameters

- **deviceKey** (*Spinnaker::System::SendActionCommand(unsigned int)*)
- **groupKey** (*int*)
- **groupMask** (*unsigned int*)
- **actionTime** (*unsigned long long*)
- **requestAck** (*bool*)
- **pResultSize** (*unsigned int **)
- **results** (*Spinnaker::ActionCommandResult []*)
- **void**
- **deviceKey**
- **unsigned**
- **groupKey**
- **groupMask**
- **actionTime=0** (*unsigned long long*)

:param : :param unsigned int *pResultSize=0: :param ActionCommandResult results[]=NULL): :param Broadcast an Action Command to all devices on system: :param Parameters: :param ———: :param deviceKey: :type deviceKey: The Action Command's device key :param groupKey: :type groupKey: The Action Command's group key :param groupMask: :type groupMask: The Action Command's group mask :param actionTime: :type actionTime: (Optional) Time when to assert a future action. Zero :param means

immediate action.: :param pResultSize: :type pResultSize: (Optional) The number of results in the results array. :param The value passed should be equal to the expected number of devices: :param that acknowledge the command. Returns the number of received results.: :param results: :type results: (Optional) An Array with *pResultSize elements to hold the :param action command result status. The buffer is filled starting from index: :param 0. If received results are less than expected number of devices that: :param acknowledge the command: :param remaining results are not changed. If: :param received results are more than expected number of devices that: :param acknowledge the command: :param extra results are ignored and not appended to: :param array. This parameter is ignored if pResultSize is 0. Thus this: :param parameter can be NULL if pResultSize is 0 or NULL.:

SetLoggingEventPriorityLevel(*self*, *level*)

Parameters

- **level** (*enum Spinnaker::SpinnakerLogLevel*)
- **void**
- **Spinnaker::System::SetLoggingEventPriorityLevel**(*SpinnakerLogLevel*
- *level*)
- **events** (*Sets a threshold priority level for logging event. Logging*)
- **callbacks.** (*below such level will not trigger*)
- **logging** (*Spinnaker uses five levels of*)
- **intervention.** (*Warning - failures that are recoverable without user*)
- **intervention.**
- **removal** (*Notice - information about events such as camera arrival and*)

:param : :param initialization and deinitialization: :param starting and stopping image: :param acquisition: :param and feature modification.: :param Info - information about recurring events that are generated regularly: :param such as information on individual images.: :param Debug - information that can be used to troubleshoot the system.: :param See: :type See: *SpinnakerLogLevel* :param Parameters: :param ———: :param level: :type level: The threshold level

UnregisterAllLoggingEventHandlers(*self*)

UnregisterEventHandler(*self*, *evtHandlerToUnregister*)

Parameters

evtHandlerToUnregister (*Spinnaker::EventHandler &*)

UnregisterLoggingEventHandler(*self*, *handler*)

Parameters

handler (*Spinnaker::LoggingEventHandler &*)

UpdateCameras(*self*, *updateInterfaces=True*) → bool

Parameters

- **updateInterfaces** (*bool*)
- **bool**
- **updateInterfaces=true**) (*Spinnaker::System::UpdateCameras*(*bool*)
- **that** (*Updates the list of cameras on the system. Note*)

- **each** (*System::GetCameras()* internally calls *UpdateCameras()* for)
- **the** (*interface it enumerates. If the list changed between this call and*)
- **true** (*last time UpdateCameras was called then the return value will be*)

:param : :param otherwise it is false.: :param See: :type See: *GetCameras()* :param Parameters: :param ———: :param *updateInterfaces*: :type *updateInterfaces*: Determines whether or not *UpdateInterfaceList()* is :param called before updating cameras for available interfaces on the system: :param True if cameras changed on interface and false otherwise.:

UpdateInterfaceList(*self*)

property thisown

The membership flag

class PySpin.PySpin.**SystemEventHandler**

Bases: *ISystemEventHandler*

Proxy of C++ Spinnaker::SystemEventHandler class.

OnInterfaceArrival(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

OnInterfaceRemoval(*self*, *pInterface*)

Parameters

pInterface (*Spinnaker::InterfacePtr*)

property thisown

The membership flag

class PySpin.PySpin.**SystemPtr**(*args)

Bases: *_SWIG_SysPtr*

A reference tracked pointer to a system object.

C++ includes: *SystemPtr.h*

property thisown

The membership flag

class PySpin.PySpin.**TIFFOption**

Bases: *object*

Options for saving TIFF images.

C++ includes: *SpinnakerDefs.h*

property compression

property reserved

property thisown

The membership flag

PySpin.PySpin.**ThrowBadAlloc**()

SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc()

PySpin.PySpin.Tokenize(*str*, *delimiters*=' ')

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **delimiters** (*Spinnaker::GenICam::gcstring const &*)
- **SPINNAKER_API**
- **&str** (*void Spinnaker::GenICam::Tokenize(const gcstring)*)
- **gcstring_vector**
- **&tokens**
- **)** (*const gcstring &delimiters=""*)
- **delimiter** (*splits str input string into a list of tokens using the*)

class PySpin.PySpin.TransportLayerDevice(*nodeMapTLDevice*)

Bases: object

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

C++ includes: TransportLayerDevice.h

property DeviceAccessStatus

property DeviceBootloaderVersion

property DeviceCurrentSpeed

property DeviceDisplayName

property DeviceDriverVersion

property DeviceEndianessMechanism

property DeviceID

property DeviceInstanceId

property DeviceIsUpdater

property DeviceLinkSpeed

property DeviceLocation

property DeviceModelName

property DeviceMulticastMonitorMode

property DevicePortId

property DeviceReset

property DeviceSerialNumber

property DeviceType

property DeviceU3VProtocol

property DeviceUserID

property DeviceVendorName
property DeviceVersion
property GUIXMLLocation
property GUIXMLPath
property GenICamXMLLocation
property GenICamXMLPath
property GevCCP
property GevDeviceAutoForceIP
property GevDeviceDiscoverMaximumPacketSize
property GevDeviceForceGateway
property GevDeviceForceIP
property GevDeviceForceIPAddress
property GevDeviceForceSubnetMask
property GevDeviceGateway
property GevDeviceIPAddress
property GevDeviceIsWrongSubnet
property GevDeviceMACAddress
property GevDeviceMaximumPacketSize
property GevDeviceMaximumRetryCount
property GevDeviceModeIsBigEndian
property GevDevicePort
property GevDeviceReadAndWriteTimeout
property GevDeviceSubnetMask
property GevVersionMajor
property GevVersionMinor
property StreamID
property StreamSelector
property thisown

The membership flag

class PySpin.PySpin.TransportLayerInterface(*nodeMapTLDevice*)

Bases: object

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

C++ includes: TransportLayerInterface.h

property ActionCommand

property DeviceAccessStatus

property DeviceCount

property DeviceID

property DeviceModelName

property DeviceSelector

property DeviceSerialNumber

property DeviceUnlock

property DeviceUpdateList

property DeviceVendorName

property FLIRFilterDriverStatus

property GevActionAckRequired

property GevActionDeviceKey

property GevActionGroupKey

property GevActionGroupMask

property GevActionTime

property GevDeviceAutoForceIP

property GevDeviceDisableDiscovery

property GevDeviceDiscoveryEnabled

property GevDeviceEnableDiscovery

property GevDeviceForceGateway

property GevDeviceForceIP

property GevDeviceForceIPAddress

property GevDeviceForceSubnetMask

property GevDeviceGateway

property GevDeviceIPAddress

property GevDeviceMACAddress

property `GevDeviceSubnetMask`
property `GevInterfaceGateway`
property `GevInterfaceGatewaySelector`
property `GevInterfaceIsIPConflict`
property `GevInterfaceMACAddress`
property `GevInterfaceMTU`
property `GevInterfaceReceiveLinkSpeed`
property `GevInterfaceSubnetIPAddress`
property `GevInterfaceSubnetMask`
property `GevInterfaceSubnetSelector`
property `GevInterfaceTransmitLinkSpeed`
property `HostAdapterDriverVersion`
property `HostAdapterName`
property `HostAdapterVendor`
property `IncompatibleDeviceCount`
property `IncompatibleDeviceID`
property `IncompatibleDeviceModelName`
property `IncompatibleDeviceSelector`
property `IncompatibleDeviceVendorName`
property `IncompatibleGevDeviceIPAddress`
property `IncompatibleGevDeviceMACAddress`
property `IncompatibleGevDeviceSubnetMask`
property `InterfaceDisplayName`
property `InterfaceID`
property `InterfaceType`
property `POEStatus`
property `TeledyneGigeVisionFilterDriverStatus`
property `thisown`

The membership flag

class PySpin.PySpin.TransportLayerStream(*nodeMapTLDevice*)

Bases: object

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

C++ includes: TransportLayerStream.h

property StreamAnnounceBufferMinimum

property StreamAnnouncedBufferCount

property StreamBlockTransferSize

property StreamBlocksProcessingTimeLast

property StreamBlocksProcessingTimeMax

property StreamBlocksProcessingTimeMin

property StreamBlocksReceptionTimeLast

property StreamBlocksReceptionTimeMax

property StreamBlocksReceptionTimeMin

property StreamBufferAlignment

property StreamBufferCountManual

property StreamBufferCountMax

property StreamBufferCountMode

property StreamBufferCountResult

property StreamBufferHandlingMode

property StreamCRCCheckEnable

property StreamChunkCountMaximum

property StreamDeliveredFrameCount

property StreamDroppedFrameCount

property StreamID

property StreamIncompleteFrameCount

property StreamInputBufferCount

property StreamIsGrabbing

property StreamLostFrameCount

property StreamMissedPacketCount

property StreamMode

property StreamOutputBufferCount

property StreamPacketResendEnable
 property StreamPacketResendMaxRequests
 property StreamPacketResendReceivedPacketCount
 property StreamPacketResendRequestCount
 property StreamPacketResendRequestTimeoutCount
 property StreamPacketResendRequestedPacketCount
 property StreamPacketResendTimeout
 property StreamPacketsDuplicatedCount
 property StreamPacketsNotYetAvailableCount
 property StreamPacketsPerFrameCount
 property StreamPacketsTemporarilyUnavailableCount
 property StreamPacketsTimeoutCount
 property StreamPacketsUnavailableCount
 property StreamReceivedFrameCount
 property StreamReceivedPacketCount
 property StreamStartedFrameCount
 property StreamType
 property thisown

The membership flag

class PySpin.PySpin.TransportLayerSystem(*nodeMapTLDevice*)

Bases: object

Part of the QuickSpin API to provide access to system information.

C++ includes: TransportLayerSystem.h

property EnumerateGEVInterfaces
 property EnumerateGen2Cameras
 property EnumerateUSBInterfaces
 property GEVAutoAssignIPEnable
 property GenTL_SFNC_Version_Major
 property GenTL_SFNC_Version_Minor
 property GenTL_SFNC_Version_SubMinor
 property GenTL_Version_Major
 property GenTL_Version_Minor

`property` `GevInterfaceDefaultGateway`
`property` `GevInterfaceDefaultIPAddress`
`property` `GevInterfaceDefaultSubnetMask`
`property` `GevInterfaceMACAddress`
`property` `GevVersionMajor`
`property` `GevVersionMinor`
`property` `InterfaceDisplayName`
`property` `InterfaceID`
`property` `InterfaceSelector`
`property` `InterfaceUpdateList`
`property` `TLDisplayName`
`property` `TLFileName`
`property` `TLID`
`property` `TLModelName`
`property` `TLPath`
`property` `TLType`
`property` `TLVendorName`
`property` `TLVersion`
`property` `thisown`

The membership flag

`PySpin.PySpin.UpdateFirmware(args) → int`

Parameters

`args` (`std::vector< std::string >`)

`PySpin.PySpin.UpdateFirmwareConsole(numArgs) → int`

Parameters

`numArgs` (`unsigned int`)

`PySpin.PySpin.UpdateFirmwareGUI(args) → int`

Parameters

`args` (`std::string`)

`PySpin.PySpin.UrlDecode(Input) → gcstring`

Parameters

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

- **equivalent** (*Replaces xx escapes by their char*)

PySpin.PySpin.UrlEncode(*Input*) → *gcstring*

Parameters

- **Input** (*Spinnaker::GenICam::gcstring const &*)
- **SPINNAKER_API**
- **&Input** (*gcstring Spinnaker::GenICam::UrlEncode(const gcstring)*)
- **xx** (*Converts \ to / and replaces all unsafe characters by their*)
- **equivalent**

class PySpin.PySpin.ValueNode(*args, **kwargs)

Bases: *IValue, Node*

Interface for value properties.

C++ includes: ValueNode.h

FromString(*self, ValueStr, Verify=True*)

Parameters

- **ValueStr** (*The value to set*)
- **Verify** (*Enables AccessMode and Range verification (default = true)*)
- **Spinnaker::GenApi::ValueNode::FromString(const (virtual void)**
- **&ValueStr** (*GenICam::gcstring*)
- **Verify=true** (*bool*)
- **string** (*Set content of the node as*)
- **Parameters**
- -----
- **ValueStr**
- **Verify**

GetNode(*self*) → *INode*

virtual INode* Spinnaker::GenApi::ValueNode::GetNode()

IsValueCacheValid(*self*) → bool

virtual bool Spinnaker::GenApi::ValueNode::IsValueCacheValid() const

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

SetReference(*self, pBase*)

Parameters

- **pBase** (*Spinnaker::GenApi::INode **)
- ***pBase** (*virtual void Spinnaker::GenApi::ValueNode::SetReference(INode)*
—
- **Value** (*overload SetReference for*)

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → *gcstring*

Parameters

- **Verify** (*Enables Range verification (default = false). The AccessMode*)
- **IgnoreCache** (*If true the value is read ignoring any caches (default =)*)
- **Spinnaker::GenApi::ValueNode::ToString**(*bool* *GenICam::gcstring*) (virtual)
- **Verify=false**
- **IgnoreCache=false**) (*bool*)
- **string** (*Get content of the node as*)
- **Parameters**
- -----
- **Verify**
- **checked** (*is always*)
- **IgnoreCache**
- **false**)
- **read** (*The value*)

property thisown

The membership flag

class PySpin.PySpin.**Version_t**

Bases: object

Version

C++ includes: GCTypes.h

property Major

property Minor

property SubMinor

property thisown

The membership flag

class PySpin.PySpin.**double_autovector_t**(*args)

Bases: object

Vector of doubles with reference counting.

C++ includes: Autovector.h

size(*self*) → *size_t*

size_t Spinnaker::GenApi::double_autovector_t::size() const

property thisown

The membership flag

class PySpin.PySpin.gcstring(*args)

Bases: object

Proxy of C++ Spinnaker::GenICam::gcstring class.

append(self, str) → gcstring

Parameters

- **str** (Spinnaker::GenICam::gcstring const &)
- **append**(self
- **count** (gcstring& Spinnaker::GenICam::gcstring::append(size_t)
- **gcstring** (ch) ->)
- **count**
- **ch** (char)
- **virtual**
- **count**
- **ch** (char)

assign(self, str) → gcstring

Parameters

- **str** (Spinnaker::GenICam::gcstring const &)
- **assign**(self
- **count** (size_t)
- **gcstring** (n) ->)
- **count**
- **ch** (char)
- **assign**(self
- **gcstring**
- **pc** (char const *)
- **assign**(self
- **pc**
- **gcstring**
- **pc**
- **n** (size_t)
- **virtual**
- ***pc** (gcstring& Spinnaker::GenICam::gcstring::assign(const char) –
- **size_t**
- **n**)

c_str(self) → char const *

virtual const char* Spinnaker::GenICam::gcstring::c_str(void) const

compare(*self*, *str*) → int

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **virtual**
- **const** (*int Spinnaker::GenICam::gcstring::compare(const gcstring &str)*)

empty(*self*) → bool

virtual bool Spinnaker::GenICam::gcstring::empty(void) const

find(*self*, *ch*, *offset=0*) → size_t

Parameters

- **ch** (*char*)
- **offset** (*size_t*)
- **find(self**
- **str** (*Spinnaker::GenICam::gcstring const &*)
- **size_t** (*count*) ->)
- **str**
- **offset**
- **find(self**
- **str**
- **offset**
- **size_t**
- **str**
- **offset**
- **count** (*size_t*)
- **find(self**
- **pc** (*char const **)
- **size_t**
- **pc**
- **offset**
- **find(self**
- **pc**
- **offset**
- **size_t**
- **pc**
- **offset**
- **count**

- **virtual**
- ***pc** (*size_t Spinnaker::GenICam::gcstring::find(const char) –*
- **size_t**
- **offset**
- **const** (*size_t count*)

find_first_not_of(*self, str, offset=0*) → *size_t*

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **offset** (*size_t*)
- **Spinnaker::GenICam::gcstring::find_first_not_of**(*const (virtual size_t*
- **&str** (*gcstring*)
- **const** (*size_t offset=0*)

find_first_of(*self, str, offset=0*) → *size_t*

Parameters

- **str** (*Spinnaker::GenICam::gcstring const &*)
- **offset** (*size_t*)
- **Spinnaker::GenICam::gcstring::find_first_of**(*const (virtual size_t*
- **&str** (*gcstring*)
- **const** (*size_t offset=0*)

length(*self*) → *size_t*

virtual size_t Spinnaker::GenICam::gcstring::length(void) const

max_size(*self*) → *size_t*

virtual size_t Spinnaker::GenICam::gcstring::max_size() const

npos = 18446744073709551615

resize(*self, n*)

Parameters

- **n** (*size_t*)
- **virtual**
- **n** (*void Spinnaker::GenICam::gcstring::resize(size_t*

size(*self*) → *size_t*

virtual size_t Spinnaker::GenICam::gcstring::size(void) const

substr(*self, offset=0, count=size_t(-1)*) → *gcstring*

Parameters

- **offset** (*size_t*)
- **count** (*size_t*)

- **virtual**
- **offset=0** (*gcstring Spinnaker::GenICam::gcstring::substr(size_t)*)
- **size_t**
- **const** (*count=GCSTRING_NPOS*)

swap(*self*, *Right*)

Parameters

- **Right** (*Spinnaker::GenICam::gcstring &*)
- **virtual**
- **&Right** (*void Spinnaker::GenICam::gcstring::swap(gcstring)*)

property thisown

The membership flag

class PySpin.PySpin.**int64_autovector_t**(*args)

Bases: object

Vector of integers with reference counting.

C++ includes: Autovector.h

size(*self*) → *size_t*

size_t Spinnaker::GenApi::int64_autovector_t::size() const

property thisown

The membership flag

class PySpin.PySpin.**node_vector**(*args)

Bases: object

Proxy of C++ Spinnaker::GenApi::node_vector class.

assign(*self*, *n*, *val*)

Parameters

- **n** (*size_t*)
- **val** (*Spinnaker::GenApi::node_vector::T const &*)

at(*self*, *uiIndex*) → *INode*

Parameters

- **uiIndex** (*size_t*)
- **at**(*self*
- **INode** (*uiIndex*) →>)
- **uiIndex**

back(*self*) → *INode*

back(*self*) → *INode*

begin(*self*) → Spinnaker::GenApi::node_vector::iterator

begin(*self*) → Spinnaker::GenApi::node_vector::const_iterator

capacity(*self*) → size_t

clear(*self*)

empty(*self*) → bool

end(*self*) → Spinnaker::GenApi::node_vector::iterator

end(*self*) → Spinnaker::GenApi::node_vector::const_iterator

erase(*self*, *pos*) → Spinnaker::GenApi::node_vector::iterator

Parameters

- **pos** (Spinnaker::GenApi::node_vector::iterator)
- **erase**(*self*
- **uiIndex**)
- **uiIndex** (size_t)

front(*self*) → *INode*

front(*self*) → *INode*

insert(*self*, *pos*, *val*) → Spinnaker::GenApi::node_vector::iterator

Parameters

- **pos** (Spinnaker::GenApi::node_vector::iterator)
- **val** (Spinnaker::GenApi::node_vector::T const &)
- **insert**(*self*
- **uiIndex** (size_t)
- **val**)
- **uiIndex**
- **val**

max_size(*self*) → size_t

pop_back(*self*)

push_back(*self*, *val*)

Parameters

- **val** (Spinnaker::GenApi::node_vector::T const &)

reserve(*self*, *uiSize*)

Parameters

- **uiSize** (size_t)

resize(*self*, *uiSize*)

Parameters

- **uiSize** (size_t)

size(*self*) → size_t

property thisown

The membership flag

class PySpin.PySpin.**value_vector**(*args)

Bases: object

Proxy of C++ Spinnaker::GenApi::value_vector class.

assign(self, n, obj)**Parameters**

- **n** (size_t)
- **obj** (Spinnaker::GenApi::value_vector::T const &)

at(self, uiIndex) → IValue**Parameters**

- **uiIndex** (size_t)
- **at**(self
- **IValue** (uiIndex) ->)
- **uiIndex**

back(self) → IValue**back**(self) → IValue**begin**(self) → Spinnaker::GenApi::value_vector::iterator**begin**(self) → Spinnaker::GenApi::value_vector::const_iterator**capacity**(self) → size_t**clear**(self)**empty**(self) → bool**end**(self) → Spinnaker::GenApi::value_vector::iterator**end**(self) → Spinnaker::GenApi::value_vector::const_iterator**erase**(self, pos) → Spinnaker::GenApi::value_vector::iterator**Parameters**

- **pos** (Spinnaker::GenApi::value_vector::iterator)
- **erase**(self
- **uiIndex**)
- **uiIndex** (size_t)

front(self) → IValue**front**(self) → IValue**insert**(self, pos, val) → Spinnaker::GenApi::value_vector::iterator**Parameters**

- **pos** (Spinnaker::GenApi::value_vector::iterator)

- **val** (*Spinnaker::GenApi::value_vector::T const &*)
- **insert**(*self*
- **uiIndex** (*size_t*)
- **val**)
- **uiIndex**
- **val**

max_size(*self*) → *size_t*

pop_back(*self*)

push_back(*self*, *val*)

Parameters

- val** (*Spinnaker::GenApi::value_vector::T const &*)

reserve(*self*, *uiSize*)

Parameters

- uiSize** (*size_t*)

resize(*self*, *uiSize*, *val*)

Parameters

- **uiSize** (*size_t*)
- **val** (*Spinnaker::GenApi::value_vector::T const &*)

size(*self*) → *size_t*

property thisown

The membership flag

PYTHON MODULE INDEX

p

`PySpin.PySpin`, 81

A

- AasRoiEnable (*PySpin.Camera* property), 10
- AasRoiEnable (*PySpin.PySpin.Camera* property), 115
- AasRoiHeight (*PySpin.Camera* property), 10
- AasRoiHeight (*PySpin.PySpin.Camera* property), 115
- AasRoiOffsetX (*PySpin.Camera* property), 10
- AasRoiOffsetX (*PySpin.PySpin.Camera* property), 115
- AasRoiOffsetY (*PySpin.Camera* property), 10
- AasRoiOffsetY (*PySpin.PySpin.Camera* property), 115
- AasRoiWidth (*PySpin.Camera* property), 10
- AasRoiWidth (*PySpin.PySpin.Camera* property), 115
- AcquisitionAbort (*PySpin.Camera* property), 10
- AcquisitionAbort (*PySpin.PySpin.Camera* property), 115
- AcquisitionArm (*PySpin.Camera* property), 10
- AcquisitionArm (*PySpin.PySpin.Camera* property), 116
- AcquisitionBurstFrameCount (*PySpin.Camera* property), 10
- AcquisitionBurstFrameCount (*PySpin.PySpin.Camera* property), 116
- AcquisitionFrameCount (*PySpin.Camera* property), 10
- AcquisitionFrameCount (*PySpin.PySpin.Camera* property), 116
- AcquisitionFrameRate (*PySpin.Camera* property), 10
- AcquisitionFrameRate (*PySpin.PySpin.Camera* property), 116
- AcquisitionFrameRateEnable (*PySpin.Camera* property), 10
- AcquisitionFrameRateEnable (*PySpin.PySpin.Camera* property), 116
- AcquisitionFrameRatePersistence (*PySpin.Camera* property), 10
- AcquisitionFrameRatePersistence (*PySpin.PySpin.Camera* property), 116
- AcquisitionLineRate (*PySpin.Camera* property), 10
- AcquisitionLineRate (*PySpin.PySpin.Camera* property), 116
- AcquisitionMode (*PySpin.Camera* property), 10
- AcquisitionMode (*PySpin.PySpin.Camera* property), 116
- AcquisitionResultingFrameRate (*PySpin.Camera* property), 10
- AcquisitionResultingFrameRate (*PySpin.PySpin.Camera* property), 116
- AcquisitionStart (*PySpin.Camera* property), 10
- AcquisitionStart (*PySpin.PySpin.Camera* property), 116
- AcquisitionStatus (*PySpin.Camera* property), 10
- AcquisitionStatus (*PySpin.PySpin.Camera* property), 116
- AcquisitionStatusSelector (*PySpin.Camera* property), 10
- AcquisitionStatusSelector (*PySpin.PySpin.Camera* property), 116
- AcquisitionStop (*PySpin.Camera* property), 10
- AcquisitionStop (*PySpin.PySpin.Camera* property), 116
- AcquisitionTransferFrameRate (*PySpin.Camera* property), 11
- AcquisitionTransferFrameRate (*PySpin.PySpin.Camera* property), 116
- ActionCommand (*PySpin.PySpin.TransportLayerInterface* property), 398
- ActionCommand (*PySpin.TransportLayerInterface* property), 77
- ActionCommandResult (class in *PySpin.PySpin*), 81
- ActionDeviceKey (*PySpin.Camera* property), 11
- ActionDeviceKey (*PySpin.PySpin.Camera* property), 116
- ActionGroupKey (*PySpin.Camera* property), 11
- ActionGroupKey (*PySpin.PySpin.Camera* property), 116
- ActionGroupMask (*PySpin.Camera* property), 11
- ActionGroupMask (*PySpin.PySpin.Camera* property), 116
- ActionQueueEmpty (*PySpin.Camera* property), 11
- ActionQueueEmpty (*PySpin.PySpin.Camera* property), 116
- ActionQueueSize (*PySpin.Camera* property), 11
- ActionQueueSize (*PySpin.PySpin.Camera* property), 116
- ActionSelector (*PySpin.Camera* property), 11

- ActionSelector (*PySpin.PySpin.Camera* property), 116
- ActionSignalSize (*PySpin.Camera* property), 11
- ActionSignalSize (*PySpin.PySpin.Camera* property), 116
- ActionUnconditionalMode (*PySpin.Camera* property), 11
- ActionUnconditionalMode (*PySpin.PySpin.Camera* property), 116
- AdaptiveCompressionEnable (*PySpin.Camera* property), 11
- AdaptiveCompressionEnable (*PySpin.PySpin.Camera* property), 116
- AdcBitDepth (*PySpin.Camera* property), 11
- AdcBitDepth (*PySpin.PySpin.Camera* property), 116
- Add() (*PySpin.CameraList* method), 39
- Add() (*PySpin.ImageList* method), 55
- Add() (*PySpin.InterfaceList* method), 67
- Add() (*PySpin.PySpin.CameraList* method), 145
- Add() (*PySpin.PySpin.ICameraList* method), 171
- Add() (*PySpin.PySpin.IImageList* method), 324
- Add() (*PySpin.PySpin.IInterfaceList* method), 329
- Add() (*PySpin.PySpin.ImageList* method), 349
- Add() (*PySpin.PySpin.InterfaceList* method), 364
- AddPoint() (*PySpin.PointCloud* method), 68
- AddPoint() (*PySpin.PySpin.IPointCloud* method), 334
- AddPoint() (*PySpin.PySpin.PointCloud* method), 383
- aPAUSEMACCtrlFramesReceived (*PySpin.Camera* property), 35
- aPAUSEMACCtrlFramesReceived (*PySpin.PySpin.Camera* property), 140
- aPAUSEMACCtrlFramesTransmitted (*PySpin.Camera* property), 35
- aPAUSEMACCtrlFramesTransmitted (*PySpin.PySpin.Camera* property), 140
- Append() (*PySpin.CameraList* method), 39
- Append() (*PySpin.ImageList* method), 55
- Append() (*PySpin.InterfaceList* method), 67
- Append() (*PySpin.PySpin.CameraList* method), 145
- append() (*PySpin.PySpin.gcstring* method), 405
- Append() (*PySpin.PySpin.ICameraList* method), 171
- Append() (*PySpin.PySpin.IImageList* method), 324
- Append() (*PySpin.PySpin.IInterfaceList* method), 329
- Append() (*PySpin.PySpin.ImageList* method), 349
- Append() (*PySpin.PySpin.InterfaceList* method), 364
- Append() (*PySpin.PySpin.SpinVideo* method), 387
- Append() (*PySpin.SpinVideo* method), 69
- Application (*PySpin.PySpin.CCMSettings* property), 85
- ApplicationToString() (*PySpin.ImageUtilityCCM* static method), 59
- ApplicationToString() (*PySpin.PySpin.ImageUtilityCCM* static method), 353
- ApplyGamma() (*PySpin.ImageProcessor* method), 56
- ApplyGamma() (*PySpin.PySpin.IImageProcessor* method), 326
- ApplyGamma() (*PySpin.PySpin.ImageProcessor* method), 350
- assign() (*PySpin.PySpin.gcstring* method), 405
- assign() (*PySpin.PySpin.node_vector* method), 408
- assign() (*PySpin.PySpin.value_vector* method), 410
- at() (*PySpin.PySpin.node_vector* method), 408
- at() (*PySpin.PySpin.value_vector* method), 410
- AutoAlgorithmSelector (*PySpin.Camera* property), 11
- AutoAlgorithmSelector (*PySpin.PySpin.Camera* property), 116
- AutoExposureControlLoopDamping (*PySpin.Camera* property), 11
- AutoExposureControlLoopDamping (*PySpin.PySpin.Camera* property), 116
- AutoExposureControlPriority (*PySpin.Camera* property), 11
- AutoExposureControlPriority (*PySpin.PySpin.Camera* property), 116
- AutoExposureEVCompensation (*PySpin.Camera* property), 11
- AutoExposureEVCompensation (*PySpin.PySpin.Camera* property), 116
- AutoExposureExposureTimeLowerLimit (*PySpin.Camera* property), 11
- AutoExposureExposureTimeLowerLimit (*PySpin.PySpin.Camera* property), 116
- AutoExposureExposureTimeUpperLimit (*PySpin.Camera* property), 11
- AutoExposureExposureTimeUpperLimit (*PySpin.PySpin.Camera* property), 116
- AutoExposureGainLowerLimit (*PySpin.Camera* property), 11
- AutoExposureGainLowerLimit (*PySpin.PySpin.Camera* property), 116
- AutoExposureGainUpperLimit (*PySpin.Camera* property), 11
- AutoExposureGainUpperLimit (*PySpin.PySpin.Camera* property), 117
- AutoExposureGreyValueLowerLimit (*PySpin.Camera* property), 11
- AutoExposureGreyValueLowerLimit (*PySpin.PySpin.Camera* property), 117
- AutoExposureGreyValueUpperLimit (*PySpin.Camera* property), 11
- AutoExposureGreyValueUpperLimit (*PySpin.PySpin.Camera* property), 117
- AutoExposureLightingMode (*PySpin.Camera* property), 11
- AutoExposureLightingMode (*PySpin.PySpin.Camera* property), 117

- AutoExposureMeteringMode (*PySpin.Camera* property), 11
- AutoExposureMeteringMode (*PySpin.PySpin.Camera* property), 117
- AutoExposureTargetGreyValue (*PySpin.Camera* property), 11
- AutoExposureTargetGreyValue (*PySpin.PySpin.Camera* property), 117
- AutoExposureTargetGreyValueAuto (*PySpin.Camera* property), 11
- AutoExposureTargetGreyValueAuto (*PySpin.PySpin.Camera* property), 117
- AVIOption (class in *PySpin.PySpin*), 81
- ## B
- b (*PySpin.PySpin.Stereo3DPoint* property), 389
- back() (*PySpin.PySpin.node_vector* method), 408
- back() (*PySpin.PySpin.value_vector* method), 410
- BalanceRatio (*PySpin.Camera* property), 11
- BalanceRatio (*PySpin.PySpin.Camera* property), 117
- BalanceRatioSelector (*PySpin.Camera* property), 11
- BalanceRatioSelector (*PySpin.PySpin.Camera* property), 117
- BalanceWhiteAuto (*PySpin.Camera* property), 11
- BalanceWhiteAuto (*PySpin.PySpin.Camera* property), 117
- BalanceWhiteAutoDamping (*PySpin.Camera* property), 11
- BalanceWhiteAutoDamping (*PySpin.PySpin.Camera* property), 117
- BalanceWhiteAutoLowerLimit (*PySpin.Camera* property), 11
- BalanceWhiteAutoLowerLimit (*PySpin.PySpin.Camera* property), 117
- BalanceWhiteAutoProfile (*PySpin.Camera* property), 11
- BalanceWhiteAutoProfile (*PySpin.PySpin.Camera* property), 117
- BalanceWhiteAutoUpperLimit (*PySpin.Camera* property), 12
- BalanceWhiteAutoUpperLimit (*PySpin.PySpin.Camera* property), 117
- baseline (*PySpin.PySpin.StereoCameraParameters* property), 389
- begin() (*PySpin.PySpin.node_vector* method), 408
- begin() (*PySpin.PySpin.value_vector* method), 410
- BeginAcquisition() (*PySpin.CameraBase* method), 35
- BeginAcquisition() (*PySpin.PySpin.CameraBase* method), 140
- BeginAcquisition() (*PySpin.PySpin.ICameraBase* method), 168
- binaryFile (*PySpin.PySpin.PGMOption* property), 383
- binaryFile (*PySpin.PySpin.PPMOption* property), 383
- BinningHorizontal (*PySpin.Camera* property), 12
- BinningHorizontal (*PySpin.PySpin.Camera* property), 117
- BinningHorizontalMode (*PySpin.Camera* property), 12
- BinningHorizontalMode (*PySpin.PySpin.Camera* property), 117
- BinningSelector (*PySpin.Camera* property), 12
- BinningSelector (*PySpin.PySpin.Camera* property), 117
- BinningVertical (*PySpin.Camera* property), 12
- BinningVertical (*PySpin.PySpin.Camera* property), 117
- BinningVerticalMode (*PySpin.Camera* property), 12
- BinningVerticalMode (*PySpin.PySpin.Camera* property), 117
- bitrate (*PySpin.PySpin.H264Option* property), 167
- BlackLevel (*PySpin.Camera* property), 12
- BlackLevel (*PySpin.PySpin.Camera* property), 117
- BlackLevelAuto (*PySpin.Camera* property), 12
- BlackLevelAuto (*PySpin.PySpin.Camera* property), 117
- BlackLevelAutoBalance (*PySpin.Camera* property), 12
- BlackLevelAutoBalance (*PySpin.PySpin.Camera* property), 117
- BlackLevelClampingEnable (*PySpin.Camera* property), 12
- BlackLevelClampingEnable (*PySpin.PySpin.Camera* property), 117
- BlackLevelRaw (*PySpin.Camera* property), 12
- BlackLevelRaw (*PySpin.PySpin.Camera* property), 117
- BlackLevelSelector (*PySpin.Camera* property), 12
- BlackLevelSelector (*PySpin.PySpin.Camera* property), 117
- BMPOption (class in *PySpin.PySpin*), 81
- BooleanNode (class in *PySpin.PySpin*), 81
- bottomRightXCoord (*PySpin.PySpin.InferenceBoxRect* property), 361
- bottomRightXCoord (*PySpin.PySpin.InferenceBoxRotatedRect* property), 361
- bottomRightYCoord (*PySpin.PySpin.InferenceBoxRect* property), 361
- bottomRightYCoord (*PySpin.PySpin.InferenceBoxRotatedRect* property), 361
- boxType (*PySpin.PySpin.InferenceBoundingBox* property), 360
- BsiFlatFieldCorrectionAuto (*PySpin.Camera* property), 12
- BsiFlatFieldCorrectionAuto (*PySpin.PySpin.Camera* property), 117
- BsiFlatFieldCorrectionAutoDamping (*PySpin.Camera* property), 12
- BsiFlatFieldCorrectionAutoDamping

(*PySpin.PySpin.Camera* property), 117
BsiFlatFieldCorrectionEnable (*PySpin.Camera* property), 12
BsiFlatFieldCorrectionEnable (*PySpin.PySpin.Camera* property), 117
BsiFlatFieldCorrectionGain (*PySpin.Camera* property), 12
BsiFlatFieldCorrectionGain (*PySpin.PySpin.Camera* property), 117
BsiFlatFieldCorrectionGainSelector (*PySpin.Camera* property), 12
BsiFlatFieldCorrectionGainSelector (*PySpin.PySpin.Camera* property), 117
BufferedBurstFrameCountMax (*PySpin.Camera* property), 12
BufferedBurstFrameCountMax (*PySpin.PySpin.Camera* property), 117
BufferedBurstMode (*PySpin.Camera* property), 12
BufferedBurstMode (*PySpin.PySpin.Camera* property), 118
build (*PySpin.PySpin.LibraryVersion* property), 371

C

c_str() (*PySpin.PySpin.gcstring* method), 405
CalculateChannelStatistics() (*PySpin.PySpin.IImage* method), 320
CalculateStatistics() (*PySpin.PySpin.IImage* method), 320
CallbackFunction() (*PySpin.PySpin.NodeCallback* method), 377
Camera (class in *PySpin*), 10
Camera (class in *PySpin.PySpin*), 115
CameraBase (class in *PySpin*), 35
CameraBase (class in *PySpin.PySpin*), 140
CameraList (class in *PySpin*), 39
CameraList (class in *PySpin.PySpin*), 145
CameraPtr (class in *PySpin*), 41
CameraPtr (class in *PySpin.PySpin*), 147
capacity() (*PySpin.PySpin.node_vector* method), 408
capacity() (*PySpin.PySpin.value_vector* method), 410
CategoryNode (class in *PySpin.PySpin*), 147
CBasePtr (class in *PySpin*), 10
CBasePtr (class in *PySpin.PySpin*), 83
CBooleanPtr (class in *PySpin.PySpin*), 83
CCategoryPtr (class in *PySpin.PySpin*), 86
CCMSettings (class in *PySpin.PySpin*), 85
CCommandPtr (class in *PySpin.PySpin*), 88
CDeviceInfoPtr (class in *PySpin.PySpin*), 91
centerXCoord (*PySpin.PySpin.InferenceBoxCircle* property), 360
centerYCoord (*PySpin.PySpin.InferenceBoxCircle* property), 361
CEnumEntryPtr (class in *PySpin.PySpin*), 91
CEnumerationPtr (class in *PySpin.PySpin*), 94

CFeatureBag (class in *PySpin.PySpin*), 97
CFloatPtr (class in *PySpin.PySpin*), 98
channel (*PySpin.ChannelStatistics* property), 42
channel (*PySpin.PySpin.ChannelStatistics* property), 148
ChannelStatistics (class in *PySpin*), 42
ChannelStatistics (class in *PySpin.PySpin*), 147
CheckCRC() (*PySpin.Image* method), 46
CheckCRC() (*PySpin.PySpin.IImage* method), 320
CheckCRC() (*PySpin.PySpin.Image* method), 339
ChunkBlackLevel (*PySpin.Camera* property), 12
ChunkBlackLevel (*PySpin.PySpin.Camera* property), 118
ChunkBlackLevelSelector (*PySpin.Camera* property), 12
ChunkBlackLevelSelector (*PySpin.PySpin.Camera* property), 118
ChunkCompressionMode (*PySpin.Camera* property), 12
ChunkCompressionMode (*PySpin.PySpin.Camera* property), 118
ChunkCompressionRatio (*PySpin.Camera* property), 12
ChunkCompressionRatio (*PySpin.PySpin.Camera* property), 118
ChunkCounterSelector (*PySpin.Camera* property), 12
ChunkCounterSelector (*PySpin.PySpin.Camera* property), 118
ChunkCounterValue (*PySpin.Camera* property), 12
ChunkCounterValue (*PySpin.PySpin.Camera* property), 118
ChunkCRC (*PySpin.Camera* property), 12
ChunkCRC (*PySpin.PySpin.Camera* property), 118
ChunkCurrentDatarate (*PySpin.Camera* property), 12
ChunkCurrentDatarate (*PySpin.PySpin.Camera* property), 118
ChunkData (class in *PySpin*), 42
ChunkData (class in *PySpin.PySpin*), 148
ChunkEnable (*PySpin.Camera* property), 12
ChunkEnable (*PySpin.PySpin.Camera* property), 118
ChunkEncoderSelector (*PySpin.Camera* property), 12
ChunkEncoderSelector (*PySpin.PySpin.Camera* property), 118
ChunkEncoderStatus (*PySpin.Camera* property), 12
ChunkEncoderStatus (*PySpin.PySpin.Camera* property), 118
ChunkEncoderValue (*PySpin.Camera* property), 12
ChunkEncoderValue (*PySpin.PySpin.Camera* property), 118
ChunkExposureEndLineStatusAll (*PySpin.Camera* property), 13
ChunkExposureEndLineStatusAll (*PySpin.PySpin.Camera* property), 118
ChunkExposureTime (*PySpin.Camera* property), 13
ChunkExposureTime (*PySpin.PySpin.Camera* property),

- 118
- ChunkExposureTimeSelector (*PySpin.Camera property*), 13
- ChunkExposureTimeSelector (*PySpin.PySpin.Camera property*), 118
- ChunkFrameID (*PySpin.Camera property*), 13
- ChunkFrameID (*PySpin.PySpin.Camera property*), 118
- ChunkGain (*PySpin.Camera property*), 13
- ChunkGain (*PySpin.PySpin.Camera property*), 118
- ChunkGainSelector (*PySpin.Camera property*), 13
- ChunkGainSelector (*PySpin.PySpin.Camera property*), 118
- ChunkHeight (*PySpin.Camera property*), 13
- ChunkHeight (*PySpin.PySpin.Camera property*), 118
- ChunkImage (*PySpin.Camera property*), 13
- ChunkImage (*PySpin.PySpin.Camera property*), 118
- ChunkImageComponent (*PySpin.Camera property*), 13
- ChunkImageComponent (*PySpin.PySpin.Camera property*), 118
- ChunkInferenceBoundingBoxResult (*PySpin.Camera property*), 13
- ChunkInferenceBoundingBoxResult (*PySpin.PySpin.Camera property*), 118
- ChunkInferenceConfidence (*PySpin.Camera property*), 13
- ChunkInferenceConfidence (*PySpin.PySpin.Camera property*), 118
- ChunkInferenceFrameId (*PySpin.Camera property*), 13
- ChunkInferenceFrameId (*PySpin.PySpin.Camera property*), 118
- ChunkInferenceResult (*PySpin.Camera property*), 13
- ChunkInferenceResult (*PySpin.PySpin.Camera property*), 118
- ChunkLinePitch (*PySpin.Camera property*), 13
- ChunkLinePitch (*PySpin.PySpin.Camera property*), 118
- ChunkLineStatusAll (*PySpin.Camera property*), 13
- ChunkLineStatusAll (*PySpin.PySpin.Camera property*), 118
- ChunkModeActive (*PySpin.Camera property*), 13
- ChunkModeActive (*PySpin.PySpin.Camera property*), 118
- ChunkOffsetX (*PySpin.Camera property*), 13
- ChunkOffsetX (*PySpin.PySpin.Camera property*), 118
- ChunkOffsetY (*PySpin.Camera property*), 13
- ChunkOffsetY (*PySpin.PySpin.Camera property*), 118
- ChunkPartSelector (*PySpin.Camera property*), 13
- ChunkPartSelector (*PySpin.PySpin.Camera property*), 119
- ChunkPixelDynamicRangeMax (*PySpin.Camera property*), 13
- ChunkPixelDynamicRangeMax (*PySpin.PySpin.Camera property*), 119
- ChunkPixelDynamicRangeMin (*PySpin.Camera property*), 13
- ChunkPixelDynamicRangeMin (*PySpin.PySpin.Camera property*), 119
- ChunkPixelFormat (*PySpin.Camera property*), 13
- ChunkPixelFormat (*PySpin.PySpin.Camera property*), 119
- ChunkRegionID (*PySpin.Camera property*), 13
- ChunkRegionID (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dAxisMax (*PySpin.Camera property*), 13
- ChunkScan3dAxisMax (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dAxisMin (*PySpin.Camera property*), 13
- ChunkScan3dAxisMin (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateOffset (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateOffset (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateReferenceSelector (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateReferenceSelector (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateReferenceValue (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateReferenceValue (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateScale (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateScale (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateSelector (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateSelector (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateSystem (*PySpin.Camera property*), 13
- ChunkScan3dCoordinateSystem (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateSystemReference (*PySpin.Camera property*), 14
- ChunkScan3dCoordinateSystemReference (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dCoordinateTransformSelector (*PySpin.Camera property*), 14
- ChunkScan3dCoordinateTransformSelector (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dDistanceUnit (*PySpin.Camera property*), 14
- ChunkScan3dDistanceUnit (*PySpin.PySpin.Camera property*), 119
- ChunkScan3dInvalidDataFlag (*PySpin.Camera property*), 14
- ChunkScan3dInvalidDataFlag

- (*PySpin.PySpin.Camera* property), 119
- ChunkScan3dInvalidDataValue (*PySpin.Camera* property), 14
- ChunkScan3dInvalidDataValue (*PySpin.PySpin.Camera* property), 119
- ChunkScan3dOutputMode (*PySpin.Camera* property), 14
- ChunkScan3dOutputMode (*PySpin.PySpin.Camera* property), 119
- ChunkScan3dTransformValue (*PySpin.Camera* property), 14
- ChunkScan3dTransformValue (*PySpin.PySpin.Camera* property), 119
- ChunkScanLineSelector (*PySpin.Camera* property), 14
- ChunkScanLineSelector (*PySpin.PySpin.Camera* property), 119
- ChunkSelector (*PySpin.Camera* property), 14
- ChunkSelector (*PySpin.PySpin.Camera* property), 119
- ChunkSequencerSetActive (*PySpin.Camera* property), 14
- ChunkSequencerSetActive (*PySpin.PySpin.Camera* property), 119
- ChunkSerialData (*PySpin.Camera* property), 14
- ChunkSerialData (*PySpin.PySpin.Camera* property), 119
- ChunkSerialDataLength (*PySpin.Camera* property), 14
- ChunkSerialDataLength (*PySpin.PySpin.Camera* property), 119
- ChunkSerialReceiveOverflow (*PySpin.Camera* property), 14
- ChunkSerialReceiveOverflow (*PySpin.PySpin.Camera* property), 119
- ChunkSourceID (*PySpin.Camera* property), 14
- ChunkSourceID (*PySpin.PySpin.Camera* property), 119
- ChunkStreamChannelID (*PySpin.Camera* property), 14
- ChunkStreamChannelID (*PySpin.PySpin.Camera* property), 119
- ChunkTimerSelector (*PySpin.Camera* property), 14
- ChunkTimerSelector (*PySpin.PySpin.Camera* property), 119
- ChunkTimerValue (*PySpin.Camera* property), 14
- ChunkTimerValue (*PySpin.PySpin.Camera* property), 119
- ChunkTimestamp (*PySpin.Camera* property), 14
- ChunkTimestamp (*PySpin.PySpin.Camera* property), 119
- ChunkTimestampLatchValue (*PySpin.Camera* property), 14
- ChunkTimestampLatchValue (*PySpin.PySpin.Camera* property), 120
- ChunkTransferBlockID (*PySpin.Camera* property), 14
- ChunkTransferBlockID (*PySpin.PySpin.Camera* property), 120
- ChunkTransferQueueCurrentBlockCount (*PySpin.Camera* property), 14
- ChunkTransferQueueCurrentBlockCount (*PySpin.PySpin.Camera* property), 120
- ChunkTransferStreamID (*PySpin.Camera* property), 14
- ChunkTransferStreamID (*PySpin.PySpin.Camera* property), 120
- ChunkWidth (*PySpin.Camera* property), 14
- ChunkWidth (*PySpin.PySpin.Camera* property), 120
- CIntegerPtr (class in *PySpin.PySpin*), 98
- circle (*PySpin.PySpin.InferenceBoundingBox* property), 360
- classId (*PySpin.PySpin.InferenceBoundingBox* property), 360
- ClConfiguration (*PySpin.Camera* property), 14
- ClConfiguration (*PySpin.PySpin.Camera* property), 120
- Clear() (*PySpin.CameraList* method), 40
- Clear() (*PySpin.ImageList* method), 55
- Clear() (*PySpin.InterfaceList* method), 67
- Clear() (*PySpin.PySpin.CameraList* method), 145
- Clear() (*PySpin.PySpin.ICameraList* method), 171
- Clear() (*PySpin.PySpin.IImageList* method), 325
- Clear() (*PySpin.PySpin.IInterfaceList* method), 329
- Clear() (*PySpin.PySpin.ImageList* method), 349
- Clear() (*PySpin.PySpin.InterfaceList* method), 364
- clear() (*PySpin.PySpin.node_vector* method), 409
- clear() (*PySpin.PySpin.value_vector* method), 410
- ClearAllNodes() (*PySpin.PySpin.CNodeMapDynPtr* method), 101
- ClearAllNodes() (*PySpin.PySpin.INodeMapDyn* method), 332
- ClearXMLCache() (*PySpin.PySpin.NodeMap* static method), 378
- Close() (*PySpin.PySpin.SpinVideo* method), 387
- Close() (*PySpin.SpinVideo* method), 69
- ClTimeSlotsCount (*PySpin.Camera* property), 14
- ClTimeSlotsCount (*PySpin.PySpin.Camera* property), 120
- CNodeMapDynPtr (class in *PySpin.PySpin*), 101
- CNodeMapPtr (class in *PySpin.PySpin*), 103
- CNodePtr (class in *PySpin.PySpin*), 104
- ColorSpace (*PySpin.PySpin.CCMSettings* property), 86
- ColorSpaceToString() (*PySpin.ImageUtilityCCM* static method), 60
- ColorSpaceToString() (*PySpin.PySpin.ImageUtilityCCM* static method), 353
- ColorTemperature (*PySpin.PySpin.CCMSettings* property), 86
- ColorTemperatureToString() (*PySpin.ImageUtilityCCM* static method),

- 60
- ColorTemperatureToString()
(*PySpin.PySpin.ImageUtilityCCM static method*), 354
- ColorTransformationEnable (*PySpin.Camera property*), 14
- ColorTransformationEnable
(*PySpin.PySpin.Camera property*), 120
- ColorTransformationSelector (*PySpin.Camera property*), 14
- ColorTransformationSelector
(*PySpin.PySpin.Camera property*), 120
- ColorTransformationValue (*PySpin.Camera property*), 14
- ColorTransformationValue (*PySpin.PySpin.Camera property*), 120
- ColorTransformationValueSelector
(*PySpin.Camera property*), 14
- ColorTransformationValueSelector
(*PySpin.PySpin.Camera property*), 120
- Combine() (*in module PySpin.PySpin*), 151
- CommandNode (*class in PySpin.PySpin*), 152
- compare() (*PySpin.PySpin.gcstring method*), 405
- ComponentActiveCount (*PySpin.Camera property*), 14
- ComponentActiveCount (*PySpin.PySpin.Camera property*), 120
- ComponentDestination (*PySpin.Camera property*), 14
- ComponentDestination (*PySpin.PySpin.Camera property*), 120
- ComponentEnable (*PySpin.Camera property*), 15
- ComponentEnable (*PySpin.PySpin.Camera property*), 120
- ComponentSelector (*PySpin.Camera property*), 15
- ComponentSelector (*PySpin.PySpin.Camera property*), 120
- CompressedFrameDropCount (*PySpin.Camera property*), 15
- CompressedFrameDropCount (*PySpin.PySpin.Camera property*), 120
- compression (*PySpin.PySpin.TIFFOption property*), 395
- compressionLevel (*PySpin.PySpin.PNGOption property*), 383
- CompressionSaturationPriority (*PySpin.Camera property*), 15
- CompressionSaturationPriority
(*PySpin.PySpin.Camera property*), 120
- Compute3DPointFromPixel()
(*PySpin.ImageUtilityStereo static method*), 64
- Compute3DPointFromPixel()
(*PySpin.PySpin.ImageUtilityStereo static method*), 358
- ComputeDistanceBetweenPoints()
(*PySpin.ImageUtilityStereo static method*), 64
- ComputeDistanceBetweenPoints()
(*PySpin.PySpin.ImageUtilityStereo static method*), 358
- ComputeDistanceToPoint()
(*PySpin.ImageUtilityStereo static method*), 64
- ComputeDistanceToPoint()
(*PySpin.PySpin.ImageUtilityStereo static method*), 358
- ComputePointCloud() (*PySpin.ImageUtilityStereo static method*), 64
- ComputePointCloud()
(*PySpin.PySpin.ImageUtilityStereo static method*), 358
- confidence (*PySpin.PySpin.DeviceEventInferenceData property*), 154
- confidence (*PySpin.PySpin.InferenceBoundingBox property*), 360
- Connect() (*PySpin.PySpin.CNodeMapDynPtr method*), 101
- Connect() (*PySpin.PySpin.CNodeMapPtr method*), 103
- Connect() (*PySpin.PySpin.INodeMap method*), 332
- Connect() (*PySpin.PySpin.NodeMap method*), 378
- ControlPacketsReservedBandwidth
(*PySpin.Camera property*), 15
- ControlPacketsReservedBandwidth
(*PySpin.PySpin.Camera property*), 120
- Convert() (*PySpin.ImageProcessor method*), 57
- Convert() (*PySpin.PySpin.IImageProcessor method*), 326
- Convert() (*PySpin.PySpin.ImageProcessor method*), 351
- coordinateOffset (*PySpin.PySpin.StereoCameraParameters property*), 389
- CounterDelay (*PySpin.Camera property*), 15
- CounterDelay (*PySpin.PySpin.Camera property*), 120
- CounterDuration (*PySpin.Camera property*), 15
- CounterDuration (*PySpin.PySpin.Camera property*), 120
- CounterEventActivation (*PySpin.Camera property*), 15
- CounterEventActivation (*PySpin.PySpin.Camera property*), 120
- CounterEventSource (*PySpin.Camera property*), 15
- CounterEventSource (*PySpin.PySpin.Camera property*), 120
- CounterReset (*PySpin.Camera property*), 15
- CounterReset (*PySpin.PySpin.Camera property*), 120
- CounterResetActivation (*PySpin.Camera property*), 15
- CounterResetActivation (*PySpin.PySpin.Camera property*), 120

- CounterResetSource (*PySpin.Camera* property), 15
- CounterResetSource (*PySpin.PySpin.Camera* property), 120
- CounterSelector (*PySpin.Camera* property), 15
- CounterSelector (*PySpin.PySpin.Camera* property), 120
- CounterStatus (*PySpin.Camera* property), 15
- CounterStatus (*PySpin.PySpin.Camera* property), 120
- CounterTriggerActivation (*PySpin.Camera* property), 15
- CounterTriggerActivation (*PySpin.PySpin.Camera* property), 120
- CounterTriggerSource (*PySpin.Camera* property), 15
- CounterTriggerSource (*PySpin.PySpin.Camera* property), 120
- CounterValue (*PySpin.Camera* property), 15
- CounterValue (*PySpin.PySpin.Camera* property), 120
- CounterValueAtReset (*PySpin.Camera* property), 15
- CounterValueAtReset (*PySpin.PySpin.Camera* property), 120
- Create() (*PySpin.Image* static method), 46
- Create() (*PySpin.PySpin.Image* static method), 339
- CreateAolp() (*PySpin.ImageUtilityPolarization* static method), 62
- CreateAolp() (*PySpin.PySpin.ImageUtilityPolarization* static method), 355
- CreateColorCorrected() (*PySpin.ImageUtilityCCM* static method), 60
- CreateColorCorrected() (*PySpin.PySpin.ImageUtilityCCM* static method), 354
- CreateDepthImage() (*PySpin.ImageUtilityStereo* static method), 65
- CreateDepthImage() (*PySpin.PySpin.ImageUtilityStereo* static method), 359
- CreateDolp() (*PySpin.ImageUtilityPolarization* static method), 62
- CreateDolp() (*PySpin.PySpin.ImageUtilityPolarization* static method), 356
- CreateGlareReduced() (*PySpin.ImageUtilityPolarization* static method), 62
- CreateGlareReduced() (*PySpin.PySpin.ImageUtilityPolarization* static method), 356
- CreateHeatmap() (*PySpin.ImageUtilityHeatmap* static method), 60
- CreateHeatmap() (*PySpin.PySpin.ImageUtilityHeatmap* static method), 354
- CreateNormalized() (*PySpin.ImageUtility* static method), 58
- CreateNormalized() (*PySpin.PySpin.ImageUtility* static method), 352
- CreateScaled() (*PySpin.ImageUtility* static method), 59
- CreateScaled() (*PySpin.PySpin.ImageUtility* static method), 353
- CreateStokesS0() (*PySpin.ImageUtilityPolarization* static method), 62
- CreateStokesS0() (*PySpin.PySpin.ImageUtilityPolarization* static method), 356
- CreateStokesS1() (*PySpin.ImageUtilityPolarization* static method), 63
- CreateStokesS1() (*PySpin.PySpin.ImageUtilityPolarization* static method), 357
- CreateStokesS2() (*PySpin.ImageUtilityPolarization* static method), 63
- CreateStokesS2() (*PySpin.PySpin.ImageUtilityPolarization* static method), 357
- CRegisterPtr (class in *PySpin.PySpin*), 106
- crf (*PySpin.PySpin.H264Option* property), 167
- CSelectorPtr (class in *PySpin.PySpin*), 109
- CSelectorSet (class in *PySpin.PySpin*), 110
- CStringPtr (class in *PySpin.PySpin*), 110
- CustomCCMCode (*PySpin.PySpin.CCMSettings* property), 86
- CValuePtr (class in *PySpin.PySpin*), 113
- CxpConnectionSelector (*PySpin.Camera* property), 15
- CxpConnectionSelector (*PySpin.PySpin.Camera* property), 121
- CxpConnectionTestErrorCount (*PySpin.Camera* property), 15
- CxpConnectionTestErrorCount (*PySpin.PySpin.Camera* property), 121
- CxpConnectionTestMode (*PySpin.Camera* property), 15
- CxpConnectionTestMode (*PySpin.PySpin.Camera* property), 121
- CxpConnectionTestPacketCount (*PySpin.Camera* property), 15
- CxpConnectionTestPacketCount (*PySpin.PySpin.Camera* property), 121
- CxpLinkConfiguration (*PySpin.Camera* property), 15
- CxpLinkConfiguration (*PySpin.PySpin.Camera* property), 121
- CxpLinkConfigurationPreferred (*PySpin.Camera* property), 15
- CxpLinkConfigurationPreferred (*PySpin.PySpin.Camera* property), 121
- CxpLinkConfigurationStatus (*PySpin.Camera* property), 15
- CxpLinkConfigurationStatus (*PySpin.PySpin.Camera* property), 121
- CxpPoCxpAuto (*PySpin.Camera* property), 15
- CxpPoCxpAuto (*PySpin.PySpin.Camera* property), 121
- CxpPoCxpStatus (*PySpin.Camera* property), 15
- CxpPoCxpStatus (*PySpin.PySpin.Camera* property), 121

- 121
 CxpPoCxpTripReset (*PySpin.Camera* property), 15
 CxpPoCxpTripReset (*PySpin.PySpin.Camera* property), 121
 CxpPoCxpTurnOff (*PySpin.Camera* property), 15
 CxpPoCxpTurnOff (*PySpin.PySpin.Camera* property), 121
- ## D
- decimationFactor (*PySpin.PySpin.PointCloudParameters* property), 384
 DecimationHorizontal (*PySpin.Camera* property), 15
 DecimationHorizontal (*PySpin.PySpin.Camera* property), 121
 DecimationHorizontalMode (*PySpin.Camera* property), 15
 DecimationHorizontalMode (*PySpin.PySpin.Camera* property), 121
 DecimationSelector (*PySpin.Camera* property), 16
 DecimationSelector (*PySpin.PySpin.Camera* property), 121
 DecimationVertical (*PySpin.Camera* property), 16
 DecimationVertical (*PySpin.PySpin.Camera* property), 121
 DecimationVerticalMode (*PySpin.Camera* property), 16
 DecimationVerticalMode (*PySpin.PySpin.Camera* property), 121
 DeepCopy() (*PySpin.Image* method), 47
 DeepCopy() (*PySpin.PySpin.Image* method), 320
 DeepCopy() (*PySpin.PySpin.Image* method), 341
 DefectCorrectionMode (*PySpin.Camera* property), 16
 DefectCorrectionMode (*PySpin.PySpin.Camera* property), 121
 DefectCorrectStaticEnable (*PySpin.Camera* property), 16
 DefectCorrectStaticEnable (*PySpin.PySpin.Camera* property), 121
 DefectTableApply (*PySpin.Camera* property), 16
 DefectTableApply (*PySpin.PySpin.Camera* property), 121
 DefectTableCoordinateX (*PySpin.Camera* property), 16
 DefectTableCoordinateX (*PySpin.PySpin.Camera* property), 121
 DefectTableCoordinateY (*PySpin.Camera* property), 16
 DefectTableCoordinateY (*PySpin.PySpin.Camera* property), 121
 DefectTableFactoryRestore (*PySpin.Camera* property), 16
 DefectTableFactoryRestore (*PySpin.PySpin.Camera* property), 121
 DefectTableIndex (*PySpin.Camera* property), 16
 DefectTableIndex (*PySpin.PySpin.Camera* property), 121
 DefectTablePixelCount (*PySpin.Camera* property), 16
 DefectTablePixelCount (*PySpin.PySpin.Camera* property), 121
 DefectTableSave (*PySpin.Camera* property), 16
 DefectTableSave (*PySpin.PySpin.Camera* property), 121
 DefectTableSensor (*PySpin.Camera* property), 16
 DefectTableSensor (*PySpin.PySpin.Camera* property), 121
 DeInit() (*PySpin.CameraBase* method), 35
 DeInit() (*PySpin.PySpin.CameraBase* method), 140
 DeInit() (*PySpin.PySpin.ICameraBase* method), 168
 Deinterlacing (*PySpin.Camera* property), 16
 Deinterlacing (*PySpin.PySpin.Camera* property), 121
 DeregisterCallback() (*PySpin.PySpin.CBooleanPtr* method), 83
 DeregisterCallback() (*PySpin.PySpin.CCategoryPtr* method), 86
 DeregisterCallback() (*PySpin.PySpin.CCommandPtr* method), 88
 DeregisterCallback() (*PySpin.PySpin.CEnumEntryPtr* method), 91
 DeregisterCallback() (*PySpin.PySpin.CEnumerationPtr* method), 94
 DeregisterCallback() (*PySpin.PySpin.CIntegerPtr* method), 98
 DeregisterCallback() (*PySpin.PySpin.CNodePtr* method), 104
 DeregisterCallback() (*PySpin.PySpin.CRegisterPtr* method), 106
 DeregisterCallback() (*PySpin.PySpin.CStringPtr* method), 110
 DeregisterCallback() (*PySpin.PySpin.CValuePtr* method), 113
 DeregisterCallback() (*PySpin.PySpin.INode* method), 330
 DeregisterCallback() (*PySpin.PySpin.Node* method), 373
 DeregisterNodeCallback() (*PySpin.PySpin* module), 153
 Destroy() (*PySpin.PySpin.IDestroy* method), 174
 Destroy() (*PySpin.PySpin.NodeMap* method), 378
 DeviceAccessStatus (*PySpin.PySpin.TransportLayerDevice* property), 396
 DeviceAccessStatus (*PySpin.PySpin.TransportLayerInterface* property), 398
 DeviceAccessStatus (*PySpin.TransportLayerDevice* property), 75
 DeviceAccessStatus (*PySpin.TransportLayerInterface*

<i>property</i>), 77	<i>(PySpin.TransportLayerDevice property)</i> , 75
DeviceAddress (<i>PySpin.PySpin.ActionCommandResult property</i>), 81	DeviceEventChannelCount (<i>PySpin.Camera property</i>), 16
DeviceArrivalEventHandler (<i>class in PySpin</i>), 5	DeviceEventChannelCount (<i>PySpin.PySpin.Camera property</i>), 122
DeviceArrivalEventHandler (<i>class in PySpin.PySpin</i>), 153	DeviceEventExposureEndData (<i>class in PySpin.PySpin</i>), 153
DeviceBootloaderVersion (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	DeviceEventHandler (<i>class in PySpin</i>), 5
DeviceBootloaderVersion (<i>PySpin.TransportLayerDevice property</i>), 75	DeviceEventHandler (<i>class in PySpin.PySpin</i>), 153
DeviceCharacterSet (<i>PySpin.Camera property</i>), 16	DeviceEventInferenceData (<i>class in PySpin.PySpin</i>), 153
DeviceCharacterSet (<i>PySpin.PySpin.Camera property</i>), 121	DeviceFamilyName (<i>PySpin.Camera property</i>), 16
DeviceClockFrequency (<i>PySpin.Camera property</i>), 16	DeviceFamilyName (<i>PySpin.PySpin.Camera property</i>), 122
DeviceClockFrequency (<i>PySpin.PySpin.Camera property</i>), 121	DeviceFeaturePersistenceEnd (<i>PySpin.Camera property</i>), 16
DeviceClockSelector (<i>PySpin.Camera property</i>), 16	DeviceFeaturePersistenceEnd (<i>PySpin.PySpin.Camera property</i>), 122
DeviceClockSelector (<i>PySpin.PySpin.Camera property</i>), 121	DeviceFeaturePersistenceStart (<i>PySpin.Camera property</i>), 16
DeviceConnectionSelector (<i>PySpin.Camera property</i>), 16	DeviceFeaturePersistenceStart (<i>PySpin.PySpin.Camera property</i>), 122
DeviceConnectionSelector (<i>PySpin.PySpin.Camera property</i>), 121	DeviceFirmwareVersion (<i>PySpin.Camera property</i>), 16
DeviceConnectionSpeed (<i>PySpin.Camera property</i>), 16	DeviceFirmwareVersion (<i>PySpin.PySpin.Camera property</i>), 122
DeviceConnectionSpeed (<i>PySpin.PySpin.Camera property</i>), 122	DeviceGenCPVersionMajor (<i>PySpin.Camera property</i>), 16
DeviceConnectionStatus (<i>PySpin.Camera property</i>), 16	DeviceGenCPVersionMajor (<i>PySpin.PySpin.Camera property</i>), 122
DeviceConnectionStatus (<i>PySpin.PySpin.Camera property</i>), 122	DeviceGenCPVersionMinor (<i>PySpin.Camera property</i>), 16
DeviceCount (<i>PySpin.PySpin.TransportLayerInterface property</i>), 398	DeviceGenCPVersionMinor (<i>PySpin.PySpin.Camera property</i>), 122
DeviceCount (<i>PySpin.TransportLayerInterface property</i>), 77	DeviceID (<i>PySpin.Camera property</i>), 16
DeviceCurrentSpeed (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	DeviceID (<i>PySpin.PySpin.Camera property</i>), 122
DeviceCurrentSpeed (<i>PySpin.TransportLayerDevice property</i>), 75	DeviceID (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
DeviceDisplayName (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	DeviceID (<i>PySpin.PySpin.TransportLayerInterface property</i>), 398
DeviceDisplayName (<i>PySpin.TransportLayerDevice property</i>), 75	DeviceID (<i>PySpin.TransportLayerDevice property</i>), 75
DeviceDriverVersion (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	DeviceID (<i>PySpin.TransportLayerInterface property</i>), 77
DeviceDriverVersion (<i>PySpin.TransportLayerDevice property</i>), 75	DeviceIndicatorMode (<i>PySpin.Camera property</i>), 16
DeviceEndianessMechanism (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	DeviceIndicatorMode (<i>PySpin.PySpin.Camera property</i>), 122
DeviceEndianessMechanism	DeviceInstanceId (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
	DeviceInstanceId (<i>PySpin.TransportLayerDevice property</i>), 75
	DeviceIsUpdater (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
	DeviceIsUpdater (<i>PySpin.TransportLayerDevice property</i>), 75

<code>DeviceLinkBandwidthReserve</code> (<i>PySpin.Camera property</i>), 16	<code>DeviceManifestSchemaMajorVersion</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkBandwidthReserve</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestSchemaMajorVersion</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkCommandTimeout</code> (<i>PySpin.Camera property</i>), 16	<code>DeviceManifestSchemaMinorVersion</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkCommandTimeout</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestSchemaMinorVersion</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkConnectionCount</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceManifestSecondaryURL</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkConnectionCount</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestSecondaryURL</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkCurrentThroughput</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceManifestXMLMajorVersion</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkCurrentThroughput</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestXMLMajorVersion</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkHeartbeatMode</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceManifestXMLMinorVersion</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkHeartbeatMode</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestXMLMinorVersion</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkHeartbeatTimeout</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceManifestXMLSubMinorVersion</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkHeartbeatTimeout</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManifestXMLSubMinorVersion</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkSelector</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceManufacturerInfo</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkSelector</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceManufacturerInfo</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkSpeed</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceMaxThroughput</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkSpeed</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceMaxThroughput</code> (<i>PySpin.PySpin.Camera property</i>), 122
<code>DeviceLinkSpeed</code> (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	<code>DeviceModelName</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceLinkSpeed</code> (<i>PySpin.TransportLayerDevice property</i>), 75	<code>DeviceModelName</code> (<i>PySpin.PySpin.Camera property</i>), 123
<code>DeviceLinkThroughputLimit</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceModelName</code> (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
<code>DeviceLinkThroughputLimit</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceModelName</code> (<i>PySpin.PySpin.TransportLayerInterface property</i>), 398
<code>DeviceLinkThroughputLimitMode</code> (<i>PySpin.Camera property</i>), 17	<code>DeviceModelName</code> (<i>PySpin.TransportLayerDevice property</i>), 75
<code>DeviceLinkThroughputLimitMode</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DeviceModelName</code> (<i>PySpin.TransportLayerInterface property</i>), 77
<code>DeviceLocation</code> (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396	<code>DeviceMulticastMonitorMode</code> (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
<code>DeviceLocation</code> (<i>PySpin.TransportLayerDevice property</i>), 75	<code>DeviceMulticastMonitorMode</code> (<i>PySpin.TransportLayerDevice property</i>), 75
<code>DeviceManifestEntrySelector</code> (<i>PySpin.Camera property</i>), 17	<code>DevicePortId</code> (<i>PySpin.PySpin.TransportLayerDevice property</i>), 396
<code>DeviceManifestEntrySelector</code> (<i>PySpin.PySpin.Camera property</i>), 122	<code>DevicePortId</code> (<i>PySpin.TransportLayerDevice property</i>), 75
<code>DeviceManifestPrimaryURL</code> (<i>PySpin.Camera property</i>), 17	<code>DevicePowerSupplySelector</code> (<i>PySpin.Camera property</i>), 17
<code>DeviceManifestPrimaryURL</code> (<i>PySpin.PySpin.Camera property</i>), 122	

- DevicePowerSupplySelector (*PySpin.PySpin.Camera* property), 123
- DeviceRegistersCheck (*PySpin.Camera* property), 17
- DeviceRegistersCheck (*PySpin.PySpin.Camera* property), 123
- DeviceRegistersEndianness (*PySpin.Camera* property), 17
- DeviceRegistersEndianness (*PySpin.PySpin.Camera* property), 123
- DeviceRegistersStreamingEnd (*PySpin.Camera* property), 17
- DeviceRegistersStreamingEnd (*PySpin.PySpin.Camera* property), 123
- DeviceRegistersStreamingStart (*PySpin.Camera* property), 17
- DeviceRegistersStreamingStart (*PySpin.PySpin.Camera* property), 123
- DeviceRegistersValid (*PySpin.Camera* property), 17
- DeviceRegistersValid (*PySpin.PySpin.Camera* property), 123
- DeviceRemovalEventHandler (class in *PySpin*), 6
- DeviceRemovalEventHandler (class in *PySpin.PySpin*), 154
- DeviceReset (*PySpin.Camera* property), 17
- DeviceReset (*PySpin.PySpin.Camera* property), 123
- DeviceReset (*PySpin.PySpin.TransportLayerDevice* property), 396
- DeviceReset (*PySpin.TransportLayerDevice* property), 75
- DeviceScanType (*PySpin.Camera* property), 17
- DeviceScanType (*PySpin.PySpin.Camera* property), 123
- DeviceSelector (*PySpin.PySpin.TransportLayerInterface* property), 398
- DeviceSelector (*PySpin.TransportLayerInterface* property), 77
- DeviceSensorChroma (*PySpin.Camera* property), 17
- DeviceSensorChroma (*PySpin.PySpin.Camera* property), 123
- DeviceSerialNumber (*PySpin.Camera* property), 18
- DeviceSerialNumber (*PySpin.PySpin.Camera* property), 123
- DeviceSerialNumber (*PySpin.PySpin.TransportLayerDevice* property), 396
- DeviceSerialNumber (*PySpin.PySpin.TransportLayerInterface* property), 398
- DeviceSerialNumber (*PySpin.TransportLayerDevice* property), 75
- DeviceSerialNumber (*PySpin.TransportLayerInterface* property), 77
- DeviceSerialPortBaudRate (*PySpin.Camera* property), 18
- DeviceSerialPortBaudRate (*PySpin.PySpin.Camera* property), 123
- DeviceSerialPortSelector (*PySpin.Camera* property), 18
- DeviceSerialPortSelector (*PySpin.PySpin.Camera* property), 123
- DeviceSFNCVersionMajor (*PySpin.Camera* property), 17
- DeviceSFNCVersionMajor (*PySpin.PySpin.Camera* property), 123
- DeviceSFNCVersionMinor (*PySpin.Camera* property), 17
- DeviceSFNCVersionMinor (*PySpin.PySpin.Camera* property), 123
- DeviceSFNCVersionSubMinor (*PySpin.Camera* property), 17
- DeviceSFNCVersionSubMinor (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelCount (*PySpin.Camera* property), 18
- DeviceStreamChannelCount (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelEndianness (*PySpin.Camera* property), 18
- DeviceStreamChannelEndianness (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelLink (*PySpin.Camera* property), 18
- DeviceStreamChannelLink (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelPacketSize (*PySpin.Camera* property), 18
- DeviceStreamChannelPacketSize (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelSelector (*PySpin.Camera* property), 18
- DeviceStreamChannelSelector (*PySpin.PySpin.Camera* property), 123
- DeviceStreamChannelType (*PySpin.Camera* property), 18
- DeviceStreamChannelType (*PySpin.PySpin.Camera* property), 123
- DeviceTapGeometry (*PySpin.Camera* property), 18
- DeviceTapGeometry (*PySpin.PySpin.Camera* property), 123
- DeviceTemperature (*PySpin.Camera* property), 18
- DeviceTemperature (*PySpin.PySpin.Camera* property), 123
- DeviceTemperatureSelector (*PySpin.Camera* property), 18
- DeviceTemperatureSelector (*PySpin.PySpin.Camera* property), 123
- DeviceTLType (*PySpin.Camera* property), 18
- DeviceTLType (*PySpin.PySpin.Camera* property), 123
- DeviceTLVersionMajor (*PySpin.Camera* property), 18
- DeviceTLVersionMajor (*PySpin.PySpin.Camera* property), 123

- erty), 123
- DeviceTLVersionMinor (PySpin.Camera property), 18
- DeviceTLVersionMinor (PySpin.PySpin.Camera property), 123
- DeviceTLVersionSubMinor (PySpin.Camera property), 18
- DeviceTLVersionSubMinor (PySpin.PySpin.Camera property), 123
- DeviceType (PySpin.Camera property), 18
- DeviceType (PySpin.PySpin.Camera property), 123
- DeviceType (PySpin.PySpin.TransportLayerDevice property), 396
- DeviceType (PySpin.TransportLayerDevice property), 76
- DeviceU3VProtocol (PySpin.PySpin.TransportLayerDevice property), 396
- DeviceU3VProtocol (PySpin.TransportLayerDevice property), 76
- DeviceUnlock (PySpin.PySpin.TransportLayerInterface property), 398
- DeviceUnlock (PySpin.TransportLayerInterface property), 77
- DeviceUpdateList (PySpin.PySpin.TransportLayerInterface property), 398
- DeviceUpdateList (PySpin.TransportLayerInterface property), 77
- DeviceUptime (PySpin.Camera property), 18
- DeviceUptime (PySpin.PySpin.Camera property), 123
- DeviceUserID (PySpin.Camera property), 18
- DeviceUserID (PySpin.PySpin.Camera property), 124
- DeviceUserID (PySpin.PySpin.TransportLayerDevice property), 396
- DeviceUserID (PySpin.TransportLayerDevice property), 76
- DeviceVendorName (PySpin.Camera property), 18
- DeviceVendorName (PySpin.PySpin.Camera property), 124
- DeviceVendorName (PySpin.PySpin.TransportLayerDevice property), 396
- DeviceVendorName (PySpin.PySpin.TransportLayerInterface property), 398
- DeviceVendorName (PySpin.TransportLayerDevice property), 76
- DeviceVendorName (PySpin.TransportLayerInterface property), 77
- DeviceVersion (PySpin.Camera property), 18
- DeviceVersion (PySpin.PySpin.Camera property), 124
- DeviceVersion (PySpin.PySpin.TransportLayerDevice property), 397
- DeviceVersion (PySpin.TransportLayerDevice property), 76
- DiscoverMaxPacketSize() (PySpin.CameraBase method), 35
- DiscoverMaxPacketSize() (PySpin.PySpin.CameraBase method), 141
- DiscoverMaxPacketSize() (PySpin.PySpin.ICameraBase method), 168
- disparityScaleFactor (PySpin.PySpin.StereoCameraParameters property), 389
- DoesEnvironmentVariableExist() (in module PySpin.PySpin), 154
- double_autovector_t (class in PySpin.PySpin), 404
- ## E
- EAccessModeClass (class in PySpin.PySpin), 154
- EatComments() (in module PySpin.PySpin), 160
- ECachingModeClass (class in PySpin.PySpin), 155
- EDisplayNotationClass (class in PySpin.PySpin), 155
- EEndianessClass (class in PySpin.PySpin), 155
- EGenApiSchemaVersionClass (class in PySpin.PySpin), 156
- EInputDirectionClass (class in PySpin.PySpin), 156
- empty() (PySpin.PySpin.gcstring method), 406
- empty() (PySpin.PySpin.node_vector method), 409
- empty() (PySpin.PySpin.value_vector method), 410
- ENamespaceClass (class in PySpin.PySpin), 157
- EncoderDivider (PySpin.Camera property), 18
- EncoderDivider (PySpin.PySpin.Camera property), 124
- EncoderMode (PySpin.Camera property), 18
- EncoderMode (PySpin.PySpin.Camera property), 124
- EncoderOutputMode (PySpin.Camera property), 18
- EncoderOutputMode (PySpin.PySpin.Camera property), 124
- EncoderReset (PySpin.Camera property), 18
- EncoderReset (PySpin.PySpin.Camera property), 124
- EncoderResetActivation (PySpin.Camera property), 18
- EncoderResetActivation (PySpin.PySpin.Camera property), 124
- EncoderResetSource (PySpin.Camera property), 18
- EncoderResetSource (PySpin.PySpin.Camera property), 124
- EncoderSelector (PySpin.Camera property), 18
- EncoderSelector (PySpin.PySpin.Camera property), 124
- EncoderSourceA (PySpin.Camera property), 18
- EncoderSourceA (PySpin.PySpin.Camera property), 124
- EncoderSourceB (PySpin.Camera property), 18
- EncoderSourceB (PySpin.PySpin.Camera property), 124
- EncoderStatus (PySpin.Camera property), 18
- EncoderStatus (PySpin.PySpin.Camera property), 124
- EncoderTimeout (PySpin.Camera property), 19
- EncoderTimeout (PySpin.PySpin.Camera property), 124

EncoderValue (PySpin.Camera property), 19
EncoderValue (PySpin.PySpin.Camera property), 124
EncoderValueAtReset (PySpin.Camera property), 19
EncoderValueAtReset (PySpin.PySpin.Camera property), 124
EncryptColorCorrectionMatrix()
(PySpin.ImageUtilityCCM static method), 60
EncryptColorCorrectionMatrix()
(PySpin.PySpin.ImageUtilityCCM static method), 354
end() (PySpin.PySpin.node_vector method), 409
end() (PySpin.PySpin.value_vector method), 410
EndAcquisition() (PySpin.CameraBase method), 35
EndAcquisition() (PySpin.PySpin.CameraBase method), 141
EndAcquisition() (PySpin.PySpin.ICameraBase method), 168
EnumEntryNode (class in PySpin.PySpin), 160
EnumerateGen2Cameras
(PySpin.PySpin.TransportLayerSystem property), 401
EnumerateGEVInterfaces
(PySpin.PySpin.TransportLayerSystem property), 401
EnumerateUSBInterfaces
(PySpin.PySpin.TransportLayerSystem property), 401
EnumerationCount (PySpin.Camera property), 19
EnumerationCount (PySpin.PySpin.Camera property), 124
EnumNode (class in PySpin.PySpin), 161
erase() (PySpin.PySpin.node_vector method), 409
erase() (PySpin.PySpin.value_vector method), 410
ERepresentationClass (class in PySpin.PySpin), 157
errorCode (PySpin.SpinnakerException attribute), 69
ESignClass (class in PySpin.PySpin), 158
ESlopeClass (class in PySpin.PySpin), 158
EStandardNameSpaceClass (class in PySpin.PySpin), 159
EventAcquisitionEnd (PySpin.Camera property), 19
EventAcquisitionEnd (PySpin.PySpin.Camera property), 124
EventAcquisitionEndFrameID (PySpin.Camera property), 19
EventAcquisitionEndFrameID
(PySpin.PySpin.Camera property), 124
EventAcquisitionEndTimestamp (PySpin.Camera property), 19
EventAcquisitionEndTimestamp
(PySpin.PySpin.Camera property), 124
EventAcquisitionError (PySpin.Camera property), 19
EventAcquisitionError (PySpin.PySpin.Camera property), 124
EventAcquisitionErrorFrameID (PySpin.Camera property), 19
EventAcquisitionErrorFrameID
(PySpin.PySpin.Camera property), 124
EventAcquisitionErrorTimestamp (PySpin.Camera property), 19
EventAcquisitionErrorTimestamp
(PySpin.PySpin.Camera property), 124
EventAcquisitionStart (PySpin.Camera property), 19
EventAcquisitionStart (PySpin.PySpin.Camera property), 124
EventAcquisitionStartFrameID (PySpin.Camera property), 19
EventAcquisitionStartFrameID
(PySpin.PySpin.Camera property), 124
EventAcquisitionStartTimestamp (PySpin.Camera property), 19
EventAcquisitionStartTimestamp
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferEnd (PySpin.Camera property), 19
EventAcquisitionTransferEnd
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferEndFrameID
(PySpin.Camera property), 19
EventAcquisitionTransferEndFrameID
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferEndTimestamp
(PySpin.Camera property), 19
EventAcquisitionTransferEndTimestamp
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferStart (PySpin.Camera property), 19
EventAcquisitionTransferStart
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferStartFrameID
(PySpin.Camera property), 19
EventAcquisitionTransferStartFrameID
(PySpin.PySpin.Camera property), 124
EventAcquisitionTransferStartTimestamp
(PySpin.Camera property), 19
EventAcquisitionTransferStartTimestamp
(PySpin.PySpin.Camera property), 125
EventAcquisitionTrigger (PySpin.Camera property), 19
EventAcquisitionTrigger (PySpin.PySpin.Camera property), 125
EventAcquisitionTriggerFrameID (PySpin.Camera property), 19
EventAcquisitionTriggerFrameID
(PySpin.PySpin.Camera property), 125
EventAcquisitionTriggerTimestamp

<i>(PySpin.Camera property)</i> , 19	<i>EventCounter1StartTimestamp</i> <i>(PySpin.Camera property)</i> , 20
<i>EventAcquisitionTriggerTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventCounter1StartTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventActionLate</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0Restarted</i> <i>(PySpin.Camera property)</i> , 20
<i>EventActionLate</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0Restarted</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventActionLateFrameID</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0RestartedFrameID</i> <i>(PySpin.Camera property)</i> , 20
<i>EventActionLateFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0RestartedFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventActionLateTimestamp</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0RestartedTimestamp</i> <i>(PySpin.Camera property)</i> , 20
<i>EventActionLateTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0RestartedTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0End</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0Stopped</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0End</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0Stopped</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0EndFrameID</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0StoppedFrameID</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0EndFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0StoppedFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0EndTimestamp</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder0StoppedTimestamp</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0EndTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder0StoppedTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0Start</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder1Restarted</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0Start</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1Restarted</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0StartFrameID</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder1RestartedFrameID</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0StartFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1RestartedFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter0StartTimestamp</i> <i>(PySpin.Camera property)</i> , 19	<i>EventEncoder1RestartedTimestamp</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter0StartTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1RestartedTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter1End</i> <i>(PySpin.Camera property)</i> , 20	<i>EventEncoder1Stopped</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter1End</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1Stopped</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter1EndFrameID</i> <i>(PySpin.Camera property)</i> , 20	<i>EventEncoder1StoppedFrameID</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter1EndFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1StoppedFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter1EndTimestamp</i> <i>(PySpin.Camera property)</i> , 20	<i>EventEncoder1StoppedTimestamp</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter1EndTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventEncoder1StoppedTimestamp</i> <i>(PySpin.PySpin.Camera property)</i> , 125
<i>EventCounter1Start</i> <i>(PySpin.Camera property)</i> , 20	<i>EventError</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter1Start</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventError</i> <i>(PySpin.PySpin.Camera property)</i> , 126
<i>EventCounter1StartFrameID</i> <i>(PySpin.Camera property)</i> , 20	<i>EventErrorCode</i> <i>(PySpin.Camera property)</i> , 20
<i>EventCounter1StartFrameID</i> <i>(PySpin.PySpin.Camera property)</i> , 125	<i>EventErrorCode</i> <i>(PySpin.PySpin.Camera property)</i> , 126

- 126
- `EventErrorFrameID` (*PySpin.Camera* property), 20
- `EventErrorFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventErrorTimestamp` (*PySpin.Camera* property), 20
- `EventErrorTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventExposureEnd` (*PySpin.Camera* property), 20
- `EventExposureEnd` (*PySpin.PySpin.Camera* property), 126
- `EventExposureEndFrameID` (*PySpin.Camera* property), 20
- `EventExposureEndFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventExposureEndTimestamp` (*PySpin.Camera* property), 20
- `EventExposureEndTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventExposureStart` (*PySpin.Camera* property), 20
- `EventExposureStart` (*PySpin.PySpin.Camera* property), 126
- `EventExposureStartFrameID` (*PySpin.Camera* property), 20
- `EventExposureStartFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventExposureStartTimestamp` (*PySpin.Camera* property), 20
- `EventExposureStartTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstEnd` (*PySpin.Camera* property), 20
- `EventFrameBurstEnd` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstEndFrameID` (*PySpin.Camera* property), 20
- `EventFrameBurstEndFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstEndTimestamp` (*PySpin.Camera* property), 20
- `EventFrameBurstEndTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstStart` (*PySpin.Camera* property), 21
- `EventFrameBurstStart` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstStartFrameID` (*PySpin.Camera* property), 21
- `EventFrameBurstStartFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameBurstStartTimestamp` (*PySpin.Camera* property), 21
- `EventFrameBurstStartTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameEnd` (*PySpin.Camera* property), 21
- `EventFrameEnd` (*PySpin.PySpin.Camera* property), 126
- `EventFrameEndFrameID` (*PySpin.Camera* property), 21
- `EventFrameEndFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameEndTimestamp` (*PySpin.Camera* property), 21
- `EventFrameEndTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameStart` (*PySpin.Camera* property), 21
- `EventFrameStart` (*PySpin.PySpin.Camera* property), 126
- `EventFrameStartFrameID` (*PySpin.Camera* property), 21
- `EventFrameStartFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameStartTimestamp` (*PySpin.Camera* property), 21
- `EventFrameStartTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferEnd` (*PySpin.Camera* property), 21
- `EventFrameTransferEnd` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferEndFrameID` (*PySpin.Camera* property), 21
- `EventFrameTransferEndFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferEndTimestamp` (*PySpin.Camera* property), 21
- `EventFrameTransferEndTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferStart` (*PySpin.Camera* property), 21
- `EventFrameTransferStart` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferStartFrameID` (*PySpin.Camera* property), 21
- `EventFrameTransferStartFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTransferStartTimestamp` (*PySpin.Camera* property), 21
- `EventFrameTransferStartTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTrigger` (*PySpin.Camera* property), 21
- `EventFrameTrigger` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTriggerFrameID` (*PySpin.Camera* property), 21
- `EventFrameTriggerFrameID` (*PySpin.PySpin.Camera* property), 126
- `EventFrameTriggerTimestamp` (*PySpin.Camera* property), 21
- `EventFrameTriggerTimestamp` (*PySpin.PySpin.Camera* property), 126
- `EventHandler` (class in *PySpin*), 6
- `EventHandler` (class in *PySpin.PySpin*), 163

- [EventLine0AnyEdge \(PySpin.Camera property\), 21](#)
[EventLine0AnyEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0AnyEdgeFrameID \(PySpin.Camera property\), 21](#)
[EventLine0AnyEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0AnyEdgeTimestamp \(PySpin.Camera property\), 21](#)
[EventLine0AnyEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0FallingEdge \(PySpin.Camera property\), 21](#)
[EventLine0FallingEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0FallingEdgeFrameID \(PySpin.Camera property\), 21](#)
[EventLine0FallingEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0FallingEdgeTimestamp \(PySpin.Camera property\), 21](#)
[EventLine0FallingEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0RisingEdge \(PySpin.Camera property\), 21](#)
[EventLine0RisingEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0RisingEdgeFrameID \(PySpin.Camera property\), 21](#)
[EventLine0RisingEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine0RisingEdgeTimestamp \(PySpin.Camera property\), 21](#)
[EventLine0RisingEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1AnyEdge \(PySpin.Camera property\), 21](#)
[EventLine1AnyEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1AnyEdgeFrameID \(PySpin.Camera property\), 21](#)
[EventLine1AnyEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1AnyEdgeTimestamp \(PySpin.Camera property\), 21](#)
[EventLine1AnyEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1FallingEdge \(PySpin.Camera property\), 21](#)
[EventLine1FallingEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1FallingEdgeFrameID \(PySpin.Camera property\), 22](#)
[EventLine1FallingEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1FallingEdgeTimestamp \(PySpin.Camera property\), 22](#)
[EventLine1FallingEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1RisingEdge \(PySpin.Camera property\), 22](#)
[EventLine1RisingEdge \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1RisingEdgeFrameID \(PySpin.Camera property\), 22](#)
[EventLine1RisingEdgeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLine1RisingEdgeTimestamp \(PySpin.Camera property\), 22](#)
[EventLine1RisingEdgeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkSpeedChange \(PySpin.Camera property\), 22](#)
[EventLinkSpeedChange \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkSpeedChangeFrameID \(PySpin.Camera property\), 22](#)
[EventLinkSpeedChangeFrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkSpeedChangeTimestamp \(PySpin.Camera property\), 22](#)
[EventLinkSpeedChangeTimestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger0 \(PySpin.Camera property\), 22](#)
[EventLinkTrigger0 \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger0FrameID \(PySpin.Camera property\), 22](#)
[EventLinkTrigger0FrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger0Timestamp \(PySpin.Camera property\), 22](#)
[EventLinkTrigger0Timestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger1 \(PySpin.Camera property\), 22](#)
[EventLinkTrigger1 \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger1FrameID \(PySpin.Camera property\), 22](#)
[EventLinkTrigger1FrameID \(PySpin.PySpin.Camera property\), 127](#)
[EventLinkTrigger1Timestamp \(PySpin.Camera property\), 22](#)
[EventLinkTrigger1Timestamp \(PySpin.PySpin.Camera property\), 127](#)
[EventNotification \(PySpin.Camera property\), 22](#)
[EventNotification \(PySpin.PySpin.Camera property\), 127](#)
[EventSelector \(PySpin.Camera property\), 22](#)
[EventSelector \(PySpin.PySpin.Camera property\), 127](#)
[EventSequencerSetChange \(PySpin.Camera property\), 22](#)

`EventSequencerSetChange` (*PySpin.PySpin.Camera property*), 127

`EventSequencerSetChangeFrameID` (*PySpin.Camera property*), 22

`EventSequencerSetChangeFrameID` (*PySpin.PySpin.Camera property*), 127

`EventSequencerSetChangeTimestamp` (*PySpin.Camera property*), 22

`EventSequencerSetChangeTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventSerialData` (*PySpin.Camera property*), 22

`EventSerialData` (*PySpin.PySpin.Camera property*), 128

`EventSerialDataLength` (*PySpin.Camera property*), 22

`EventSerialDataLength` (*PySpin.PySpin.Camera property*), 128

`EventSerialPortReceive` (*PySpin.Camera property*), 22

`EventSerialPortReceive` (*PySpin.PySpin.Camera property*), 128

`EventSerialPortReceiveTimestamp` (*PySpin.Camera property*), 22

`EventSerialPortReceiveTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventSerialReceiveOverflow` (*PySpin.Camera property*), 22

`EventSerialReceiveOverflow` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockEnd` (*PySpin.Camera property*), 22

`EventStream0TransferBlockEnd` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockEndFrameID` (*PySpin.Camera property*), 22

`EventStream0TransferBlockEndFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockEndTimestamp` (*PySpin.Camera property*), 22

`EventStream0TransferBlockEndTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockStart` (*PySpin.Camera property*), 22

`EventStream0TransferBlockStart` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockStartFrameID` (*PySpin.Camera property*), 22

`EventStream0TransferBlockStartFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockStartTimestamp` (*PySpin.Camera property*), 22

`EventStream0TransferBlockStartTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockTrigger` (*PySpin.Camera property*), 22

`EventStream0TransferBlockTrigger` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockTriggerFrameID` (*PySpin.Camera property*), 23

`EventStream0TransferBlockTriggerFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBlockTriggerTimestamp` (*PySpin.Camera property*), 23

`EventStream0TransferBlockTriggerTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstEnd` (*PySpin.Camera property*), 23

`EventStream0TransferBurstEnd` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstEndFrameID` (*PySpin.Camera property*), 23

`EventStream0TransferBurstEndFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstEndTimestamp` (*PySpin.Camera property*), 23

`EventStream0TransferBurstEndTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstStart` (*PySpin.Camera property*), 23

`EventStream0TransferBurstStart` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstStartFrameID` (*PySpin.Camera property*), 23

`EventStream0TransferBurstStartFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferBurstStartTimestamp` (*PySpin.Camera property*), 23

`EventStream0TransferBurstStartTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferEnd` (*PySpin.Camera property*), 23

`EventStream0TransferEnd` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferEndFrameID` (*PySpin.Camera property*), 23

`EventStream0TransferEndFrameID` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferEndTimestamp` (*PySpin.Camera property*), 23

`EventStream0TransferEndTimestamp` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferOverflow` (*PySpin.Camera property*), 23

`EventStream0TransferOverflow` (*PySpin.PySpin.Camera property*), 128

`EventStream0TransferOverflowFrameID` (*PySpin.Camera property*), 23

`EventStream0TransferOverflowFrameID` (*PySpin.PySpin.Camera property*), 128

- (PySpin.PySpin.Camera property)*, 128
- EventStream@TransferOverflowTimestamp *(PySpin.Camera property)*, 23
- EventStream@TransferOverflowTimestamp *(PySpin.PySpin.Camera property)*, 128
- EventStream@TransferPause *(PySpin.Camera property)*, 23
- EventStream@TransferPause *(PySpin.PySpin.Camera property)*, 128
- EventStream@TransferPauseFrameID *(PySpin.Camera property)*, 23
- EventStream@TransferPauseFrameID *(PySpin.PySpin.Camera property)*, 128
- EventStream@TransferPauseTimestamp *(PySpin.Camera property)*, 23
- EventStream@TransferPauseTimestamp *(PySpin.PySpin.Camera property)*, 128
- EventStream@TransferResume *(PySpin.Camera property)*, 23
- EventStream@TransferResume *(PySpin.PySpin.Camera property)*, 128
- EventStream@TransferResumeFrameID *(PySpin.Camera property)*, 23
- EventStream@TransferResumeFrameID *(PySpin.PySpin.Camera property)*, 129
- EventStream@TransferResumeTimestamp *(PySpin.Camera property)*, 23
- EventStream@TransferResumeTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventStream@TransferStart *(PySpin.Camera property)*, 23
- EventStream@TransferStart *(PySpin.PySpin.Camera property)*, 129
- EventStream@TransferStartFrameID *(PySpin.Camera property)*, 23
- EventStream@TransferStartFrameID *(PySpin.PySpin.Camera property)*, 129
- EventStream@TransferStartTimestamp *(PySpin.Camera property)*, 23
- EventStream@TransferStartTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventTest *(PySpin.Camera property)*, 23
- EventTest *(PySpin.PySpin.Camera property)*, 129
- EventTestTimestamp *(PySpin.Camera property)*, 23
- EventTestTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventTimer@End *(PySpin.Camera property)*, 23
- EventTimer@End *(PySpin.PySpin.Camera property)*, 129
- EventTimer@EndFrameID *(PySpin.Camera property)*, 23
- EventTimer@EndFrameID *(PySpin.PySpin.Camera property)*, 129
- EventTimer@EndTimestamp *(PySpin.Camera property)*, 23
- EventTimer@EndTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventTimer@Start *(PySpin.Camera property)*, 23
- EventTimer@Start *(PySpin.PySpin.Camera property)*, 129
- EventTimer@StartFrameID *(PySpin.Camera property)*, 23
- EventTimer@StartFrameID *(PySpin.PySpin.Camera property)*, 129
- EventTimer@StartTimestamp *(PySpin.Camera property)*, 23
- EventTimer@StartTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventTimer1End *(PySpin.Camera property)*, 24
- EventTimer1End *(PySpin.PySpin.Camera property)*, 129
- EventTimer1EndFrameID *(PySpin.Camera property)*, 24
- EventTimer1EndFrameID *(PySpin.PySpin.Camera property)*, 129
- EventTimer1EndTimestamp *(PySpin.Camera property)*, 24
- EventTimer1EndTimestamp *(PySpin.PySpin.Camera property)*, 129
- EventTimer1Start *(PySpin.Camera property)*, 24
- EventTimer1Start *(PySpin.PySpin.Camera property)*, 129
- EventTimer1StartFrameID *(PySpin.Camera property)*, 24
- EventTimer1StartFrameID *(PySpin.PySpin.Camera property)*, 129
- EventTimer1StartTimestamp *(PySpin.Camera property)*, 24
- EventTimer1StartTimestamp *(PySpin.PySpin.Camera property)*, 129
- EVisibilityClass (class in *PySpin.PySpin*), 159
- Execute() *(PySpin.PySpin.CCommandPtr method)*, 88
- Execute() *(PySpin.PySpin.CommandNode method)*, 152
- Execute() *(PySpin.PySpin.ICommand method)*, 173
- ExposureActiveMode *(PySpin.Camera property)*, 24
- ExposureActiveMode *(PySpin.PySpin.Camera property)*, 129
- ExposureAuto *(PySpin.Camera property)*, 24
- ExposureAuto *(PySpin.PySpin.Camera property)*, 129
- ExposureMode *(PySpin.Camera property)*, 24
- ExposureMode *(PySpin.PySpin.Camera property)*, 129
- ExposureTime *(PySpin.Camera property)*, 24
- ExposureTime *(PySpin.PySpin.Camera property)*, 129
- ExposureTimeMode *(PySpin.Camera property)*, 24
- ExposureTimeMode *(PySpin.PySpin.Camera property)*, 129
- ExposureTimeSelector *(PySpin.Camera property)*, 24

- ExposureTimeSelector (*PySpin.PySpin.Camera property*), 129
- ExternalVoltageEnable (*PySpin.Camera property*), 24
- ExternalVoltageEnable (*PySpin.PySpin.Camera property*), 129
- ExternalVoltageSelector (*PySpin.Camera property*), 24
- ExternalVoltageSelector (*PySpin.PySpin.Camera property*), 129
- ExternalVoltageValue (*PySpin.Camera property*), 24
- ExternalVoltageValue (*PySpin.PySpin.Camera property*), 129
- ExtractIndependentSubtree()
(*PySpin.PySpin.CNodeMapDynPtr method*), 101
- ExtractIndependentSubtree()
(*PySpin.PySpin.INodeMapDyn method*), 332
- ExtractPolarQuadrant()
(*PySpin.ImageUtilityPolarization static method*), 63
- ExtractPolarQuadrant()
(*PySpin.PySpin.ImageUtilityPolarization static method*), 357
- EYesNoClass (*class in PySpin.PySpin*), 159
- ## F
- FactoryReset (*PySpin.Camera property*), 24
- FactoryReset (*PySpin.PySpin.Camera property*), 129
- FfcEnable (*PySpin.Camera property*), 24
- FfcEnable (*PySpin.PySpin.Camera property*), 129
- FfcMode (*PySpin.Camera property*), 24
- FfcMode (*PySpin.PySpin.Camera property*), 129
- FfcUserGain (*PySpin.Camera property*), 24
- FfcUserGain (*PySpin.PySpin.Camera property*), 130
- FfcUserOffset (*PySpin.Camera property*), 24
- FfcUserOffset (*PySpin.PySpin.Camera property*), 130
- FfcUserTableReset (*PySpin.Camera property*), 24
- FfcUserTableReset (*PySpin.PySpin.Camera property*), 130
- FfcUserTableSave (*PySpin.Camera property*), 24
- FfcUserTableSave (*PySpin.PySpin.Camera property*), 130
- FfcUserTableXCoordinate (*PySpin.Camera property*), 24
- FfcUserTableXCoordinate (*PySpin.PySpin.Camera property*), 130
- FileAccessBuffer (*PySpin.Camera property*), 24
- FileAccessBuffer (*PySpin.PySpin.Camera property*), 130
- FileAccessLength (*PySpin.Camera property*), 24
- FileAccessLength (*PySpin.PySpin.Camera property*), 130
- FileAccessOffset (*PySpin.Camera property*), 24
- FileAccessOffset (*PySpin.PySpin.Camera property*), 130
- FileOpenMode (*PySpin.Camera property*), 24
- FileOpenMode (*PySpin.PySpin.Camera property*), 130
- FileOperationExecute (*PySpin.Camera property*), 24
- FileOperationExecute (*PySpin.PySpin.Camera property*), 130
- FileOperationResult (*PySpin.Camera property*), 24
- FileOperationResult (*PySpin.PySpin.Camera property*), 130
- FileOperationSelector (*PySpin.Camera property*), 24
- FileOperationSelector (*PySpin.PySpin.Camera property*), 130
- FileOperationStatus (*PySpin.Camera property*), 24
- FileOperationStatus (*PySpin.PySpin.Camera property*), 130
- FileSelector (*PySpin.Camera property*), 25
- FileSelector (*PySpin.PySpin.Camera property*), 130
- FileSize (*PySpin.Camera property*), 25
- FileSize (*PySpin.PySpin.Camera property*), 130
- FilterSpeckles() (*PySpin.ImageUtilityStereo static method*), 65
- FilterSpeckles() (*PySpin.PySpin.ImageUtilityStereo static method*), 359
- FilterSpecklesFromImage()
(*PySpin.ImageUtilityStereo static method*), 66
- FilterSpecklesFromImage()
(*PySpin.PySpin.ImageUtilityStereo static method*), 359
- find() (*PySpin.PySpin.gcstring method*), 406
- find_first_not_of() (*PySpin.PySpin.gcstring method*), 407
- find_first_of() (*PySpin.PySpin.gcstring method*), 407
- FLIRFilterDriverStatus
(*PySpin.PySpin.TransportLayerInterface property*), 398
- FLIRFilterDriverStatus
(*PySpin.TransportLayerInterface property*), 77
- FloatNode (*class in PySpin.PySpin*), 163
- FloatRegNode (*class in PySpin.PySpin*), 165
- focalLength (*PySpin.PySpin.StereoCameraParameters property*), 389
- ForceIP() (*PySpin.CameraBase method*), 36
- ForceIP() (*PySpin.PySpin.CameraBase method*), 141
- ForceIP() (*PySpin.PySpin.ICameraBase method*), 168
- frameID (*PySpin.PySpin.DeviceEventExposureEndData property*), 153
- frameID (*PySpin.PySpin.DeviceEventInferenceData property*), 154
- frameRate (*PySpin.PySpin.AVIOption property*), 81

- [frameRate \(PySpin.PySpin.H264Option property\)](#), 168
[frameRate \(PySpin.PySpin.MJPGOption property\)](#), 372
[FromString\(\) \(PySpin.PySpin.CBooleanPtr method\)](#), 83
[FromString\(\) \(PySpin.PySpin.CCategoryPtr method\)](#), 86
[FromString\(\) \(PySpin.PySpin.CCommandPtr method\)](#), 88
[FromString\(\) \(PySpin.PySpin.CEnumEntryPtr method\)](#), 92
[FromString\(\) \(PySpin.PySpin.CEnumerationPtr method\)](#), 94
[FromString\(\) \(PySpin.PySpin.CIntegerPtr method\)](#), 98
[FromString\(\) \(PySpin.PySpin.CRegisterPtr method\)](#), 106
[FromString\(\) \(PySpin.PySpin.CStringPtr method\)](#), 111
[FromString\(\) \(PySpin.PySpin.CValuePtr method\)](#), 113
[FromString\(\) \(PySpin.PySpin.EAccessModeClass static method\)](#), 154
[FromString\(\) \(PySpin.PySpin.ECachingModeClass static method\)](#), 155
[FromString\(\) \(PySpin.PySpin.EDisplayNotationClass static method\)](#), 155
[FromString\(\) \(PySpin.PySpin.EEndianessClass static method\)](#), 156
[FromString\(\) \(PySpin.PySpin.EGenApiSchemaVersionClass static method\)](#), 156
[FromString\(\) \(PySpin.PySpin.EInputDirectionClass static method\)](#), 156
[FromString\(\) \(PySpin.PySpin.ENamespaceClass static method\)](#), 157
[FromString\(\) \(PySpin.PySpin.ERepresentationClass static method\)](#), 157
[FromString\(\) \(PySpin.PySpin.ESignClass static method\)](#), 158
[FromString\(\) \(PySpin.PySpin.ESlopeClass static method\)](#), 158
[FromString\(\) \(PySpin.PySpin.EStandardNameSpaceClass static method\)](#), 159
[FromString\(\) \(PySpin.PySpin.EVisibilityClass static method\)](#), 159
[FromString\(\) \(PySpin.PySpin.EYesNoClass static method\)](#), 160
[FromString\(\) \(PySpin.PySpin.IValue method\)](#), 339
[FromString\(\) \(PySpin.PySpin.ValueNode method\)](#), 403
[front\(\) \(PySpin.PySpin.node_vector method\)](#), 409
[front\(\) \(PySpin.PySpin.value_vector method\)](#), 410
[fullmessage \(PySpin.SpinnakerException attribute\)](#), 69
- ## G
- [g \(PySpin.PySpin.Stereo3DPoint property\)](#), 389
[Gain \(PySpin.Camera property\)](#), 25
[Gain \(PySpin.PySpin.Camera property\)](#), 130
[GainAuto \(PySpin.Camera property\)](#), 25
[GainAuto \(PySpin.PySpin.Camera property\)](#), 130
[GainAutoBalance \(PySpin.Camera property\)](#), 25
[GainAutoBalance \(PySpin.PySpin.Camera property\)](#), 130
[GainConversion \(PySpin.Camera property\)](#), 25
[GainConversion \(PySpin.PySpin.Camera property\)](#), 130
[GainSelector \(PySpin.Camera property\)](#), 25
[GainSelector \(PySpin.PySpin.Camera property\)](#), 130
[Gamma \(PySpin.Camera property\)](#), 25
[Gamma \(PySpin.PySpin.Camera property\)](#), 130
[GammaEnable \(PySpin.Camera property\)](#), 25
[GammaEnable \(PySpin.PySpin.Camera property\)](#), 130
[gcstring \(class in PySpin.PySpin\)](#), 404
[GenICamXMLLocation \(PySpin.PySpin.TransportLayerDevice property\)](#), 397
[GenICamXMLLocation \(PySpin.TransportLayerDevice property\)](#), 76
[GenICamXMLPath \(PySpin.PySpin.TransportLayerDevice property\)](#), 397
[GenICamXMLPath \(PySpin.TransportLayerDevice property\)](#), 76
[GenTL_SFNC_Version_Major \(PySpin.PySpin.TransportLayerSystem property\)](#), 401
[GenTL_SFNC_Version_Minor \(PySpin.PySpin.TransportLayerSystem property\)](#), 401
[GenTL_SFNC_Version_SubMinor \(PySpin.PySpin.TransportLayerSystem property\)](#), 401
[GenTLVersionMajor \(PySpin.PySpin.TransportLayerSystem property\)](#), 401
[GenTLVersionMinor \(PySpin.PySpin.TransportLayerSystem property\)](#), 401
[Get\(\) \(PySpin.PySpin.CRegisterPtr method\)](#), 106
[Get\(\) \(PySpin.PySpin.IRegister method\)](#), 335
[Get\(\) \(PySpin.PySpin.RegisterNode method\)](#), 384
[GetAccessMode\(\) \(PySpin.CameraBase method\)](#), 36
[GetAccessMode\(\) \(PySpin.CBasePtr method\)](#), 10
[GetAccessMode\(\) \(PySpin.PySpin.CameraBase method\)](#), 141
[GetAccessMode\(\) \(PySpin.PySpin.CBasePtr method\)](#), 83
[GetAccessMode\(\) \(PySpin.PySpin.CBooleanPtr method\)](#), 83
[GetAccessMode\(\) \(PySpin.PySpin.CCategoryPtr method\)](#), 86
[GetAccessMode\(\) \(PySpin.PySpin.CCommandPtr method\)](#), 88
[GetAccessMode\(\) \(PySpin.PySpin.CEnumEntryPtr method\)](#), 92
[GetAccessMode\(\) \(PySpin.PySpin.CEnumerationPtr method\)](#), 94

GetAccessMode()	(PySpin.PySpin.CIntegerPtr method), 98	method), 360
GetAccessMode()	(PySpin.PySpin.CNodePtr method), 104	GetBoxSize() (PySpin.PySpin.InferenceBoundingBoxResult method), 360
GetAccessMode()	(PySpin.PySpin.CRegisterPtr method), 107	GetBufferOwnership() (PySpin.CameraBase method), 36
GetAccessMode()	(PySpin.PySpin.CSelectorPtr method), 109	GetBufferOwnership() (PySpin.PySpin.CameraBase method), 141
GetAccessMode()	(PySpin.PySpin.CStringPtr method), 111	GetBufferOwnership() (PySpin.PySpin.ICameraBase method), 169
GetAccessMode()	(PySpin.PySpin.CValuePtr method), 113	GetBufferSize() (PySpin.Image method), 48
GetAccessMode()	(PySpin.PySpin.IBase method), 168	GetBufferSize() (PySpin.PySpin.IImage method), 321
GetAccessMode()	(PySpin.PySpin.ICameraBase method), 169	GetBufferSize() (PySpin.PySpin.Image method), 341
GetAccessMode()	(PySpin.PySpin.Node method), 373	GetByDeviceID() (PySpin.CameraList method), 40
GetActiveNumDataStreams()	(PySpin.CameraBase method), 36	GetByDeviceID() (PySpin.PySpin.CameraList method), 145
GetActiveNumDataStreams()	(PySpin.PySpin.CameraBase method), 141	GetByDeviceID() (PySpin.PySpin.ICameraList method), 171
GetActiveNumDataStreams()	(PySpin.PySpin.ICameraBase method), 169	GetByIndex() (PySpin.CameraList method), 40
GetAddress()	(PySpin.PySpin.CRegisterPtr method), 107	GetByIndex() (PySpin.ImageList method), 55
GetAddress()	(PySpin.PySpin.IRegister method), 336	GetByIndex() (PySpin.InterfaceList method), 67
GetAddress()	(PySpin.PySpin.RegisterNode method), 385	GetByIndex() (PySpin.PySpin.CameraList method), 145
GetAlias()	(PySpin.PySpin.CBooleanPtr method), 83	GetByIndex() (PySpin.PySpin.ICameraList method), 171
GetAlias()	(PySpin.PySpin.CCategoryPtr method), 86	GetByIndex() (PySpin.PySpin.IImageList method), 325
GetAlias()	(PySpin.PySpin.CCommandPtr method), 89	GetByIndex() (PySpin.PySpin.IInterfaceList method), 329
GetAlias()	(PySpin.PySpin.CEnumEntryPtr method), 92	GetByIndex() (PySpin.PySpin.ImageList method), 349
GetAlias()	(PySpin.PySpin.CEnumerationPtr method), 94	GetByIndex() (PySpin.PySpin.InterfaceList method), 365
GetAlias()	(PySpin.PySpin.CIntegerPtr method), 99	GetByInterfaceID() (PySpin.InterfaceList method), 67
GetAlias()	(PySpin.PySpin.CNodePtr method), 104	GetByInterfaceID() (PySpin.PySpin.InterfaceList method), 365
GetAlias()	(PySpin.PySpin.CRegisterPtr method), 107	GetByPayloadType() (PySpin.ImageList method), 55
GetAlias()	(PySpin.PySpin.CStringPtr method), 111	GetByPayloadType() (PySpin.PySpin.IImageList method), 325
GetAlias()	(PySpin.PySpin.CValuePtr method), 113	GetByPayloadType() (PySpin.PySpin.ImageList method), 349
GetAlias()	(PySpin.PySpin.INode method), 330	GetByPixelFormat() (PySpin.ImageList method), 55
GetAlias()	(PySpin.PySpin.Node method), 373	GetByPixelFormat() (PySpin.PySpin.IImageList method), 325
GetBitsPerPixel()	(PySpin.Image method), 47	GetByPixelFormat() (PySpin.PySpin.ImageList method), 349
GetBitsPerPixel()	(PySpin.PySpin.IImage method), 321	GetBySerial() (PySpin.CameraList method), 40
GetBitsPerPixel()	(PySpin.PySpin.Image method), 341	GetBySerial() (PySpin.PySpin.CameraList method), 145
GetBlackLevel()	(PySpin.ChunkData method), 42	GetBySerial() (PySpin.PySpin.ICameraList method), 171
GetBlackLevel()	(PySpin.PySpin.ChunkData method), 148	GetByStreamIndex() (PySpin.ImageList method), 55
GetBlackLevel()	(PySpin.PySpin.IChunkData method), 172	GetByStreamIndex() (PySpin.PySpin.IImageList method), 325
GetBoxAt()	(PySpin.PySpin.InferenceBoundingBoxResult method), 360	GetByStreamIndex() (PySpin.PySpin.ImageList method), 349
GetBoxCount()	(PySpin.PySpin.InferenceBoundingBoxResult method), 360	

GetCachingMode()	(PySpin.PySpin.CBooleanPtr method), 83	86	GetChildren()	(PySpin.PySpin.CCommandPtr method), 89	
GetCachingMode()	(PySpin.PySpin.CCategoryPtr method), 86		GetChildren()	(PySpin.PySpin.CEnumEntryPtr method), 92	
GetCachingMode()	(PySpin.PySpin.CCommandPtr method), 89		GetChildren()	(PySpin.PySpin.CEnumerationPtr method), 94	
GetCachingMode()	(PySpin.PySpin.CEnumEntryPtr method), 92		GetChildren()	(PySpin.PySpin.CIntegerPtr method), 99	
GetCachingMode()	(PySpin.PySpin.CEnumerationPtr method), 94		GetChildren()	(PySpin.PySpin.CNodePtr method), 104	
GetCachingMode()	(PySpin.PySpin.CIntegerPtr method), 99		GetChildren()	(PySpin.PySpin.CRegisterPtr method), 107	
GetCachingMode()	(PySpin.PySpin.CNodePtr method), 104		GetChildren()	(PySpin.PySpin.CStringPtr method), 111	
GetCachingMode()	(PySpin.PySpin.CRegisterPtr method), 107		GetChildren()	(PySpin.PySpin.CValuePtr method), 113	
GetCachingMode()	(PySpin.PySpin.CStringPtr method), 111		GetChildren()	(PySpin.PySpin.INode method), 330	
GetCachingMode()	(PySpin.PySpin.CValuePtr method), 113		GetChildren()	(PySpin.PySpin.Node method), 373	
GetCachingMode()	(PySpin.PySpin.INode method), 330		GetChunkData()	(PySpin.Image method), 48	
GetCachingMode()	(PySpin.PySpin.Node method), 373		GetChunkData()	(PySpin.PySpin.IImage method), 321	
GetCameras()	(PySpin.IInterface method), 66		GetChunkData()	(PySpin.PySpin.Image method), 341	
GetCameras()	(PySpin.PySpin.IInterface method), 328		GetChunkLayoutId()	(PySpin.Image method), 48	
GetCameras()	(PySpin.PySpin.ISystem method), 337		GetChunkLayoutId()	(PySpin.PySpin.IImage method), 321	
GetCameras()	(PySpin.PySpin.System method), 391		GetChunkLayoutId()	(PySpin.PySpin.Image method), 341	
GetCameras()	(PySpin.System method), 70		GetColorProcessing()	(PySpin.Image method), 48	
GetCastAlias()	(PySpin.PySpin.CBooleanPtr method), 83		GetColorProcessing()	(PySpin.ImageProcessor method), 57	
GetCastAlias()	(PySpin.PySpin.CCategoryPtr method), 86		GetColorProcessing()	(PySpin.PySpin.IImage method), 321	
GetCastAlias()	(PySpin.PySpin.CCommandPtr method), 89		GetColorProcessing()	(PySpin.PySpin.IImageProcessor method), 327	
GetCastAlias()	(PySpin.PySpin.CEnumEntryPtr method), 92		GetColorProcessing()	(PySpin.PySpin.Image method), 341	
GetCastAlias()	(PySpin.PySpin.CEnumerationPtr method), 94		GetColorProcessing()	(PySpin.PySpin.ImageProcessor method), 351	
GetCastAlias()	(PySpin.PySpin.CIntegerPtr method), 99		GetCompressionMode()	(PySpin.ChunkData method), 42	
GetCastAlias()	(PySpin.PySpin.CNodePtr method), 104		GetCompressionMode()	(PySpin.PySpin.ChunkData method), 148	
GetCastAlias()	(PySpin.PySpin.CRegisterPtr method), 107		GetCompressionMode()	(PySpin.PySpin.IChunkData method), 172	
GetCastAlias()	(PySpin.PySpin.CStringPtr method), 111		GetCompressionRatio()	(PySpin.ChunkData method), 42	
GetCastAlias()	(PySpin.PySpin.CValuePtr method), 113		GetCompressionRatio()	(PySpin.PySpin.ChunkData method), 148	
GetCastAlias()	(PySpin.PySpin.INode method), 330		GetCompressionRatio()	(PySpin.PySpin.IChunkData method), 172	
GetCastAlias()	(PySpin.PySpin.Node method), 373		GetCounterValue()	(PySpin.ChunkData method), 42	
GetCategoryName()	(PySpin.PySpin.LoggingEventData method), 371		GetCounterValue()	(PySpin.PySpin.ChunkData method), 148	
GetChildren()	(PySpin.PySpin.CBooleanPtr method), 83				
GetChildren()	(PySpin.PySpin.CCategoryPtr method),				

GetCounterValue() (PySpin.PySpin.IChunkData method), 172	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrect method), 189
GetCRC() (PySpin.ChunkData method), 42	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrect method), 190
GetCRC() (PySpin.PySpin.ChunkData method), 148	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkBlackLevelSel method), 190
GetCRC() (PySpin.PySpin.IChunkData method), 172	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkCounterSelect method), 191
GetCurrentDatarate() (PySpin.ChunkData method), 43	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkEncoderSelect method), 192
GetCurrentDatarate() (PySpin.PySpin.ChunkData method), 148	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkEncoderStatu method), 192
GetCurrentDatarate() (PySpin.PySpin.IChunkData method), 172	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkExposureTime method), 193
GetCurrentEntry() (PySpin.PySpin.CEnumerationPtr method), 94	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkGainSelector method), 194
GetCurrentEntry() (PySpin.PySpin.EnumNode method), 161	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkImageCompo method), 194
GetCurrentEntry() (PySpin.PySpin.IEnumeration method), 176	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkPixelFormatE method), 195
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Acquis method), 177	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkRegionIDenu method), 196
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Acquis method), 177	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoord method), 197
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Action method), 178	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoord method), 197
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Action method), 178	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoord method), 198
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Adc method), 179	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoord method), 199
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Auto method), 180	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoord method), 199
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Auto method), 180	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dDistan method), 200
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Auto method), 181	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dOutput method), 201
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Auto method), 182	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkSelectorEnum method), 201
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Auto method), 182	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkSourceIDenu method), 202
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Balanc method), 183	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkTimerSelector method), 203
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Balanc method), 184	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ChunkTransferStream method), 203
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Balanc method), 184	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CIConfigurationEnu method), 204
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Binn method), 185	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CITimeSlotsCountEn method), 205
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Binn method), 186	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ColorTransformation method), 205
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Binn method), 186	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ColorTransformation method), 206
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Black method), 187	GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ComponentDestinat method), 206
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Black method), 188	
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_Black method), 188	

method), 207

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CompareCurrentEntry() method), 207

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CompareCurrentEntry() method), 208

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 209

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 209

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 210

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 211

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 211

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 212

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 213

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CountCurrentEntry() method), 213

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CxpGetCurrentEntry() method), 214

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CxpGetCurrentEntry() method), 215

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CxpGetCurrentEntry() method), 215

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CxpGetCurrentEntry() method), 216

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_CxpGetCurrentEntry() method), 217

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DecisionGetCurrentEntry() method), 217

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DecisionGetCurrentEntry() method), 218

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DecisionGetCurrentEntry() method), 219

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DefectGetCurrentEntry() method), 219

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeinterleaveGetCurrentEntry() method), 220

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 221

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 221

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 222

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 223

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 223

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 224

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceGetCurrentEntry() method), 225

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceLinkHeartbeatGetCurrentEntry() method), 225

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceLinkThroughGetCurrentEntry() method), 226

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DevicePowerSupplyGetCurrentEntry() method), 227

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceRegistersEndGetCurrentEntry() method), 227

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceScanTypeEnumGetCurrentEntry() method), 228

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceSensorChromGetCurrentEntry() method), 229

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceSerialPortBaseGetCurrentEntry() method), 229

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceSerialPortSelectorGetCurrentEntry() method), 230

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceStreamChannelGetCurrentEntry() method), 231

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceStreamChannelGetCurrentEntry() method), 231

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceTapGeometryGetCurrentEntry() method), 233

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceTemperatureSensorGetCurrentEntry() method), 233

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceTLTypeEnumGetCurrentEntry() method), 232

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceTypeEnumGetCurrentEntry() method), 234

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_DeviceTypeEnumGetCurrentEntry() method), 235

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderModeEnumGetCurrentEntry() method), 235

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderOutputModeGetCurrentEntry() method), 236

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderResetActivationGetCurrentEntry() method), 237

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderResetSourceGetCurrentEntry() method), 237

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderSelectorEnumGetCurrentEntry() method), 238

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderSourceAEnumGetCurrentEntry() method), 239

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderSourceBEnumGetCurrentEntry() method), 239

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EncoderStatusEnumGetCurrentEntry() method), 240

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EventNotificationEnumGetCurrentEntry() method), 241

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_EventSelectorEnumGetCurrentEntry() method), 241

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExposureActiveModeGetCurrentEntry() method), 242

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExposureAutoEnumGetCurrentEntry() method), 242

method), 243

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExposureModesEnum), 243

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExposureModesEnum), 244

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExposureModesEnum), 245

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ExternalDataEnum), 245

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FirmwareEnum), 247

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FileFormatEnum), 247

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FileFormatEnum), 248

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FileFormatEnum), 249

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FileFormatEnum), 249

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_FLIREnum), 246

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GainEnum), 251

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GainEnum), 251

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GainEnum), 252

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GainEnum), 253

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GenlockEnum), 253

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 254

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 255

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 255

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 256

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 257

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 257

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 258

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 259

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 259

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 260

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 261

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 261

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionsEnum), 262

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_GUIXMLLocationEnum), 250

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ImageComponentSelectionModeEnum), 263

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionJpegEnum), 263

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionMpegEnum), 264

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionNoneEnum), 265

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionRawEnum), 265

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_InterfaceTypeEnum), 265

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LensShadingCoefficientEnum), 267

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LensShadingCorrectionEnum), 267

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineFormatEnums), 268

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineInputFilterSelectedEnum), 269

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineModeEnums), 269

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineSelectorEnums), 270

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineSourceEnums), 271

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LineSourceEnums), 271

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputEnum), 271

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputEnum), 272

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputEnum), 273

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputEnum), 273

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTSelectorEnum), 274

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LogicBlockSelectorEnum), 274

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_LUTSelectorEnums), 266

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_MultiRoiConfigurationEnum), 275

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_MultiRoiSelectorEnum), 275

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_PixelColorFilterEnum), 277

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_PixelFormatEnums), 277

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_PixelFormatInfoSelectorEnum), 278

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_PixelSizeEnums), 279

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_POEStatusEnum), 279

method), 276

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_RegioMethod), 279

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_RegioMethod), 280

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_RegioMethod), 281

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_RgbTransformEntry), 281

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 282

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 283

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 283

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 284

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 285

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 285

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_ScanMethod), 286

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SensorMethod), 287

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SensorMethod), 287

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SensorMethod), 288

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 289

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 289

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 290

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 291

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 291

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SequenceMethod), 292

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SerialPortEntry), 293

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SerialPortEntry), 293

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SerialPortEntry), 294

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SerialPortEntry), 295

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SerialPortEntry), 295

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SoftTriggerEntry), 296

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_SourceMethod), 297

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_StereoResolutionEntry), 297

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_StreamBufferCountMethod), 298

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_StreamBufferHandleMethod), 299

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_StreamModeEnum), 299

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_StreamTypeEnum), 300

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TeledyneGigeVisionEntry), 301

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TestPatternEnums), 302

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TestPatternGeneratorEntry), 303

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TimerSelectorEnum), 303

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TimerStatusEnums), 304

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TimerTriggerActivationMethod), 305

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TimerTriggerSourceMethod), 305

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TLTypeEnum), 301

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferComponentMethod), 306

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferControlModeMethod), 307

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferOperationMethod), 307

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferQueueModeMethod), 308

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferSelectorEnum), 309

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferStatusSelectorMethod), 309

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferTriggerActivationMethod), 310

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferTriggerModeMethod), 311

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferTriggerSelectorMethod), 311

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TransferTriggerSourceMethod), 312

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TriggerActivationEnum), 313

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TriggerModeEnums), 313

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TriggerOverlapEnum), 314

GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TriggerSelectorEnum), 314

method), 315	GetDeviceEventName()
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_TriggerSourcePtr method), 153	GetCurrentEntry() (PySpin.PySpin.DeviceEventHandler method), 153
method), 315	
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_U3VGetDeviceEventName()	method), 316
method), 316	(PySpin.PySpin.IDeviceEventHandler method), 153
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_UserOutputSelectorEnums	method), 317
method), 317	GetDeviceID() (PySpin.CameraBase method), 36
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_UserSetDeviceID()	method), 317
method), 317	SetDeviceID() (PySpin.PySpin.CameraBase method), 141
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_UserSetDeviceID()	method), 318
method), 318	SetDeviceID() (PySpin.PySpin.ICameraBase method), 169
GetCurrentEntry() (PySpin.PySpin.IEnumerationT_WhiteGetDeviceName()	method), 319
method), 319	GetDeviceName() (PySpin.PySpin.CBooleanPtr method), 83
GetData() (PySpin.PySpin.IImage method), 321	GetDeviceName() (PySpin.PySpin.CCategoryPtr method), 86
GetDataAbsoluteMax() (PySpin.Image method), 48	method), 86
GetDataAbsoluteMax() (PySpin.PySpin.IImage	GetDeviceName() (PySpin.PySpin.CCommandPtr method), 89
method), 321	method), 89
GetDataAbsoluteMax() (PySpin.PySpin.Image	GetDeviceName() (PySpin.PySpin.CEnumEntryPtr method), 92
method), 341	method), 92
GetDataAbsoluteMin() (PySpin.Image method), 48	GetDeviceName() (PySpin.PySpin.CEnumerationPtr method), 94
GetDataAbsoluteMin() (PySpin.PySpin.IImage	method), 94
method), 321	GetDeviceName() (PySpin.PySpin.CIntegerPtr method), 99
GetDataAbsoluteMin() (PySpin.PySpin.Image	method), 99
method), 342	GetDeviceName() (PySpin.PySpin.CNodeMapDynPtr method), 102
GetDescription() (PySpin.PySpin.CBooleanPtr	method), 102
method), 83	GetDeviceName() (PySpin.PySpin.CNodeMapPtr method), 104
GetDescription() (PySpin.PySpin.CCategoryPtr	method), 104
method), 86	GetDeviceName() (PySpin.PySpin.CNodePtr method), 104
GetDescription() (PySpin.PySpin.CCommandPtr	method), 104
method), 89	GetDeviceName() (PySpin.PySpin.CRegisterPtr method), 107
GetDescription() (PySpin.PySpin.CEnumEntryPtr	method), 107
method), 92	GetDeviceName() (PySpin.PySpin.CStringPtr method), 111
GetDescription() (PySpin.PySpin.CEnumerationPtr	method), 111
method), 94	GetDeviceName() (PySpin.PySpin.CValuePtr method), 113
GetDescription() (PySpin.PySpin.CIntegerPtr	method), 113
method), 99	GetDeviceName() (PySpin.PySpin.INode method), 330
GetDescription() (PySpin.PySpin.CNodePtr method), 104	GetDeviceName() (PySpin.PySpin.INodeMap method), 332
method), 104	method), 332
GetDescription() (PySpin.PySpin.CRegisterPtr	GetDeviceName() (PySpin.PySpin.Node method), 374
method), 107	GetDeviceName() (PySpin.PySpin.NodeMap method), 378
GetDescription() (PySpin.PySpin.CStringPtr	method), 378
method), 111	GetDeviceVersion() (PySpin.PySpin.CDeviceInfoPtr method), 91
GetDescription() (PySpin.PySpin.CValuePtr	method), 91
method), 113	GetDeviceVersion() (PySpin.PySpin.IDeviceInfo method), 174
GetDescription() (PySpin.PySpin.INode method), 330	method), 174
GetDescription() (PySpin.PySpin.Node method), 374	GetDeviceVersion() (PySpin.PySpin.NodeMap method), 378
GetDeviceEventId() (PySpin.DeviceEventHandler	method), 378
method), 5	GetDisplayName() (PySpin.PySpin.CBooleanPtr method), 83
GetDeviceEventId() (PySpin.PySpin.DeviceEventHandler	method), 83
method), 153	GetDisplayName() (PySpin.PySpin.CCategoryPtr method), 86
GetDeviceEventId() (PySpin.PySpin.IDeviceEventHandler	method), 86
method), 174	GetDisplayName() (PySpin.PySpin.CCommandPtr method), 89
GetDeviceEventName() (PySpin.DeviceEventHandler	method), 89
method), 6	GetDisplayName() (PySpin.PySpin.CEnumEntryPtr method), 92
method), 6	method), 92

- method*), 92
- GetDisplayName() (PySpin.PySpin.CEnumerationPtr *method*), 95
- GetDisplayName() (PySpin.PySpin.CIntegerPtr *method*), 99
- GetDisplayName() (PySpin.PySpin.CNodePtr *method*), 104
- GetDisplayName() (PySpin.PySpin.CRegisterPtr *method*), 107
- GetDisplayName() (PySpin.PySpin.CStringPtr *method*), 111
- GetDisplayName() (PySpin.PySpin.CValuePtr *method*), 114
- GetDisplayName() (PySpin.PySpin.INode *method*), 330
- GetDisplayName() (PySpin.PySpin.Node *method*), 374
- GetDisplayNotation() (PySpin.PySpin.FloatNode *method*), 163
- GetDisplayNotation() (PySpin.PySpin.IFloat *method*), 319
- GetDisplayPrecision() (PySpin.PySpin.FloatNode *method*), 163
- GetDisplayPrecision() (PySpin.PySpin.IFloat *method*), 319
- GetDocuURL() (PySpin.PySpin.CBooleanPtr *method*), 83
- GetDocuURL() (PySpin.PySpin.CCategoryPtr *method*), 86
- GetDocuURL() (PySpin.PySpin.CCommandPtr *method*), 89
- GetDocuURL() (PySpin.PySpin.CEnumEntryPtr *method*), 92
- GetDocuURL() (PySpin.PySpin.CEnumerationPtr *method*), 95
- GetDocuURL() (PySpin.PySpin.CIntegerPtr *method*), 99
- GetDocuURL() (PySpin.PySpin.CNodePtr *method*), 104
- GetDocuURL() (PySpin.PySpin.CRegisterPtr *method*), 107
- GetDocuURL() (PySpin.PySpin.CStringPtr *method*), 111
- GetDocuURL() (PySpin.PySpin.CValuePtr *method*), 114
- GetDocuURL() (PySpin.PySpin.INode *method*), 330
- GetDocuURL() (PySpin.PySpin.Node *method*), 374
- GetEnable() (PySpin.ChunkData *method*), 43
- GetEnable() (PySpin.PySpin.ChunkData *method*), 148
- GetEnable() (PySpin.PySpin.IChunkData *method*), 172
- GetEncoderValue() (PySpin.ChunkData *method*), 43
- GetEncoderValue() (PySpin.PySpin.ChunkData *method*), 148
- GetEncoderValue() (PySpin.PySpin.IChunkData *method*), 172
- GetEntries() (PySpin.PySpin.CEnumerationPtr *method*), 95
- GetEntries() (PySpin.PySpin.EnumNode *method*), 161
- GetEntries() (PySpin.PySpin.IEnumeration *method*), 176
- GetEntry() (PySpin.PySpin.CEnumerationPtr *method*), 95
- GetEntry() (PySpin.PySpin.EnumNode *method*), 161
- GetEntry() (PySpin.PySpin.IEnumeration *method*), 176
- GetEntry() (PySpin.PySpin.IEnumerationT_AcquisitionModeEnums *method*), 177
- GetEntry() (PySpin.PySpin.IEnumerationT_AcquisitionStatusSelectorEnums *method*), 177
- GetEntry() (PySpin.PySpin.IEnumerationT_ActionSelectorEnums *method*), 178
- GetEntry() (PySpin.PySpin.IEnumerationT_ActionUnconditionalModeEnums *method*), 179
- GetEntry() (PySpin.PySpin.IEnumerationT_AdcBitDepthEnums *method*), 179
- GetEntry() (PySpin.PySpin.IEnumerationT_AutoAlgorithmSelectorEnums *method*), 180
- GetEntry() (PySpin.PySpin.IEnumerationT_AutoExposureControlPriority *method*), 181
- GetEntry() (PySpin.PySpin.IEnumerationT_AutoExposureLightingModeEnums *method*), 181
- GetEntry() (PySpin.PySpin.IEnumerationT_AutoExposureMeteringModelEnums *method*), 182
- GetEntry() (PySpin.PySpin.IEnumerationT_AutoExposureTargetGreyValueEnums *method*), 183
- GetEntry() (PySpin.PySpin.IEnumerationT_BalanceRatioSelectorEnums *method*), 183
- GetEntry() (PySpin.PySpin.IEnumerationT_BalanceWhiteAutoEnums *method*), 184
- GetEntry() (PySpin.PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums *method*), 185
- GetEntry() (PySpin.PySpin.IEnumerationT_BinningHorizontalModeEnums *method*), 185
- GetEntry() (PySpin.PySpin.IEnumerationT_BinningSelectorEnums *method*), 186
- GetEntry() (PySpin.PySpin.IEnumerationT_BinningVerticalModeEnums *method*), 187
- GetEntry() (PySpin.PySpin.IEnumerationT_BlackLevelAutoBalanceEnums *method*), 187
- GetEntry() (PySpin.PySpin.IEnumerationT_BlackLevelAutoEnums *method*), 188
- GetEntry() (PySpin.PySpin.IEnumerationT_BlackLevelSelectorEnums *method*), 189
- GetEntry() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums *method*), 189
- GetEntry() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums *method*), 190
- GetEntry() (PySpin.PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums *method*), 191
- GetEntry() (PySpin.PySpin.IEnumerationT_ChunkCounterSelectorEnums *method*), 191
- GetEntry() (PySpin.PySpin.IEnumerationT_ChunkEncoderSelectorEnums *method*), 192
- GetEntry() (PySpin.PySpin.IEnumerationT_ChunkEncoderStatusEnums *method*), 193

GetEntry() (PySpin.PySpin.IEnumerationT_ChunkExposureEnums method), 193
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkGainSelectorEnums method), 194
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkImageCompressionEnums method), 195
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkPixelFormatEnums method), 195
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkRegionOfInterestEnums method), 196
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 197
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 197
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 198
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 199
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 199
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 200
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkScan3dEnums method), 201
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkSelectorEnums method), 201
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkSourceEnums method), 202
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkTimerSelectorEnums method), 203
 GetEntry() (PySpin.PySpin.IEnumerationT_ChunkTransferEnums method), 203
 GetEntry() (PySpin.PySpin.IEnumerationT_ClConfigurationEnums method), 204
 GetEntry() (PySpin.PySpin.IEnumerationT_ClTimeSlotsConfigurationEnums method), 205
 GetEntry() (PySpin.PySpin.IEnumerationT_ColorTransformEnums method), 205
 GetEntry() (PySpin.PySpin.IEnumerationT_ColorTransformEnums method), 206
 GetEntry() (PySpin.PySpin.IEnumerationT_ComponentDescriptorEnums method), 207
 GetEntry() (PySpin.PySpin.IEnumerationT_ComponentSelectorEnums method), 207
 GetEntry() (PySpin.PySpin.IEnumerationT_CompressionSelectorEnums method), 208
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterEventEnums method), 209
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterEventEnums method), 209
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterResetEnums method), 210
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterResetEnums method), 211
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterSelectorEnums method), 211
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterStatusEnums method), 212
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterTriggerActivationEnums method), 213
 GetEntry() (PySpin.PySpin.IEnumerationT_CounterTriggerSourceEnums method), 213
 GetEntry() (PySpin.PySpin.IEnumerationT_CxpConnectionTestModeEnums method), 214
 GetEntry() (PySpin.PySpin.IEnumerationT_CxpLinkConfigurationEnums method), 215
 GetEntry() (PySpin.PySpin.IEnumerationT_CxpLinkConfigurationPreferenceEnums method), 215
 GetEntry() (PySpin.PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums method), 216
 GetEntry() (PySpin.PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 217
 GetEntry() (PySpin.PySpin.IEnumerationT_DecimationHorizontalModeEnums method), 217
 GetEntry() (PySpin.PySpin.IEnumerationT_DecimationSelectorEnums method), 218
 GetEntry() (PySpin.PySpin.IEnumerationT_DecimationVerticalModeEnums method), 219
 GetEntry() (PySpin.PySpin.IEnumerationT_DefectCorrectionModeEnums method), 219
 GetEntry() (PySpin.PySpin.IEnumerationT_DeinterlacingEnums method), 220
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceAccessStatusEnum method), 221
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceCharacterSetEnums method), 221
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceClockSelectorEnums method), 222
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceConnectionStatusEnum method), 223
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceCurrentSpeedEnum method), 223
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceEndianessMechanismEnum method), 224
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 225
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnum method), 225
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceLinkThroughputLimitEnum method), 226
 GetEntry() (PySpin.PySpin.IEnumerationT_DevicePowerSupplySelectorEnum method), 227
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceRegistersEndiannessEnum method), 227
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceScanTypeEnums method), 228
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceSensorChromaEnums method), 229

GetEntry() (PySpin.PySpin.IEnumerationT_DeviceSerialPortTypeEnums method), 229
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceSerialPortTypeEnums method), 230
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceStreamTransferTypeEnums method), 231
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceStreamTransferTypeEnums method), 231
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceTapGainEnums method), 233
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceTemperatureEnums method), 233
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceTLTypeEnums method), 232
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceTypeEnums method), 234
 GetEntry() (PySpin.PySpin.IEnumerationT_DeviceTypeEnums method), 235
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderModeEnums method), 235
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderOutputTypeEnums method), 236
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderResetTypeEnums method), 237
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderResetTypeEnums method), 237
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderSelectorEnums method), 238
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderSourceEnums method), 239
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderSourceEnums method), 239
 GetEntry() (PySpin.PySpin.IEnumerationT_EncoderStatusEnums method), 240
 GetEntry() (PySpin.PySpin.IEnumerationT_EventNotificationTypeEnums method), 241
 GetEntry() (PySpin.PySpin.IEnumerationT_EventSelectorEnums method), 241
 GetEntry() (PySpin.PySpin.IEnumerationT_ExposureActiveEnums method), 242
 GetEntry() (PySpin.PySpin.IEnumerationT_ExposureAutoEnums method), 243
 GetEntry() (PySpin.PySpin.IEnumerationT_ExposureModeEnums method), 243
 GetEntry() (PySpin.PySpin.IEnumerationT_ExposureTimeEnums method), 244
 GetEntry() (PySpin.PySpin.IEnumerationT_ExposureTimeEnums method), 245
 GetEntry() (PySpin.PySpin.IEnumerationT_ExternalVoltageEnums method), 245
 GetEntry() (PySpin.PySpin.IEnumerationT_FfcModeEnums method), 247
 GetEntry() (PySpin.PySpin.IEnumerationT_FileOpenModeEnums method), 247
 GetEntry() (PySpin.PySpin.IEnumerationT_FileOperationSelectorEnums method), 248
 GetEntry() (PySpin.PySpin.IEnumerationT_FileOperationStatusEnums method), 249
 GetEntry() (PySpin.PySpin.IEnumerationT_FileSelectorEnums method), 249
 GetEntry() (PySpin.PySpin.IEnumerationT_FLIRFilterDriverStatusEnums method), 246
 GetEntry() (PySpin.PySpin.IEnumerationT_GainAutoBalanceEnums method), 251
 GetEntry() (PySpin.PySpin.IEnumerationT_GainAutoEnums method), 251
 GetEntry() (PySpin.PySpin.IEnumerationT_GainConversionEnums method), 252
 GetEntry() (PySpin.PySpin.IEnumerationT_GainSelectorEnums method), 253
 GetEntry() (PySpin.PySpin.IEnumerationT_GenICamXMLLocationEnums method), 253
 GetEntry() (PySpin.PySpin.IEnumerationT_GevCCPEnum method), 254
 GetEntry() (PySpin.PySpin.IEnumerationT_GevCCPEnum method), 255
 GetEntry() (PySpin.PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums method), 255
 GetEntry() (PySpin.PySpin.IEnumerationT_GevGVCPExtendedStatusCodesEnums method), 256
 GetEntry() (PySpin.PySpin.IEnumerationT_GevGVSPExtendedIDModeEnums method), 257
 GetEntry() (PySpin.PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnums method), 257
 GetEntry() (PySpin.PySpin.IEnumerationT_GevIEEE1588ModeEnums method), 258
 GetEntry() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusEnums method), 259
 GetEntry() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums method), 259
 GetEntry() (PySpin.PySpin.IEnumerationT_GevIPConfigurationStatusEnums method), 260
 GetEntry() (PySpin.PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums method), 261
 GetEntry() (PySpin.PySpin.IEnumerationT_GevSCPDirectionEnums method), 261
 GetEntry() (PySpin.PySpin.IEnumerationT_GevSupportedOptionSelectorEnums method), 262
 GetEntry() (PySpin.PySpin.IEnumerationT_GUIXMLLocationEnums method), 250
 GetEntry() (PySpin.PySpin.IEnumerationT_ImageComponentSelectorEnums method), 263
 GetEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionJPEGFormatEnums method), 263
 GetEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionModeEnums method), 264
 GetEntry() (PySpin.PySpin.IEnumerationT_ImageCompressionRateOptionsEnums method), 265

GetEntry() (PySpin.PySpin.IEnumerationT_InterfaceTypeEnums method), 265
 GetEntry() (PySpin.PySpin.IEnumerationT_LensShadingEnums method), 267
 GetEntry() (PySpin.PySpin.IEnumerationT_LensShadingEnums method), 267
 GetEntry() (PySpin.PySpin.IEnumerationT_LineFormatEnums method), 268
 GetEntry() (PySpin.PySpin.IEnumerationT_LineInputFilterEnums method), 269
 GetEntry() (PySpin.PySpin.IEnumerationT_LineModeEnums method), 269
 GetEntry() (PySpin.PySpin.IEnumerationT_LineSelectorEnums method), 270
 GetEntry() (PySpin.PySpin.IEnumerationT_LineSourceEnums method), 271
 GetEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTEnums method), 271
 GetEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTEnums method), 272
 GetEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTEnums method), 273
 GetEntry() (PySpin.PySpin.IEnumerationT_LogicBlockLUTEnums method), 273
 GetEntry() (PySpin.PySpin.IEnumerationT_LogicBlockSelectorEnums method), 274
 GetEntry() (PySpin.PySpin.IEnumerationT_LUTSelectorEnums method), 266
 GetEntry() (PySpin.PySpin.IEnumerationT_MultiRoiConfigurationEnums method), 275
 GetEntry() (PySpin.PySpin.IEnumerationT_MultiRoiSelectorEnums method), 276
 GetEntry() (PySpin.PySpin.IEnumerationT_PixelColorFilterEnums method), 277
 GetEntry() (PySpin.PySpin.IEnumerationT_PixelFormatEnums method), 277
 GetEntry() (PySpin.PySpin.IEnumerationT_PixelFormatEnums method), 278
 GetEntry() (PySpin.PySpin.IEnumerationT_PixelSizeEnums method), 279
 GetEntry() (PySpin.PySpin.IEnumerationT_POEStatusEnums method), 276
 GetEntry() (PySpin.PySpin.IEnumerationT_RegionDestinationEnums method), 279
 GetEntry() (PySpin.PySpin.IEnumerationT_RegionModeEnums method), 280
 GetEntry() (PySpin.PySpin.IEnumerationT_RegionSelectorEnums method), 281
 GetEntry() (PySpin.PySpin.IEnumerationT_RgbTransformEnums method), 281
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 282
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 283
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemEnums method), 283
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceEnums method), 284
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateTransformEnums method), 285
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dDistanceUnitEnums method), 286
 GetEntry() (PySpin.PySpin.IEnumerationT_Scan3dOutputModeEnums method), 286
 GetEntry() (PySpin.PySpin.IEnumerationT_SensorDigitizationTapsEnums method), 287
 GetEntry() (PySpin.PySpin.IEnumerationT_SensorShutterModeEnums method), 287
 GetEntry() (PySpin.PySpin.IEnumerationT_SensorTapsEnums method), 288
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerConfigurationModeEnums method), 289
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerConfigurationValidEnums method), 289
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerModeEnums method), 290
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerSetValidEnums method), 291
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerTriggerActivationEnums method), 291
 GetEntry() (PySpin.PySpin.IEnumerationT_SequencerTriggerSourceEnums method), 292
 GetEntry() (PySpin.PySpin.IEnumerationT_SerialPortBaudRateEnums method), 293
 GetEntry() (PySpin.PySpin.IEnumerationT_SerialPortParityEnums method), 293
 GetEntry() (PySpin.PySpin.IEnumerationT_SerialPortSelectorEnums method), 294
 GetEntry() (PySpin.PySpin.IEnumerationT_SerialPortSourceEnums method), 295
 GetEntry() (PySpin.PySpin.IEnumerationT_SerialPortStopBitsEnums method), 295
 GetEntry() (PySpin.PySpin.IEnumerationT_SoftwareSignalSelectorEnums method), 296
 GetEntry() (PySpin.PySpin.IEnumerationT_SourceSelectorEnums method), 297
 GetEntry() (PySpin.PySpin.IEnumerationT_StereoResolutionEnums method), 297
 GetEntry() (PySpin.PySpin.IEnumerationT_StreamBufferCountModeEnums method), 298
 GetEntry() (PySpin.PySpin.IEnumerationT_StreamBufferHandlingModeEnums method), 299
 GetEntry() (PySpin.PySpin.IEnumerationT_StreamModeEnums method), 299
 GetEntry() (PySpin.PySpin.IEnumerationT_StreamTypeEnums method), 300
 GetEntry() (PySpin.PySpin.IEnumerationT_TeledyneGigeVisionFilterDriveEnums method), 301

[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TestPatternEnums method), 302
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TestPatternEnums method), 303
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TimerSelectorEnums method), 303
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TimerStatusEnums method), 304
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TimerTriggerEnums method), 305
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TimerTriggerEnums method), 305
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TLTypeEnum method), 301
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferComponentSelectorEnums method), 306
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferControlModeEnums method), 307
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferOperationModeEnums method), 307
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferQueueModeEnums method), 308
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferSelectorEnums method), 309
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferSelectorEnums method), 309
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferTriggerEnums method), 310
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferTriggerEnums method), 311
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferTriggerSelectorEnums method), 311
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TransferTriggerSourceEnums method), 312
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TriggerActivationEnums method), 313
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TriggerModeEnums method), 313
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TriggerOverlapEnums method), 314
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TriggerSelectorEnums method), 315
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_TriggerSourceEnums method), 315
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_U3VCurrentSensorEnums method), 316
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_UserOutputSelectorEnums method), 317
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_UserSetDefaultEnums method), 317
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_UserSetSelectorEnums method), 318
[GetEntry\(\)](#) (PySpin.PySpin.IEnumerationT_WhiteClipSelectorEnums method), 319
[GetEntryByName\(\)](#) (PySpin.PySpin.CEnumerationPtr method), 95
[GetEntryByName\(\)](#) (PySpin.PySpin.EnumNode method), 161
[GetEntryByName\(\)](#) (PySpin.PySpin.IEnumeration method), 176
[GetEnumAlias\(\)](#) (PySpin.PySpin.CFloatPtr method), 98
[GetEnumAlias\(\)](#) (PySpin.PySpin.FloatNode method), 163
[GetErrorMessage\(\)](#) (in module PySpin.PySpin), 166
[GetEventID\(\)](#) (PySpin.PySpin.CBooleanPtr method), 84
[GetEventID\(\)](#) (PySpin.PySpin.CCategoryPtr method), 84
[GetEventID\(\)](#) (PySpin.PySpin.CCommandPtr method), 84
[GetEventID\(\)](#) (PySpin.PySpin.CEnumEntryPtr method), 92
[GetEventID\(\)](#) (PySpin.PySpin.CEnumerationPtr method), 95
[GetEventID\(\)](#) (PySpin.PySpin.CIntegerPtr method), 99
[GetEventID\(\)](#) (PySpin.PySpin.CNodePtr method), 104
[GetEventID\(\)](#) (PySpin.PySpin.CRegisterPtr method), 107
[GetEventID\(\)](#) (PySpin.PySpin.CStringPtr method), 111
[GetEventID\(\)](#) (PySpin.PySpin.CValuePtr method), 114
[GetEventID\(\)](#) (PySpin.PySpin.INode method), 330
[GetEventID\(\)](#) (PySpin.PySpin.Node method), 374
[GetEventData\(\)](#) (PySpin.EventHandler method), 6
[GetEventData\(\)](#) (PySpin.PySpin.EventHandler method), 163
[GetEventDataSize\(\)](#) (PySpin.EventHandler method), 6
[GetEventDataSize\(\)](#) (PySpin.PySpin.EventHandler method), 163
[GetEventType\(\)](#) (PySpin.EventHandler method), 6
[GetEventType\(\)](#) (PySpin.PySpin.EventHandler method), 163
[GetExposureEndLineStatusAll\(\)](#) (PySpin.ChunkData method), 43
[GetExposureEndLineStatusAll\(\)](#) (PySpin.PySpin.ChunkData method), 148
[GetExposureEndLineStatusAll\(\)](#) (PySpin.PySpin.IChunkData method), 172
[GetExposureTime\(\)](#) (PySpin.ChunkData method), 43
[GetExposureTime\(\)](#) (PySpin.PySpin.ChunkData method), 148
[GetExposureTime\(\)](#) (PySpin.PySpin.IChunkData method), 172
[GetFeatureBagHandle\(\)](#) (PySpin.PySpin.CFeatureBag method), 97
[GetFeatures\(\)](#) (PySpin.PySpin.CategoryNode method), 97

- method*), 147
- GetFeatures() (*PySpin.PySpin.CCategoryPtr method*), 86
- GetFeatures() (*PySpin.PySpin.ICategory method*), 172
- GetFiles() (*in module PySpin.PySpin*), 166
- GetFloatAlias() (*PySpin.PySpin.IntegerNode method*), 362
- GetFrameID() (*PySpin.ChunkData method*), 43
- GetFrameID() (*PySpin.Image method*), 48
- GetFrameID() (*PySpin.PySpin.ChunkData method*), 149
- GetFrameID() (*PySpin.PySpin.IChunkData method*), 172
- GetFrameID() (*PySpin.PySpin.IImage method*), 321
- GetFrameID() (*PySpin.PySpin.Image method*), 342
- GetGain() (*PySpin.ChunkData method*), 43
- GetGain() (*PySpin.PySpin.ChunkData method*), 149
- GetGain() (*PySpin.PySpin.IChunkData method*), 172
- GetGenApiVersion() (*PySpin.PySpin.CDeviceInfoPtr method*), 91
- GetGenApiVersion() (*PySpin.PySpin.IDeviceInfo method*), 174
- GetGenApiVersion() (*PySpin.PySpin.NodeMap method*), 379
- GetGenICamCacheFolder() (*in module PySpin.PySpin*), 166
- GetGenICamCLProtocolFolder() (*in module PySpin.PySpin*), 166
- GetGenICamLogConfig() (*in module PySpin.PySpin*), 166
- GetGuiXml() (*PySpin.CameraBase method*), 36
- GetGuiXml() (*PySpin.PySpin.CameraBase method*), 141
- GetGuiXml() (*PySpin.PySpin.ICameraBase method*), 169
- GetHeatmapColorGradient() (*PySpin.ImageUtilityHeatmap static method*), 61
- GetHeatmapColorGradient() (*PySpin.PySpin.ImageUtilityHeatmap static method*), 355
- GetHeatmapRange() (*PySpin.ImageUtilityHeatmap static method*), 61
- GetHeatmapRange() (*PySpin.PySpin.ImageUtilityHeatmap static method*), 355
- GetHeight() (*PySpin.ChunkData method*), 43
- GetHeight() (*PySpin.Image method*), 48
- GetHeight() (*PySpin.PySpin.ChunkData method*), 149
- GetHeight() (*PySpin.PySpin.IChunkData method*), 172
- GetHeight() (*PySpin.PySpin.IImage method*), 321
- GetHeight() (*PySpin.PySpin.Image method*), 342
- GetID() (*PySpin.Image method*), 48
- GetID() (*PySpin.PySpin.IImage method*), 321
- GetID() (*PySpin.PySpin.Image method*), 342
- GetImage() (*PySpin.ChunkData method*), 43
- GetImage() (*PySpin.PySpin.ChunkData method*), 149
- GetImage() (*PySpin.PySpin.IChunkData method*), 172
- GetImagePayloadType() (*PySpin.Image method*), 48
- GetImagePayloadType() (*PySpin.PySpin.IImage method*), 321
- GetImagePayloadType() (*PySpin.PySpin.Image method*), 342
- GetImageSize() (*PySpin.Image method*), 49
- GetImageSize() (*PySpin.PySpin.IImage method*), 321
- GetImageSize() (*PySpin.PySpin.Image method*), 342
- GetImageStatus() (*PySpin.Image method*), 49
- GetImageStatus() (*PySpin.PySpin.IImage method*), 321
- GetImageStatus() (*PySpin.PySpin.Image method*), 342
- GetImageStatusDescription() (*PySpin.Image static method*), 49
- GetImageStatusDescription() (*PySpin.PySpin.Image static method*), 342
- GetInc() (*PySpin.PySpin.CIntegerPtr method*), 99
- GetInc() (*PySpin.PySpin.FloatNode method*), 163
- GetInc() (*PySpin.PySpin.IFloat method*), 319
- GetInc() (*PySpin.PySpin.IInteger method*), 327
- GetInc() (*PySpin.PySpin.IntegerNode method*), 362
- GetIncMode() (*PySpin.PySpin.CIntegerPtr method*), 99
- GetIncMode() (*PySpin.PySpin.FloatNode method*), 163
- GetIncMode() (*PySpin.PySpin.IFloat method*), 319
- GetIncMode() (*PySpin.PySpin.IInteger method*), 327
- GetIncMode() (*PySpin.PySpin.IntegerNode method*), 362
- GetInferenceBoundingBoxResult() (*PySpin.ChunkData method*), 43
- GetInferenceBoundingBoxResult() (*PySpin.PySpin.ChunkData method*), 149
- GetInferenceBoundingBoxResult() (*PySpin.PySpin.IChunkData method*), 172
- GetInferenceConfidence() (*PySpin.ChunkData method*), 43
- GetInferenceConfidence() (*PySpin.PySpin.ChunkData method*), 149
- GetInferenceConfidence() (*PySpin.PySpin.IChunkData method*), 172
- GetInferenceFrameId() (*PySpin.ChunkData method*), 43
- GetInferenceFrameId() (*PySpin.PySpin.ChunkData method*), 149
- GetInferenceFrameId() (*PySpin.PySpin.IChunkData method*), 172
- GetInferenceResult() (*PySpin.ChunkData method*), 43
- GetInferenceResult() (*PySpin.PySpin.ChunkData method*), 149
- GetInferenceResult() (*PySpin.PySpin.IChunkData method*), 172
- GetInstance() (*PySpin.PySpin.System static method*),

- 392
- `GetInstance()` (*PySpin.System static method*), 71
- `GetIntAlias()` (*PySpin.PySpin.CFloatPtr method*), 98
- `GetIntAlias()` (*PySpin.PySpin.FloatNode method*), 163
- `GetInterfaceName()` (*in module PySpin.PySpin*), 166
- `GetInterfaces()` (*PySpin.PySpin.ISystem method*), 337
- `GetInterfaces()` (*PySpin.PySpin.System method*), 392
- `GetInterfaces()` (*PySpin.System method*), 71
- `GetIntValue()` (*PySpin.PySpin.CEnumerationPtr method*), 95
- `GetIntValue()` (*PySpin.PySpin.EnumNode method*), 162
- `GetIntValue()` (*PySpin.PySpin.IEnumeration method*), 176
- `GetLength()` (*PySpin.PySpin.CRegisterPtr method*), 107
- `GetLength()` (*PySpin.PySpin.IRegister method*), 336
- `GetLength()` (*PySpin.PySpin.RegisterNode method*), 385
- `GetLibraryVersion()` (*PySpin.PySpin.ISystem method*), 337
- `GetLibraryVersion()` (*PySpin.PySpin.System method*), 392
- `GetLibraryVersion()` (*PySpin.System method*), 71
- `GetLinePitch()` (*PySpin.ChunkData method*), 43
- `GetLinePitch()` (*PySpin.PySpin.ChunkData method*), 149
- `GetLinePitch()` (*PySpin.PySpin.IChunkData method*), 172
- `GetLineStatusAll()` (*PySpin.ChunkData method*), 43
- `GetLineStatusAll()` (*PySpin.PySpin.ChunkData method*), 149
- `GetLineStatusAll()` (*PySpin.PySpin.IChunkData method*), 172
- `GetListOfValidValues()` (*PySpin.PySpin.CIntegerPtr method*), 99
- `GetListOfValidValues()` (*PySpin.PySpin.FloatNode method*), 164
- `GetListOfValidValues()` (*PySpin.PySpin.IFloat method*), 320
- `GetListOfValidValues()` (*PySpin.PySpin.IInteger method*), 327
- `GetListOfValidValues()` (*PySpin.PySpin.IntegerNode method*), 362
- `GetLockNodes()` (*PySpin.PySpin.CBooleanPtr method*), 84
- `GetLockNodes()` (*PySpin.PySpin.CCategoryPtr method*), 86
- `GetLockNodes()` (*PySpin.PySpin.CCommandPtr method*), 89
- `GetLockNodes()` (*PySpin.PySpin.CEnumEntryPtr method*), 92
- `GetLockNodes()` (*PySpin.PySpin.CEnumerationPtr method*), 95
- `GetLockNodes()` (*PySpin.PySpin.CIntegerPtr method*), 99
- `GetLockNodes()` (*PySpin.PySpin.CNodePtr method*), 104
- `GetLockNodes()` (*PySpin.PySpin.CRegisterPtr method*), 107
- `GetLockNodes()` (*PySpin.PySpin.CStringPtr method*), 111
- `GetLockNodes()` (*PySpin.PySpin.CValuePtr method*), 114
- `GetLockNodes()` (*PySpin.PySpin.INode method*), 330
- `GetLockNodes()` (*PySpin.PySpin.Node method*), 374
- `GetLoggingEventPriorityLevel()` (*PySpin.PySpin.ISystem method*), 337
- `GetLoggingEventPriorityLevel()` (*PySpin.PySpin.System method*), 392
- `GetLoggingEventPriorityLevel()` (*PySpin.System method*), 72
- `GetLogMessage()` (*PySpin.PySpin.LoggingEventData method*), 371
- `GetMax()` (*PySpin.PySpin.CIntegerPtr method*), 99
- `GetMax()` (*PySpin.PySpin.FloatNode method*), 164
- `GetMax()` (*PySpin.PySpin.IFloat method*), 320
- `GetMax()` (*PySpin.PySpin.IInteger method*), 327
- `GetMax()` (*PySpin.PySpin.IntegerNode method*), 362
- `GetMaxLength()` (*PySpin.PySpin.CStringPtr method*), 111
- `GetMaxLength()` (*PySpin.PySpin.IString method*), 337
- `GetMaxLength()` (*PySpin.PySpin.StringNode method*), 389
- `GetMin()` (*PySpin.PySpin.CIntegerPtr method*), 99
- `GetMin()` (*PySpin.PySpin.FloatNode method*), 164
- `GetMin()` (*PySpin.PySpin.IFloat method*), 320
- `GetMin()` (*PySpin.PySpin.IInteger method*), 327
- `GetMin()` (*PySpin.PySpin.IntegerNode method*), 362
- `GetModeActive()` (*PySpin.ChunkData method*), 43
- `GetModeActive()` (*PySpin.PySpin.ChunkData method*), 149
- `GetModeActive()` (*PySpin.PySpin.IChunkData method*), 172
- `GetModelName()` (*PySpin.PySpin.CDeviceInfoPtr method*), 91
- `GetModelName()` (*PySpin.PySpin.IDeviceInfo method*), 175
- `GetModelName()` (*PySpin.PySpin.NodeMap method*), 379
- `GetModulePathFromFunction()` (*in module PySpin.PySpin*), 167
- `GetName()` (*PySpin.PySpin.CBooleanPtr method*), 84
- `GetName()` (*PySpin.PySpin.CCategoryPtr method*), 87
- `GetName()` (*PySpin.PySpin.CCommandPtr method*), 89
- `GetName()` (*PySpin.PySpin.CEnumEntryPtr method*), 92

`GetName()` (*PySpin.PySpin.CEnumerationPtr method*), 95

`GetName()` (*PySpin.PySpin.CIntegerPtr method*), 99

`GetName()` (*PySpin.PySpin.CNodePtr method*), 105

`GetName()` (*PySpin.PySpin.CRegisterPtr method*), 107

`GetName()` (*PySpin.PySpin.CStringPtr method*), 111

`GetName()` (*PySpin.PySpin.CValuePtr method*), 114

`GetName()` (*PySpin.PySpin.INode method*), 331

`GetName()` (*PySpin.PySpin.Node method*), 374

`GetNameSpace()` (*PySpin.PySpin.CBooleanPtr method*), 84

`GetNameSpace()` (*PySpin.PySpin.CCategoryPtr method*), 87

`GetNameSpace()` (*PySpin.PySpin.CCommandPtr method*), 89

`GetNameSpace()` (*PySpin.PySpin.CEnumEntryPtr method*), 92

`GetNameSpace()` (*PySpin.PySpin.CEnumerationPtr method*), 95

`GetNameSpace()` (*PySpin.PySpin.CIntegerPtr method*), 99

`GetNameSpace()` (*PySpin.PySpin.CNodePtr method*), 105

`GetNameSpace()` (*PySpin.PySpin.CRegisterPtr method*), 108

`GetNameSpace()` (*PySpin.PySpin.CStringPtr method*), 111

`GetNameSpace()` (*PySpin.PySpin.CValuePtr method*), 114

`GetNameSpace()` (*PySpin.PySpin.INode method*), 331

`GetNameSpace()` (*PySpin.PySpin.Node method*), 374

`GetNDArray()` (*PySpin.PySpin.IImage method*), 321

`GetNDC()` (*PySpin.PySpin.LoggingEventData method*), 371

`GetNextImage()` (*PySpin.CameraBase method*), 36

`GetNextImage()` (*PySpin.PySpin.CameraBase method*), 141

`GetNextImage()` (*PySpin.PySpin.ICameraBase method*), 169

`GetNextImageSync()` (*PySpin.CameraBase method*), 36

`GetNextImageSync()` (*PySpin.PySpin.CameraBase method*), 142

`GetNextImageSync()` (*PySpin.PySpin.ICameraBase method*), 169

`GetNode()` (*PySpin.PySpin.CBooleanPtr method*), 84

`GetNode()` (*PySpin.PySpin.CCategoryPtr method*), 87

`GetNode()` (*PySpin.PySpin.CCommandPtr method*), 89

`GetNode()` (*PySpin.PySpin.CEnumEntryPtr method*), 92

`GetNode()` (*PySpin.PySpin.CEnumerationPtr method*), 95

`GetNode()` (*PySpin.PySpin.CIntegerPtr method*), 99

`GetNode()` (*PySpin.PySpin.CNodeMapDynPtr method*), 102

`GetNode()` (*PySpin.PySpin.CNodeMapPtr method*), 104

`GetNode()` (*PySpin.PySpin.CRegisterPtr method*), 108

`GetNode()` (*PySpin.PySpin.CStringPtr method*), 111

`GetNode()` (*PySpin.PySpin.CValuePtr method*), 114

`GetNode()` (*PySpin.PySpin.INodeMap method*), 332

`GetNode()` (*PySpin.PySpin.IValue method*), 339

`GetNode()` (*PySpin.PySpin.NodeMap method*), 379

`GetNode()` (*PySpin.PySpin.ValueNode method*), 403

`GetNodeHandle()` (*PySpin.PySpin.Node method*), 374

`GetNodeMap()` (*PySpin.CameraBase method*), 37

`GetNodeMap()` (*PySpin.PySpin.CameraBase method*), 142

`GetNodeMap()` (*PySpin.PySpin.CBooleanPtr method*), 84

`GetNodeMap()` (*PySpin.PySpin.CCategoryPtr method*), 87

`GetNodeMap()` (*PySpin.PySpin.CCommandPtr method*), 89

`GetNodeMap()` (*PySpin.PySpin.CEnumEntryPtr method*), 92

`GetNodeMap()` (*PySpin.PySpin.CEnumerationPtr method*), 95

`GetNodeMap()` (*PySpin.PySpin.CIntegerPtr method*), 99

`GetNodeMap()` (*PySpin.PySpin.CNodePtr method*), 105

`GetNodeMap()` (*PySpin.PySpin.CRegisterPtr method*), 108

`GetNodeMap()` (*PySpin.PySpin.CStringPtr method*), 111

`GetNodeMap()` (*PySpin.PySpin.CValuePtr method*), 114

`GetNodeMap()` (*PySpin.PySpin.ICameraBase method*), 169

`GetNodeMap()` (*PySpin.PySpin.INode method*), 331

`GetNodeMap()` (*PySpin.PySpin.Node method*), 374

`GetNodeMapHandle()` (*PySpin.PySpin.NodeMap method*), 379

`GetNodes()` (*PySpin.PySpin.CNodeMapDynPtr method*), 102

`GetNodes()` (*PySpin.PySpin.CNodeMapPtr method*), 104

`GetNodes()` (*PySpin.PySpin.INodeMap method*), 332

`GetNodes()` (*PySpin.PySpin.NodeMap method*), 379

`GetNumChannels()` (*PySpin.Image method*), 49

`GetNumChannels()` (*PySpin.PySpin.IImage method*), 321

`GetNumChannels()` (*PySpin.PySpin.Image method*), 342

`GetNumDataStreams()` (*PySpin.CameraBase method*), 37

`GetNumDataStreams()` (*PySpin.PySpin.CameraBase method*), 142

`GetNumDataStreams()` (*PySpin.PySpin.ICameraBase method*), 169

`GetNumDecompressionThreads()` (*PySpin.ImageProcessor method*), 57

`GetNumDecompressionThreads()` (*PySpin.PySpin.ImageProcessor method*),

- 327
- GetNumDecompressionThreads() (PySpin.PySpin.ImageProcessor method), 351
- GetNumericValue() (PySpin.PySpin.CEnumEntryPtr method), 92
- GetNumericValue() (PySpin.PySpin.EnumEntryNode method), 160
- GetNumericValue() (PySpin.PySpin.IEnumEntry method), 175
- GetNumImagesInUse() (PySpin.CameraBase method), 37
- GetNumImagesInUse() (PySpin.PySpin.CameraBase method), 142
- GetNumImagesInUse() (PySpin.PySpin.ICameraBase method), 169
- GetNumNodes() (PySpin.PySpin.CNodeMapDynPtr method), 102
- GetNumNodes() (PySpin.PySpin.CNodeMapPtr method), 104
- GetNumNodes() (PySpin.PySpin.INodeMap method), 332
- GetNumNodes() (PySpin.PySpin.NodeMap method), 379
- GetNumPoints() (PySpin.PointCloud method), 68
- GetNumPoints() (PySpin.PySpin.IPointCloud method), 334
- GetNumPoints() (PySpin.PySpin.PointCloud method), 383
- GetOffsetX() (PySpin.ChunkData method), 44
- GetOffsetX() (PySpin.PySpin.ChunkData method), 149
- GetOffsetX() (PySpin.PySpin.IChunkData method), 172
- GetOffsetY() (PySpin.ChunkData method), 44
- GetOffsetY() (PySpin.PySpin.ChunkData method), 149
- GetOffsetY() (PySpin.PySpin.IChunkData method), 172
- GetParents() (PySpin.PySpin.CBooleanPtr method), 84
- GetParents() (PySpin.PySpin.CCategoryPtr method), 87
- GetParents() (PySpin.PySpin.CCommandPtr method), 89
- GetParents() (PySpin.PySpin.CEnumEntryPtr method), 92
- GetParents() (PySpin.PySpin.CEnumerationPtr method), 95
- GetParents() (PySpin.PySpin.CIntegerPtr method), 99
- GetParents() (PySpin.PySpin.CNodePtr method), 105
- GetParents() (PySpin.PySpin.CRegisterPtr method), 108
- GetParents() (PySpin.PySpin.CStringPtr method), 111
- GetParents() (PySpin.PySpin.CValuePtr method), 114
- GetParents() (PySpin.PySpin.INode method), 331
- GetParents() (PySpin.PySpin.Node method), 374
- GetPartSelector() (PySpin.ChunkData method), 44
- GetPartSelector() (PySpin.PySpin.ChunkData method), 149
- GetPartSelector() (PySpin.PySpin.IChunkData method), 172
- GetPayloadType() (PySpin.Image method), 49
- GetPayloadType() (PySpin.PySpin.IImage method), 321
- GetPayloadType() (PySpin.PySpin.Image method), 342
- GetPixelDynamicRangeMax() (PySpin.ChunkData method), 44
- GetPixelDynamicRangeMax() (PySpin.PySpin.ChunkData method), 149
- GetPixelDynamicRangeMax() (PySpin.PySpin.IChunkData method), 173
- GetPixelDynamicRangeMin() (PySpin.ChunkData method), 44
- GetPixelDynamicRangeMin() (PySpin.PySpin.ChunkData method), 150
- GetPixelDynamicRangeMin() (PySpin.PySpin.IChunkData method), 173
- GetPixelFormat() (PySpin.Image method), 49
- GetPixelFormat() (PySpin.PySpin.IImage method), 321
- GetPixelFormat() (PySpin.PySpin.Image method), 342
- GetPixelFormatIntType() (PySpin.Image method), 49
- GetPixelFormatIntType() (PySpin.PySpin.IImage method), 321
- GetPixelFormatIntType() (PySpin.PySpin.Image method), 343
- GetPixelFormatName() (PySpin.Image method), 49
- GetPixelFormatName() (PySpin.PySpin.IImage method), 321
- GetPixelFormatName() (PySpin.PySpin.Image method), 343
- GetPoint() (PySpin.PointCloud method), 68
- GetPoint() (PySpin.PySpin.IPointCloud method), 334
- GetPoint() (PySpin.PySpin.PointCloud method), 383
- GetPointCloudData() (PySpin.PointCloud method), 68
- GetPointCloudData() (PySpin.PySpin.IPointCloud method), 334
- GetPointCloudData() (PySpin.PySpin.PointCloud method), 383
- GetPollingTime() (PySpin.PySpin.CBooleanPtr method), 84
- GetPollingTime() (PySpin.PySpin.CCategoryPtr method), 87
- GetPollingTime() (PySpin.PySpin.CCommandPtr method), 89
- GetPollingTime() (PySpin.PySpin.CEnumEntryPtr method), 92
- GetPollingTime() (PySpin.PySpin.CEnumerationPtr method), 95

<i>method</i>), 95	
GetPollingTime() (PySpin.PySpin.CIntegerPtr <i>method</i>), 99	
GetPollingTime() (PySpin.PySpin.CNodePtr <i>method</i>), 105	
GetPollingTime() (PySpin.PySpin.CRegisterPtr <i>method</i>), 108	
GetPollingTime() (PySpin.PySpin.CStringPtr <i>method</i>), 111	
GetPollingTime() (PySpin.PySpin.CValuePtr <i>method</i>), 114	
GetPollingTime() (PySpin.PySpin.INode <i>method</i>), 331	
GetPollingTime() (PySpin.PySpin.Node <i>method</i>), 375	
GetPrincipalInterfaceType() (PySpin.PySpin.CBooleanPtr <i>method</i>), 84	
GetPrincipalInterfaceType() (PySpin.PySpin.CCategoryPtr <i>method</i>), 87	
GetPrincipalInterfaceType() (PySpin.PySpin.CCommandPtr <i>method</i>), 89	
GetPrincipalInterfaceType() (PySpin.PySpin.CEnumEntryPtr <i>method</i>), 92	
GetPrincipalInterfaceType() (PySpin.PySpin.CEnumerationPtr <i>method</i>), 95	
GetPrincipalInterfaceType() (PySpin.PySpin.CIntegerPtr <i>method</i>), 99	
GetPrincipalInterfaceType() (PySpin.PySpin.CNodePtr <i>method</i>), 105	
GetPrincipalInterfaceType() (PySpin.PySpin.CRegisterPtr <i>method</i>), 108	
GetPrincipalInterfaceType() (PySpin.PySpin.CStringPtr <i>method</i>), 111	
GetPrincipalInterfaceType() (PySpin.PySpin.CValuePtr <i>method</i>), 114	
GetPrincipalInterfaceType() (PySpin.PySpin.INode <i>method</i>), 331	
GetPrincipalInterfaceType() (PySpin.PySpin.Node <i>method</i>), 375	
GetProperty() (PySpin.PySpin.CBooleanPtr <i>method</i>), 84	
GetProperty() (PySpin.PySpin.CCategoryPtr <i>method</i>), 87	
GetProperty() (PySpin.PySpin.CCommandPtr <i>method</i>), 89	
GetProperty() (PySpin.PySpin.CEnumEntryPtr <i>method</i>), 92	
GetProperty() (PySpin.PySpin.CEnumerationPtr <i>method</i>), 95	
GetProperty() (PySpin.PySpin.CIntegerPtr <i>method</i>), 99	
GetProperty() (PySpin.PySpin.CNodePtr <i>method</i>), 105	
GetProperty() (PySpin.PySpin.CRegisterPtr <i>method</i>), 108	
GetProperty() (PySpin.PySpin.CStringPtr <i>method</i>), 111	
GetProperty() (PySpin.PySpin.CValuePtr <i>method</i>), 114	
GetProperty() (PySpin.PySpin.INode <i>method</i>), 331	
GetProperty() (PySpin.PySpin.Node <i>method</i>), 375	
GetPropertyNames() (PySpin.PySpin.CBooleanPtr <i>method</i>), 84	
GetPropertyNames() (PySpin.PySpin.CCategoryPtr <i>method</i>), 87	
GetPropertyNames() (PySpin.PySpin.CCommandPtr <i>method</i>), 89	
GetPropertyNames() (PySpin.PySpin.CEnumEntryPtr <i>method</i>), 93	
GetPropertyNames() (PySpin.PySpin.CEnumerationPtr <i>method</i>), 95	
GetPropertyNames() (PySpin.PySpin.CIntegerPtr <i>method</i>), 100	
GetPropertyNames() (PySpin.PySpin.CNodePtr <i>method</i>), 105	
GetPropertyNames() (PySpin.PySpin.CRegisterPtr <i>method</i>), 108	
GetPropertyNames() (PySpin.PySpin.CStringPtr <i>method</i>), 112	
GetPropertyNames() (PySpin.PySpin.CValuePtr <i>method</i>), 114	
GetPropertyNames() (PySpin.PySpin.INode <i>method</i>), 331	
GetPropertyNames() (PySpin.PySpin.Node <i>method</i>), 375	
GetRepresentation() (PySpin.PySpin.CIntegerPtr <i>method</i>), 100	
GetRepresentation() (PySpin.PySpin.FloatNode <i>method</i>), 164	
GetRepresentation() (PySpin.PySpin.IFloat <i>method</i>), 320	
GetRepresentation() (PySpin.PySpin.IInteger <i>method</i>), 327	
GetRepresentation() (PySpin.PySpin.IntegerNode	

- method*), 362
- GetScan3dAxisMax() (*PySpin.ChunkData method*), 44
- GetScan3dAxisMax() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dAxisMax() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dAxisMin() (*PySpin.ChunkData method*), 44
- GetScan3dAxisMin() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dAxisMin() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dCoordinateOffset() (*PySpin.ChunkData method*), 44
- GetScan3dCoordinateOffset() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dCoordinateOffset() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dCoordinateReferenceValue() (*PySpin.ChunkData method*), 44
- GetScan3dCoordinateReferenceValue() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dCoordinateReferenceValue() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dCoordinateScale() (*PySpin.ChunkData method*), 44
- GetScan3dCoordinateScale() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dCoordinateScale() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dInvalidDataFlag() (*PySpin.ChunkData method*), 44
- GetScan3dInvalidDataFlag() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dInvalidDataFlag() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dInvalidDataValue() (*PySpin.ChunkData method*), 44
- GetScan3dInvalidDataValue() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dInvalidDataValue() (*PySpin.PySpin.IChunkData method*), 173
- GetScan3dTransformValue() (*PySpin.ChunkData method*), 45
- GetScan3dTransformValue() (*PySpin.PySpin.ChunkData method*), 150
- GetScan3dTransformValue() (*PySpin.PySpin.IChunkData method*), 173
- GetScanLineSelector() (*PySpin.ChunkData method*), 45
- GetScanLineSelector() (*PySpin.PySpin.ChunkData method*), 150
- GetScanLineSelector() (*PySpin.PySpin.IChunkData method*), 173
- GetSchemaVersion() (*PySpin.PySpin.CDeviceInfoPtr method*), 91
- GetSchemaVersion() (*PySpin.PySpin.IDeviceInfo method*), 175
- GetSchemaVersion() (*PySpin.PySpin.NodeMap method*), 379
- GetSelectedFeatures() (*PySpin.PySpin.CBooleanPtr method*), 84
- GetSelectedFeatures() (*PySpin.PySpin.CCategoryPtr method*), 87
- GetSelectedFeatures() (*PySpin.PySpin.CCommandPtr method*), 89
- GetSelectedFeatures() (*PySpin.PySpin.CEnumEntryPtr method*), 93
- GetSelectedFeatures() (*PySpin.PySpin.CEnumerationPtr method*), 96
- GetSelectedFeatures() (*PySpin.PySpin.CIntegerPtr method*), 100
- GetSelectedFeatures() (*PySpin.PySpin.CNodePtr method*), 105
- GetSelectedFeatures() (*PySpin.PySpin.CRegisterPtr method*), 108
- GetSelectedFeatures() (*PySpin.PySpin.CSelectorPtr method*), 109
- GetSelectedFeatures() (*PySpin.PySpin.CStringPtr method*), 112
- GetSelectedFeatures() (*PySpin.PySpin.CValuePtr method*), 114
- GetSelectedFeatures() (*PySpin.PySpin.ISelector method*), 336
- GetSelectedFeatures() (*PySpin.PySpin.Node method*), 375
- GetSelectingFeatures() (*PySpin.PySpin.CBooleanPtr method*), 84
- GetSelectingFeatures() (*PySpin.PySpin.CCategoryPtr method*), 87
- GetSelectingFeatures() (*PySpin.PySpin.CCommandPtr method*), 90
- GetSelectingFeatures() (*PySpin.PySpin.CEnumEntryPtr method*), 93
- GetSelectingFeatures() (*PySpin.PySpin.CEnumerationPtr method*), 96
- GetSelectingFeatures() (*PySpin.PySpin.CIntegerPtr method*), 100
- GetSelectingFeatures() (*PySpin.PySpin.CNodePtr method*), 105
- GetSelectingFeatures() (*PySpin.PySpin.CRegisterPtr method*), 108
- GetSelectingFeatures()

- (*PySpin.PySpin.CSelectorPtr* method), 109
- GetSelectingFeatures()* (*PySpin.PySpin.CStringPtr* method), 112
- GetSelectingFeatures()* (*PySpin.PySpin.CValuePtr* method), 114
- GetSelectingFeatures()* (*PySpin.PySpin.ISelector* method), 336
- GetSelectingFeatures()* (*PySpin.PySpin.Node* method), 375
- GetSelectorList()* (*PySpin.PySpin.CSelectorSet* method), 110
- GetSelectorList()* (*PySpin.PySpin.ISelectorDigit* method), 336
- GetSequencerSetActive()* (*PySpin.ChunkData* method), 45
- GetSequencerSetActive()* (*PySpin.PySpin.ChunkData* method), 150
- GetSequencerSetActive()* (*PySpin.PySpin.IChunkData* method), 173
- GetSerialData()* (*PySpin.ChunkData* method), 45
- GetSerialData()* (*PySpin.PySpin.ChunkData* method), 151
- GetSerialData()* (*PySpin.PySpin.IChunkData* method), 173
- GetSerialDataLength()* (*PySpin.ChunkData* method), 45
- GetSerialDataLength()* (*PySpin.PySpin.ChunkData* method), 151
- GetSerialDataLength()* (*PySpin.PySpin.IChunkData* method), 173
- GetSerialReceiveOverflow()* (*PySpin.ChunkData* method), 45
- GetSerialReceiveOverflow()* (*PySpin.PySpin.ChunkData* method), 151
- GetSerialReceiveOverflow()* (*PySpin.PySpin.IChunkData* method), 173
- GetSize()* (*PySpin.CameraList* method), 40
- GetSize()* (*PySpin.ImageList* method), 55
- GetSize()* (*PySpin.InterfaceList* method), 68
- GetSize()* (*PySpin.PySpin.CameraList* method), 146
- GetSize()* (*PySpin.PySpin.ICameraList* method), 171
- GetSize()* (*PySpin.PySpin.IImageList* method), 325
- GetSize()* (*PySpin.PySpin.IInterfaceList* method), 329
- GetSize()* (*PySpin.PySpin.ImageList* method), 349
- GetSize()* (*PySpin.PySpin.InterfaceList* method), 365
- GetStandardNameSpace()* (*PySpin.PySpin.CDeviceInfoPtr* method), 91
- GetStandardNameSpace()* (*PySpin.PySpin.IDeviceInfo* method), 175
- GetStandardNameSpace()* (*PySpin.PySpin.NodeMap* method), 380
- GetStreamChannelID()* (*PySpin.ChunkData* method), 45
- GetStreamChannelID()* (*PySpin.PySpin.ChunkData* method), 151
- GetStreamChannelID()* (*PySpin.PySpin.IChunkData* method), 173
- GetStreamIndex()* (*PySpin.Image* method), 50
- GetStreamIndex()* (*PySpin.PySpin.IImage* method), 321
- GetStreamIndex()* (*PySpin.PySpin.Image* method), 343
- GetStride()* (*PySpin.Image* method), 50
- GetStride()* (*PySpin.PySpin.IImage* method), 321
- GetStride()* (*PySpin.PySpin.Image* method), 343
- GetSupportedSchemaVersions()* (*PySpin.PySpin.CNodeMapDynPtr* method), 102
- GetSupportedSchemaVersions()* (*PySpin.PySpin.INodeMapDyn* method), 333
- GetSupportedSchemaVersions()* (*PySpin.PySpin.NodeMap* method), 380
- GetSymbolic()* (*PySpin.PySpin.CEnumEntryPtr* method), 93
- GetSymbolic()* (*PySpin.PySpin.EnumEntryNode* method), 160
- GetSymbolic()* (*PySpin.PySpin.IEnumEntry* method), 175
- GetSymbolics()* (*PySpin.PySpin.CEnumerationPtr* method), 96
- GetSymbolics()* (*PySpin.PySpin.EnumNode* method), 162
- GetSymbolics()* (*PySpin.PySpin.IEnumeration* method), 176
- GetThreadName()* (*PySpin.PySpin.LoggingEventData* method), 372
- GetTimerValue()* (*PySpin.ChunkData* method), 45
- GetTimerValue()* (*PySpin.PySpin.ChunkData* method), 151
- GetTimerValue()* (*PySpin.PySpin.IChunkData* method), 173
- GetTimestamp()* (*PySpin.ChunkData* method), 45
- GetTimestamp()* (*PySpin.Image* method), 50
- GetTimestamp()* (*PySpin.PySpin.ChunkData* method), 151
- GetTimestamp()* (*PySpin.PySpin.IChunkData* method), 173
- GetTimestamp()* (*PySpin.PySpin.IImage* method), 321
- GetTimestamp()* (*PySpin.PySpin.Image* method), 344
- GetTimestamp()* (*PySpin.PySpin.LoggingEventData* method), 372
- GetTimestampLatchValue()* (*PySpin.ChunkData* method), 45
- GetTimestampLatchValue()* (*PySpin.PySpin.ChunkData* method), 151
- GetTimestampLatchValue()* (*PySpin.PySpin.IChunkData* method), 173

- GetTLDeviceNodeMap() (*PySpin.CameraBase* method), 37
- GetTLDeviceNodeMap() (*PySpin.PySpin.CameraBase* method), 142
- GetTLDeviceNodeMap() (*PySpin.PySpin.ICameraBase* method), 169
- GetTLNodeMap() (*PySpin.IInterface* method), 66
- GetTLNodeMap() (*PySpin.PySpin.IInterface* method), 328
- GetTLNodeMap() (*PySpin.PySpin.ISystem* method), 338
- GetTLNodeMap() (*PySpin.PySpin.System* method), 392
- GetTLNodeMap() (*PySpin.System* method), 72
- GetTLPayloadType() (*PySpin.Image* method), 50
- GetTLPayloadType() (*PySpin.PySpin.IImage* method), 321
- GetTLPayloadType() (*PySpin.PySpin.Image* method), 343
- GetTLPixelFormat() (*PySpin.Image* method), 50
- GetTLPixelFormat() (*PySpin.PySpin.IImage* method), 321
- GetTLPixelFormat() (*PySpin.PySpin.Image* method), 343
- GetTLPixelFormatNamespace() (*PySpin.Image* method), 50
- GetTLPixelFormatNamespace() (*PySpin.PySpin.IImage* method), 321
- GetTLPixelFormatNamespace() (*PySpin.PySpin.Image* method), 343
- GetTLStreamNodeMap() (*PySpin.CameraBase* method), 37
- GetTLStreamNodeMap() (*PySpin.PySpin.CameraBase* method), 142
- GetTLStreamNodeMap() (*PySpin.PySpin.ICameraBase* method), 169
- GetToolTip() (*PySpin.PySpin.CBooleanPtr* method), 84
- GetToolTip() (*PySpin.PySpin.CCategoryPtr* method), 87
- GetToolTip() (*PySpin.PySpin.CCommandPtr* method), 90
- GetToolTip() (*PySpin.PySpin.CDeviceInfoPtr* method), 91
- GetToolTip() (*PySpin.PySpin.CEnumEntryPtr* method), 93
- GetToolTip() (*PySpin.PySpin.CEnumerationPtr* method), 96
- GetToolTip() (*PySpin.PySpin.CIntegerPtr* method), 100
- GetToolTip() (*PySpin.PySpin.CNodePtr* method), 105
- GetToolTip() (*PySpin.PySpin.CRegisterPtr* method), 108
- GetToolTip() (*PySpin.PySpin.CStringPtr* method), 112
- GetToolTip() (*PySpin.PySpin.CValuePtr* method), 114
- GetToolTip() (*PySpin.PySpin.IDeviceInfo* method), 175
- GetToolTip() (*PySpin.PySpin.INode* method), 331
- GetToolTip() (*PySpin.PySpin.Node* method), 375
- GetToolTip() (*PySpin.PySpin.NodeMap* method), 380
- GetTransferBlockID() (*PySpin.ChunkData* method), 45
- GetTransferBlockID() (*PySpin.PySpin.ChunkData* method), 151
- GetTransferBlockID() (*PySpin.PySpin.IChunkData* method), 173
- GetTransferQueueCurrentBlockCount() (*PySpin.ChunkData* method), 45
- GetTransferQueueCurrentBlockCount() (*PySpin.PySpin.ChunkData* method), 151
- GetTransferQueueCurrentBlockCount() (*PySpin.PySpin.IChunkData* method), 173
- GetUniqueID() (*PySpin.CameraBase* method), 37
- GetUniqueID() (*PySpin.PySpin.CameraBase* method), 143
- GetUniqueID() (*PySpin.PySpin.ICameraBase* method), 169
- GetUnit() (*PySpin.PySpin.CIntegerPtr* method), 100
- GetUnit() (*PySpin.PySpin.FloatNode* method), 164
- GetUnit() (*PySpin.PySpin.IFloat* method), 320
- GetUnit() (*PySpin.PySpin.IInteger* method), 327
- GetUnit() (*PySpin.PySpin.IntegerNode* method), 362
- GetUserBufferCount() (*PySpin.CameraBase* method), 37
- GetUserBufferCount() (*PySpin.PySpin.CameraBase* method), 143
- GetUserBufferCount() (*PySpin.PySpin.ICameraBase* method), 169
- GetUserBufferSize() (*PySpin.CameraBase* method), 37
- GetUserBufferSize() (*PySpin.PySpin.CameraBase* method), 143
- GetUserBufferSize() (*PySpin.PySpin.ICameraBase* method), 169
- GetUserBufferTotalSize() (*PySpin.CameraBase* method), 37
- GetUserBufferTotalSize() (*PySpin.PySpin.CameraBase* method), 143
- GetUserBufferTotalSize() (*PySpin.PySpin.ICameraBase* method), 169
- GetValidPayloadSize() (*PySpin.Image* method), 50
- GetValidPayloadSize() (*PySpin.PySpin.IImage* method), 321
- GetValidPayloadSize() (*PySpin.PySpin.Image* method), 344
- GetValue() (*PySpin.PySpin.BooleanNode* method), 82
- GetValue() (*PySpin.PySpin.CBooleanPtr* method), 84
- GetValue() (*PySpin.PySpin.CEnumEntryPtr* method), 93
- GetValue() (*PySpin.PySpin.CIntegerPtr* method), 100

- GetValue() (PySpin.PySpin.CStringPtr method), 112
- GetValue() (PySpin.PySpin.EnumEntryNode method), 160
- GetValue() (PySpin.PySpin.FloatNode method), 164
- GetValue() (PySpin.PySpin.IBoolean method), 168
- GetValue() (PySpin.PySpin.IEnumEntry method), 175
- GetValue() (PySpin.PySpin.IEnumerationT_AcquisitionMode method), 177
- GetValue() (PySpin.PySpin.IEnumerationT_AcquisitionStatus method), 177
- GetValue() (PySpin.PySpin.IEnumerationT_ActionSelector method), 178
- GetValue() (PySpin.PySpin.IEnumerationT_ActionUnconditional method), 179
- GetValue() (PySpin.PySpin.IEnumerationT_AdcBitDepth method), 179
- GetValue() (PySpin.PySpin.IEnumerationT_AutoAlgorithm method), 180
- GetValue() (PySpin.PySpin.IEnumerationT_AutoExposure method), 181
- GetValue() (PySpin.PySpin.IEnumerationT_AutoExposureRegion method), 181
- GetValue() (PySpin.PySpin.IEnumerationT_AutoExposureRegion method), 182
- GetValue() (PySpin.PySpin.IEnumerationT_AutoExposureRegion method), 183
- GetValue() (PySpin.PySpin.IEnumerationT_BalanceRatio method), 183
- GetValue() (PySpin.PySpin.IEnumerationT_BalanceWhite method), 184
- GetValue() (PySpin.PySpin.IEnumerationT_BalanceWhite method), 185
- GetValue() (PySpin.PySpin.IEnumerationT_BinningHorizontal method), 185
- GetValue() (PySpin.PySpin.IEnumerationT_BinningSelector method), 186
- GetValue() (PySpin.PySpin.IEnumerationT_BinningVertical method), 187
- GetValue() (PySpin.PySpin.IEnumerationT_BlackLevelAuto method), 187
- GetValue() (PySpin.PySpin.IEnumerationT_BlackLevelAuto method), 188
- GetValue() (PySpin.PySpin.IEnumerationT_BlackLevelSelector method), 189
- GetValue() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrection method), 189
- GetValue() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrection method), 190
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkBlackLevel method), 191
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkCounterEvent method), 191
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkEncoderStatus method), 192
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkEncoderStatusEnums method), 193
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkExposureTimeSelector method), 193
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkGainSelectorEnums method), 194
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkImageComponentEnums method), 195
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkPixelFormatEnums method), 195
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkRegionIDEnums method), 196
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateReference method), 197
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSelector method), 198
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystem method), 198
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystem method), 199
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateTransform method), 200
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dDistanceUnitEnums method), 200
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dOutputModeEnums method), 201
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkSelectorEnums method), 202
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkSourceIDEnums method), 202
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkTimerSelectorEnums method), 203
- GetValue() (PySpin.PySpin.IEnumerationT_ChunkTransferStreamIDEnums method), 204
- GetValue() (PySpin.PySpin.IEnumerationT_ClConfigurationEnums method), 204
- GetValue() (PySpin.PySpin.IEnumerationT_ClTimeSlotsCountEnums method), 205
- GetValue() (PySpin.PySpin.IEnumerationT_ColorTransformationSelector method), 206
- GetValue() (PySpin.PySpin.IEnumerationT_ColorTransformationValueSelector method), 206
- GetValue() (PySpin.PySpin.IEnumerationT_ComponentDestinationEnums method), 207
- GetValue() (PySpin.PySpin.IEnumerationT_ComponentSelectorEnums method), 208
- GetValue() (PySpin.PySpin.IEnumerationT_CompressionSaturationPriority method), 208
- GetValue() (PySpin.PySpin.IEnumerationT_CounterEventActivationEnums method), 209
- GetValue() (PySpin.PySpin.IEnumerationT_CounterEventSourceEnums method), 210
- GetValue() (PySpin.PySpin.IEnumerationT_CounterResetActivationEnums method), 210

<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CounterReset method), 211	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceSensorChromaEnums method), 229
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CounterSelector method), 212	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums method), 230
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CounterStatus method), 212	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceSerialPortSelectorEnums method), 230
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CounterTrigger method), 213	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceStreamChannelEndianEnums method), 231
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CounterTrigger method), 214	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 232
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CxpConnection method), 214	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceTapGeometryEnums method), 233
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CxpLinkConfiguration method), 215	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceTemperatureSelectorEnums method), 234
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CxpLinkConfiguration method), 216	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceTLTypeEnums method), 232
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CxpLinkConfiguration method), 216	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceTypeEnum method), 234
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_CxpPoCxpStatus method), 217	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceTypeEnum method), 235
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DecimationHorizontal method), 218	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderModeEnums method), 236
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DecimationSelector method), 218	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderOutputModeEnums method), 236
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DecimationVertical method), 219	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderResetActivationEnums method), 237
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DefectCorrection method), 220	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderResetSourceEnums method), 238
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_Deinterlacing method), 220	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderSelectorEnums method), 238
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceAccess method), 221	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderSourceAEnums method), 239
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceCharacterization method), 222	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderSourceBEnums method), 240
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceClockSelector method), 222	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EncoderStatusEnums method), 240
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceConnection method), 223	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EventNotificationEnums method), 241
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceCurrentSelector method), 224	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_EventSelectorEnums method), 242
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceEndian method), 224	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExposureActiveModeEnums method), 242
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceIndication method), 225	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExposureAutoEnums method), 243
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceLinkHorizontal method), 226	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExposureModeEnums method), 244
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceLinkThermal method), 226	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExposureTimeModeEnums method), 244
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DevicePowerSelector method), 227	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExposureTimeSelectorEnums method), 245
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceRegistered method), 228	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_ExternalVoltageSelectorEnums method), 246
<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_DeviceScanType method), 228	<code>GetValue()</code> (PySpin.PySpin.IEnumerationT_FfcModeEnums method), 247

GetValue() (PySpin.PySpin.IEnumerationT_FileOpenModeEnums method), 248
 GetValue() (PySpin.PySpin.IEnumerationT_FileOperationEnums method), 248
 GetValue() (PySpin.PySpin.IEnumerationT_FileOperationEnums method), 249
 GetValue() (PySpin.PySpin.IEnumerationT_FileSelectorEnums method), 250
 GetValue() (PySpin.PySpin.IEnumerationT_FLIRFilterDriverEnums method), 246
 GetValue() (PySpin.PySpin.IEnumerationT_GainAutoBalanceEnums method), 251
 GetValue() (PySpin.PySpin.IEnumerationT_GainAutoEnums method), 252
 GetValue() (PySpin.PySpin.IEnumerationT_GainConversionEnums method), 252
 GetValue() (PySpin.PySpin.IEnumerationT_GainSelectorEnums method), 253
 GetValue() (PySpin.PySpin.IEnumerationT_GenICamXMLEnums method), 254
 GetValue() (PySpin.PySpin.IEnumerationT_GevCCPEnums method), 254
 GetValue() (PySpin.PySpin.IEnumerationT_GevCCPEnums method), 255
 GetValue() (PySpin.PySpin.IEnumerationT_GevCurrentPEnums method), 256
 GetValue() (PySpin.PySpin.IEnumerationT_GevGVCPEnums method), 256
 GetValue() (PySpin.PySpin.IEnumerationT_GevGVSPExtEnums method), 257
 GetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588Enums method), 258
 GetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588Enums method), 258
 GetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588Enums method), 259
 GetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588Enums method), 260
 GetValue() (PySpin.PySpin.IEnumerationT_GevIPConfigurationEnums method), 260
 GetValue() (PySpin.PySpin.IEnumerationT_GevPhysicalLengthEnums method), 261
 GetValue() (PySpin.PySpin.IEnumerationT_GevSCPDirectEnums method), 262
 GetValue() (PySpin.PySpin.IEnumerationT_GevSupportedEnums method), 262
 GetValue() (PySpin.PySpin.IEnumerationT_GUIXMLLocationEnums method), 250
 GetValue() (PySpin.PySpin.IEnumerationT_ImageCompositionEnums method), 263
 GetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionEnums method), 264
 GetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionEnums method), 264
 GetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionRateOptionsEnums method), 265
 GetValue() (PySpin.PySpin.IEnumerationT_InterfaceTypeEnum method), 266
 GetValue() (PySpin.PySpin.IEnumerationT_LensShadingCoefficientActiveEnums method), 267
 GetValue() (PySpin.PySpin.IEnumerationT_LensShadingCorrectionModeEnums method), 268
 GetValue() (PySpin.PySpin.IEnumerationT_LineFormatEnums method), 268
 GetValue() (PySpin.PySpin.IEnumerationT_LineInputFilterSelectorEnums method), 269
 GetValue() (PySpin.PySpin.IEnumerationT_LineModeEnums method), 270
 GetValue() (PySpin.PySpin.IEnumerationT_LineSelectorEnums method), 270
 GetValue() (PySpin.PySpin.IEnumerationT_LineSourceEnums method), 271
 GetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputActivationEnums method), 272
 GetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputSelectorEnums method), 272
 GetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputSourceEnums method), 273
 GetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTSelectorEnums method), 274
 GetValue() (PySpin.PySpin.IEnumerationT_LogicBlockSelectorEnums method), 274
 GetValue() (PySpin.PySpin.IEnumerationT_LUTSelectorEnums method), 266
 GetValue() (PySpin.PySpin.IEnumerationT_MultiRoiConfigurationInvalidEnums method), 275
 GetValue() (PySpin.PySpin.IEnumerationT_MultiRoiSelectorEnums method), 276
 GetValue() (PySpin.PySpin.IEnumerationT_PixelColorFilterEnums method), 277
 GetValue() (PySpin.PySpin.IEnumerationT_PixelFormatEnums method), 278
 GetValue() (PySpin.PySpin.IEnumerationT_PixelFormatInfoSelectorEnums method), 278
 GetValue() (PySpin.PySpin.IEnumerationT_PixelSizeEnums method), 279
 GetValue() (PySpin.PySpin.IEnumerationT_POEStatusEnum method), 276
 GetValue() (PySpin.PySpin.IEnumerationT_RegionDestinationEnums method), 280
 GetValue() (PySpin.PySpin.IEnumerationT_RegionModeEnums method), 280
 GetValue() (PySpin.PySpin.IEnumerationT_RegionSelectorEnums method), 281
 GetValue() (PySpin.PySpin.IEnumerationT_RgbTransformLightSourceEnums method), 282
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinateReferenceEnums method), 282

GetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinationEnums method), 283
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnums method), 284
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnums method), 284
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnums method), 285
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dDistanceEnums method), 286
 GetValue() (PySpin.PySpin.IEnumerationT_Scan3dOutputEnums method), 286
 GetValue() (PySpin.PySpin.IEnumerationT_SensorDigitizationEnums method), 287
 GetValue() (PySpin.PySpin.IEnumerationT_SensorShutterEnums method), 288
 GetValue() (PySpin.PySpin.IEnumerationT_SensorTapsEnums method), 288
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerConfigurationEnums method), 289
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerConfigurationEnums method), 290
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerModeEnums method), 290
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerSetEnums method), 291
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerTriggerEnums method), 292
 GetValue() (PySpin.PySpin.IEnumerationT_SequencerTriggerEnums method), 292
 GetValue() (PySpin.PySpin.IEnumerationT_SerialPortBaudRateEnums method), 293
 GetValue() (PySpin.PySpin.IEnumerationT_SerialPortParametersEnums method), 294
 GetValue() (PySpin.PySpin.IEnumerationT_SerialPortSelectorEnums method), 294
 GetValue() (PySpin.PySpin.IEnumerationT_SerialPortSourceEnums method), 295
 GetValue() (PySpin.PySpin.IEnumerationT_SerialPortStopEnums method), 296
 GetValue() (PySpin.PySpin.IEnumerationT_SoftwareSignalEnums method), 296
 GetValue() (PySpin.PySpin.IEnumerationT_SourceSelectorEnums method), 297
 GetValue() (PySpin.PySpin.IEnumerationT_StereoResolutionEnums method), 298
 GetValue() (PySpin.PySpin.IEnumerationT_StreamBufferEnums method), 298
 GetValue() (PySpin.PySpin.IEnumerationT_StreamBufferEnums method), 299
 GetValue() (PySpin.PySpin.IEnumerationT_StreamModeEnums method), 300
 GetValue() (PySpin.PySpin.IEnumerationT_StreamTypeEnums method), 300
 GetValue() (PySpin.PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverEnums method), 302
 GetValue() (PySpin.PySpin.IEnumerationT_TestPatternEnums method), 302
 GetValue() (PySpin.PySpin.IEnumerationT_TestPatternGeneratorSelectorEnums method), 303
 GetValue() (PySpin.PySpin.IEnumerationT_TimerSelectorEnums method), 304
 GetValue() (PySpin.PySpin.IEnumerationT_TimerStatusEnums method), 304
 GetValue() (PySpin.PySpin.IEnumerationT_TimerTriggerActivationEnums method), 305
 GetValue() (PySpin.PySpin.IEnumerationT_TimerTriggerSourceEnums method), 306
 GetValue() (PySpin.PySpin.IEnumerationT_TLTypeEnum method), 301
 GetValue() (PySpin.PySpin.IEnumerationT_TransferComponentSelectorEnums method), 306
 GetValue() (PySpin.PySpin.IEnumerationT_TransferControlModeEnums method), 307
 GetValue() (PySpin.PySpin.IEnumerationT_TransferOperationModeEnums method), 308
 GetValue() (PySpin.PySpin.IEnumerationT_TransferQueueModeEnums method), 308
 GetValue() (PySpin.PySpin.IEnumerationT_TransferSelectorEnums method), 309
 GetValue() (PySpin.PySpin.IEnumerationT_TransferStatusSelectorEnums method), 310
 GetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerActivationEnums method), 310
 GetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerModeEnums method), 311
 GetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerSelectorEnums method), 312
 GetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerSourceEnums method), 312
 GetValue() (PySpin.PySpin.IEnumerationT_TriggerActivationEnums method), 313
 GetValue() (PySpin.PySpin.IEnumerationT_TriggerModeEnums method), 314
 GetValue() (PySpin.PySpin.IEnumerationT_TriggerOverlapEnums method), 314
 GetValue() (PySpin.PySpin.IEnumerationT_TriggerSelectorEnums method), 315
 GetValue() (PySpin.PySpin.IEnumerationT_TriggerSourceEnums method), 316
 GetValue() (PySpin.PySpin.IEnumerationT_U3VCurrentSpeedEnums method), 316
 GetValue() (PySpin.PySpin.IEnumerationT_UserOutputSelectorEnums method), 317
 GetValue() (PySpin.PySpin.IEnumerationT_UserSetDefaultEnums method), 318
 GetValue() (PySpin.PySpin.IEnumerationT_UserSetSelectorEnums method), 318

- GetValue() (PySpin.PySpin.IEnumerationT_WhiteClipSelect method), 319
- GetValue() (PySpin.PySpin.IFloat method), 320
- GetValue() (PySpin.PySpin.IInteger method), 327
- GetValue() (PySpin.PySpin.IntegerNode method), 362
- GetValue() (PySpin.PySpin.IString method), 337
- GetValue() (PySpin.PySpin.StringNode method), 389
- GetValueOfEnvironmentVariable() (in module PySpin.PySpin), 167
- GetVendorName() (PySpin.PySpin.CDeviceInfoPtr method), 91
- GetVendorName() (PySpin.PySpin.IDeviceInfo method), 175
- GetVendorName() (PySpin.PySpin.NodeMap method), 380
- GetVersion() (PySpin.PySpin.InferenceBoundingBoxResult method), 360
- GetVersionGuid() (PySpin.PySpin.CDeviceInfoPtr method), 91
- GetVersionGuid() (PySpin.PySpin.IDeviceInfo method), 175
- GetVersionGuid() (PySpin.PySpin.NodeMap method), 380
- GetVisibility() (PySpin.PySpin.CBooleanPtr method), 84
- GetVisibility() (PySpin.PySpin.CCategoryPtr method), 87
- GetVisibility() (PySpin.PySpin.CCommandPtr method), 90
- GetVisibility() (PySpin.PySpin.CEnumEntryPtr method), 93
- GetVisibility() (PySpin.PySpin.CEnumerationPtr method), 96
- GetVisibility() (PySpin.PySpin.CIntegerPtr method), 100
- GetVisibility() (PySpin.PySpin.CNodePtr method), 105
- GetVisibility() (PySpin.PySpin.CRegisterPtr method), 108
- GetVisibility() (PySpin.PySpin.CStringPtr method), 112
- GetVisibility() (PySpin.PySpin.CValuePtr method), 114
- GetVisibility() (PySpin.PySpin.INode method), 331
- GetVisibility() (PySpin.PySpin.Node method), 376
- GetWidth() (PySpin.ChunkData method), 45
- GetWidth() (PySpin.Image method), 50
- GetWidth() (PySpin.PySpin.ChunkData method), 151
- GetWidth() (PySpin.PySpin.IChunkData method), 173
- GetWidth() (PySpin.PySpin.IImage method), 321
- GetWidth() (PySpin.PySpin.Image method), 344
- GetXOffset() (PySpin.Image method), 51
- GetXOffset() (PySpin.PySpin.IImage method), 322
- GetXOffset() (PySpin.PySpin.Image method), 344
- GetYOffset() (PySpin.Image method), 51
- GetYOffset() (PySpin.PySpin.IImage method), 322
- GetYOffset() (PySpin.PySpin.Image method), 344
- GetYPadding() (PySpin.Image method), 51
- GetYPadding() (PySpin.PySpin.IImage method), 322
- GetYPadding() (PySpin.PySpin.Image method), 344
- GevActionAckRequired (PySpin.PySpin.TransportLayerInterface property), 398
- GevActionAckRequired (PySpin.TransportLayerInterface property), 77
- GevActionDeviceKey (PySpin.PySpin.TransportLayerInterface property), 398
- GevActionDeviceKey (PySpin.TransportLayerInterface property), 77
- GevActionGroupKey (PySpin.PySpin.TransportLayerInterface property), 398
- GevActionGroupKey (PySpin.TransportLayerInterface property), 77
- GevActionGroupMask (PySpin.PySpin.TransportLayerInterface property), 398
- GevActionGroupMask (PySpin.TransportLayerInterface property), 77
- GevActionTime (PySpin.PySpin.TransportLayerInterface property), 398
- GevActionTime (PySpin.TransportLayerInterface property), 77
- GevActiveLinkCount (PySpin.Camera property), 25
- GevActiveLinkCount (PySpin.PySpin.Camera property), 130
- GEVAutoAssignIPEnable (PySpin.PySpin.TransportLayerSystem property), 401
- GevCCP (PySpin.Camera property), 25
- GevCCP (PySpin.PySpin.Camera property), 130
- GevCCP (PySpin.PySpin.TransportLayerDevice property), 397
- GevCCP (PySpin.TransportLayerDevice property), 76
- GevCurrentDefaultGateway (PySpin.Camera property), 25
- GevCurrentDefaultGateway (PySpin.PySpin.Camera property), 130
- GevCurrentIPAddress (PySpin.Camera property), 25
- GevCurrentIPAddress (PySpin.PySpin.Camera property), 130
- GevCurrentIPConfigurationDHCP (PySpin.Camera property), 25
- GevCurrentIPConfigurationDHCP (PySpin.PySpin.Camera property), 130
- GevCurrentIPConfigurationLLA (PySpin.Camera property), 25

GevCurrentIPConfigurationLLA (<i>PySpin.PySpin.Camera</i> property), 130	GevDeviceForceGateway (<i>PySpin.TransportLayerInterface</i> property), 77
GevCurrentIPConfigurationPersistentIP (<i>PySpin.Camera</i> property), 25	GevDeviceForceIP (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
GevCurrentIPConfigurationPersistentIP (<i>PySpin.PySpin.Camera</i> property), 130	GevDeviceForceIP (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398
GevCurrentPhysicalLinkConfiguration (<i>PySpin.Camera</i> property), 25	GevDeviceForceIP (<i>PySpin.TransportLayerDevice</i> property), 76
GevCurrentPhysicalLinkConfiguration (<i>PySpin.PySpin.Camera</i> property), 130	GevDeviceForceIP (<i>PySpin.TransportLayerInterface</i> property), 77
GevCurrentSubnetMask (<i>PySpin.Camera</i> property), 25	GevDeviceForceIPAddress (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
GevCurrentSubnetMask (<i>PySpin.PySpin.Camera</i> property), 130	GevDeviceForceIPAddress (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398
GevDeviceAutoForceIP (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevDeviceForceIPAddress (<i>PySpin.TransportLayerDevice</i> property), 76
GevDeviceAutoForceIP (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevDeviceForceIPAddress (<i>PySpin.TransportLayerInterface</i> property), 77
GevDeviceAutoForceIP (<i>PySpin.TransportLayerDevice</i> property), 76	GevDeviceForceSubnetMask (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
GevDeviceAutoForceIP (<i>PySpin.TransportLayerInterface</i> property), 77	GevDeviceForceSubnetMask (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398
GevDeviceDisableDiscovery (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevDeviceForceSubnetMask (<i>PySpin.TransportLayerDevice</i> property), 76
GevDeviceDisableDiscovery (<i>PySpin.TransportLayerInterface</i> property), 77	GevDeviceForceSubnetMask (<i>PySpin.TransportLayerInterface</i> property), 77
GevDeviceDiscoverMaximumPacketSize (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevDeviceGateway (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
GevDeviceDiscoverMaximumPacketSize (<i>PySpin.TransportLayerDevice</i> property), 76	GevDeviceGateway (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398
GevDeviceDiscoveryEnabled (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevDeviceGateway (<i>PySpin.TransportLayerDevice</i> property), 76
GevDeviceDiscoveryEnabled (<i>PySpin.TransportLayerInterface</i> property), 77	GevDeviceGateway (<i>PySpin.TransportLayerInterface</i> property), 77
GevDeviceEnabledDiscovery (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevDeviceIPAddress (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
GevDeviceEnabledDiscovery (<i>PySpin.TransportLayerInterface</i> property), 77	GevDeviceIPAddress (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398
GevDeviceForceGateway (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevDeviceIPAddress (<i>PySpin.TransportLayerDevice</i> property), 76
GevDeviceForceGateway (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevDeviceIPAddress (<i>PySpin.TransportLayerInterface</i> property), 77
GevDeviceForceGateway (<i>PySpin.TransportLayerDevice</i> property), 76	GevDeviceIsWrongSubnet (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
	GevDeviceIsWrongSubnet (<i>PySpin.TransportLayerDevice</i> property), 76

GevDeviceMACAddress (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevGVCPExtendedStatusCodes (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceMACAddress (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevGVCPExtendedStatusCodesSelector (<i>PySpin.Camera</i> property), 25
GevDeviceMACAddress (<i>PySpin.TransportLayerDevice</i> property), 76	GevGVCPExtendedStatusCodesSelector (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceMACAddress (<i>PySpin.TransportLayerInterface</i> property), 77	GevGVCPHeartbeatDisable (<i>PySpin.Camera</i> property), 25
GevDeviceMaximumPacketSize (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevGVCPHeartbeatDisable (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceMaximumPacketSize (<i>PySpin.TransportLayerDevice</i> property), 76	GevGVCPPendingAck (<i>PySpin.Camera</i> property), 25
GevDeviceMaximumRetryCount (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevGVCPPendingAck (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceMaximumRetryCount (<i>PySpin.TransportLayerDevice</i> property), 76	GevGVCPPendingTimeout (<i>PySpin.Camera</i> property), 25
GevDeviceModeIsBigEndian (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevGVCPPendingTimeout (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceModeIsBigEndian (<i>PySpin.TransportLayerDevice</i> property), 76	GevGVSPExtendedIDMode (<i>PySpin.Camera</i> property), 25
GevDevicePort (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevGVSPExtendedIDMode (<i>PySpin.PySpin.Camera</i> property), 131
GevDevicePort (<i>PySpin.TransportLayerDevice</i> property), 76	GevHeartbeatTimeout (<i>PySpin.Camera</i> property), 25
GevDeviceReadAndWriteTimeout (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevHeartbeatTimeout (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceReadAndWriteTimeout (<i>PySpin.TransportLayerDevice</i> property), 76	GevIEEE1588 (<i>PySpin.Camera</i> property), 25
GevDeviceSubnetMask (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397	GevIEEE1588 (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceSubnetMask (<i>PySpin.PySpin.TransportLayerInterface</i> property), 398	GevIEEE1588ClockAccuracy (<i>PySpin.Camera</i> property), 25
GevDeviceSubnetMask (<i>PySpin.TransportLayerDevice</i> property), 76	GevIEEE1588ClockAccuracy (<i>PySpin.PySpin.Camera</i> property), 131
GevDeviceSubnetMask (<i>PySpin.TransportLayerInterface</i> property), 77	GevIEEE1588ClockId (<i>PySpin.Camera</i> property), 25
GevDiscoveryAckDelay (<i>PySpin.Camera</i> property), 25	GevIEEE1588ClockId (<i>PySpin.PySpin.Camera</i> property), 131
GevDiscoveryAckDelay (<i>PySpin.PySpin.Camera</i> property), 131	GevIEEE1588DataSetLatch (<i>PySpin.Camera</i> property), 25
GevFirstURL (<i>PySpin.Camera</i> property), 25	GevIEEE1588DataSetLatch (<i>PySpin.PySpin.Camera</i> property), 131
GevFirstURL (<i>PySpin.PySpin.Camera</i> property), 131	GevIEEE1588Mode (<i>PySpin.Camera</i> property), 26
GevGVCPExtendedStatusCodes (<i>PySpin.Camera</i> property), 25	GevIEEE1588Mode (<i>PySpin.PySpin.Camera</i> property), 131
	GevIEEE1588OffsetFromMasterLatched (<i>PySpin.Camera</i> property), 26
	GevIEEE1588OffsetFromMasterLatched (<i>PySpin.PySpin.Camera</i> property), 131
	GevIEEE1588ParentClockIdLatched (<i>PySpin.Camera</i> property), 26
	GevIEEE1588ParentClockIdLatched (<i>PySpin.PySpin.Camera</i> property), 131
	GevIEEE1588Status (<i>PySpin.Camera</i> property), 26
	GevIEEE1588Status (<i>PySpin.PySpin.Camera</i> property), 131
	GevIEEE1588StatusLatched (<i>PySpin.Camera</i> property), 26

GevIEEE1588StatusLatched (<i>PySpin.PySpin.Camera</i> property), 131	GevInterfaceSubnetMask (<i>PySpin.TransportLayerInterface</i> property), 78
GevInterfaceDefaultGateway (<i>PySpin.PySpin.TransportLayerSystem</i> property), 401	GevInterfaceSubnetSelector (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399
GevInterfaceDefaultIPAddress (<i>PySpin.PySpin.TransportLayerSystem</i> property), 402	GevInterfaceSubnetSelector (<i>PySpin.TransportLayerInterface</i> property), 78
GevInterfaceDefaultSubnetMask (<i>PySpin.PySpin.TransportLayerSystem</i> property), 402	GevInterfaceTransmitLinkSpeed (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399
GevInterfaceGateway (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevInterfaceTransmitLinkSpeed (<i>PySpin.TransportLayerInterface</i> property), 78
GevInterfaceGateway (<i>PySpin.TransportLayerInterface</i> property), 78	GevIPConfigurationStatus (<i>PySpin.Camera</i> property), 26
GevInterfaceGatewaySelector (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevIPConfigurationStatus (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceGatewaySelector (<i>PySpin.TransportLayerInterface</i> property), 78	GevMACAddress (<i>PySpin.Camera</i> property), 26
GevInterfaceIsIPConflict (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevMACAddress (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceIsIPConflict (<i>PySpin.TransportLayerInterface</i> property), 78	GevMCDA (<i>PySpin.Camera</i> property), 26
GevInterfaceMACAddress (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevMCDA (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceMACAddress (<i>PySpin.PySpin.TransportLayerSystem</i> property), 402	GevMCPHostPort (<i>PySpin.Camera</i> property), 26
GevInterfaceMACAddress (<i>PySpin.TransportLayerInterface</i> property), 78	GevMCPHostPort (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceMTU (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevMCRC (<i>PySpin.Camera</i> property), 26
GevInterfaceMTU (<i>PySpin.TransportLayerInterface</i> property), 78	GevMCRC (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceReceiveLinkSpeed (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevMCSP (<i>PySpin.Camera</i> property), 26
GevInterfaceReceiveLinkSpeed (<i>PySpin.TransportLayerInterface</i> property), 78	GevMCSP (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceSelector (<i>PySpin.Camera</i> property), 26	GevMCTT (<i>PySpin.Camera</i> property), 26
GevInterfaceSelector (<i>PySpin.PySpin.Camera</i> property), 131	GevMCTT (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceSubnetIPAddress (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevNumberOfActiveLinks (<i>PySpin.Camera</i> property), 26
GevInterfaceSubnetIPAddress (<i>PySpin.TransportLayerInterface</i> property), 78	GevNumberOfActiveLinks (<i>PySpin.PySpin.Camera</i> property), 131
GevInterfaceSubnetMask (<i>PySpin.PySpin.TransportLayerInterface</i> property), 399	GevNumberOfInterfaces (<i>PySpin.Camera</i> property), 26
	GevNumberOfInterfaces (<i>PySpin.PySpin.Camera</i> property), 131
	GevPAUSEFrameReception (<i>PySpin.Camera</i> property), 26
	GevPAUSEFrameReception (<i>PySpin.PySpin.Camera</i> property), 131
	GevPAUSEFrameTransmission (<i>PySpin.Camera</i> property), 26
	GevPAUSEFrameTransmission (<i>PySpin.PySpin.Camera</i> property), 131
	GevPersistentDefaultGateway (<i>PySpin.Camera</i> property), 26
	GevPersistentDefaultGateway (<i>PySpin.PySpin.Camera</i> property), 131
	GevPersistentIPAddress (<i>PySpin.Camera</i> property), 26
	GevPersistentIPAddress (<i>PySpin.PySpin.Camera</i> property), 132
	GevPersistentSubnetMask (<i>PySpin.Camera</i> property), 26

- erty), 26
- GevPersistentSubnetMask (PySpin.PySpin.Camera property), 132
- GevPhysicalLinkConfiguration (PySpin.Camera property), 26
- GevPhysicalLinkConfiguration (PySpin.PySpin.Camera property), 132
- GevPhysicalLinkConfigurationCapability (PySpin.Camera property), 26
- GevPhysicalLinkConfigurationCapability (PySpin.PySpin.Camera property), 132
- GevPrimaryApplicationIPAddress (PySpin.Camera property), 26
- GevPrimaryApplicationIPAddress (PySpin.PySpin.Camera property), 132
- GevPrimaryApplicationSocket (PySpin.Camera property), 26
- GevPrimaryApplicationSocket (PySpin.PySpin.Camera property), 132
- GevPrimaryApplicationSwitchoverKey (PySpin.Camera property), 26
- GevPrimaryApplicationSwitchoverKey (PySpin.PySpin.Camera property), 132
- GevSCCFGAllInTransmission (PySpin.Camera property), 26
- GevSCCFGAllInTransmission (PySpin.PySpin.Camera property), 132
- GevSCCFGExtendedChunkData (PySpin.Camera property), 26
- GevSCCFGExtendedChunkData (PySpin.PySpin.Camera property), 132
- GevSCCFGPacketResendDestination (PySpin.Camera property), 26
- GevSCCFGPacketResendDestination (PySpin.PySpin.Camera property), 132
- GevSCCFGUnconditionalStreaming (PySpin.Camera property), 26
- GevSCCFGUnconditionalStreaming (PySpin.PySpin.Camera property), 132
- GevSCDA (PySpin.Camera property), 26
- GevSCDA (PySpin.PySpin.Camera property), 132
- GevSCPD (PySpin.Camera property), 26
- GevSCPD (PySpin.PySpin.Camera property), 132
- GevSCPDDirection (PySpin.Camera property), 27
- GevSCPDDirection (PySpin.PySpin.Camera property), 132
- GevSCPHostPort (PySpin.Camera property), 27
- GevSCPHostPort (PySpin.PySpin.Camera property), 132
- GevSCPIInterfaceIndex (PySpin.Camera property), 27
- GevSCPIInterfaceIndex (PySpin.PySpin.Camera property), 132
- GevSCPSBigEndian (PySpin.Camera property), 27
- GevSCPSBigEndian (PySpin.PySpin.Camera property), 132
- GevSCPSDoNotFragment (PySpin.Camera property), 27
- GevSCPSDoNotFragment (PySpin.PySpin.Camera property), 132
- GevSCPSFireTestPacket (PySpin.Camera property), 27
- GevSCPSFireTestPacket (PySpin.PySpin.Camera property), 132
- GevSCPSPacketSize (PySpin.Camera property), 27
- GevSCPSPacketSize (PySpin.PySpin.Camera property), 132
- GevSCSP (PySpin.Camera property), 27
- GevSCSP (PySpin.PySpin.Camera property), 132
- GevSCZoneConfigurationLock (PySpin.Camera property), 27
- GevSCZoneConfigurationLock (PySpin.PySpin.Camera property), 132
- GevSCZoneCount (PySpin.Camera property), 27
- GevSCZoneCount (PySpin.PySpin.Camera property), 132
- GevSCZoneDirectionAll (PySpin.Camera property), 27
- GevSCZoneDirectionAll (PySpin.PySpin.Camera property), 132
- GevSecondURL (PySpin.Camera property), 27
- GevSecondURL (PySpin.PySpin.Camera property), 132
- GevStreamChannelSelector (PySpin.Camera property), 27
- GevStreamChannelSelector (PySpin.PySpin.Camera property), 132
- GevSupportedOption (PySpin.Camera property), 27
- GevSupportedOption (PySpin.PySpin.Camera property), 132
- GevSupportedOptionSelector (PySpin.Camera property), 27
- GevSupportedOptionSelector (PySpin.PySpin.Camera property), 132
- GevTimestampTickFrequency (PySpin.Camera property), 27
- GevTimestampTickFrequency (PySpin.PySpin.Camera property), 132
- GevVersionMajor (PySpin.PySpin.TransportLayerDevice property), 397
- GevVersionMajor (PySpin.PySpin.TransportLayerSystem property), 402
- GevVersionMajor (PySpin.TransportLayerDevice property), 76
- GevVersionMinor (PySpin.PySpin.TransportLayerDevice property), 397
- GevVersionMinor (PySpin.PySpin.TransportLayerSystem property), 402
- GevVersionMinor (PySpin.TransportLayerDevice property), 76
- GUIXMLLocation (PySpin.PySpin.TransportLayerDevice

property), 397
 GUIXMLLocation (PySpin.TransportLayerDevice property), 76
 GuiXmlManifestAddress (PySpin.Camera property), 27
 GuiXmlManifestAddress (PySpin.PySpin.Camera property), 132
 GUIXMLPath (PySpin.PySpin.TransportLayerDevice property), 397
 GUIXMLPath (PySpin.TransportLayerDevice property), 76

H

H264Option (class in PySpin.PySpin), 167
 HasChunkData() (PySpin.Image method), 51
 HasChunkData() (PySpin.PySpin.IImage method), 322
 HasChunkData() (PySpin.PySpin.Image method), 345
 HasCRC() (PySpin.Image method), 51
 HasCRC() (PySpin.PySpin.IImage method), 322
 HasCRC() (PySpin.PySpin.Image method), 345
 HasInc() (PySpin.PySpin.FloatNode method), 165
 HasInc() (PySpin.PySpin.IFloat method), 320
 Height (PySpin.Camera property), 27
 height (PySpin.PySpin.AVIOption property), 81
 Height (PySpin.PySpin.Camera property), 132
 height (PySpin.PySpin.H264Option property), 168
 height (PySpin.PySpin.MJPGOption property), 372
 HeightMax (PySpin.Camera property), 27
 HeightMax (PySpin.PySpin.Camera property), 133
 histogram (PySpin.ChannelStatistics property), 42
 histogram (PySpin.PySpin.ChannelStatistics property), 148
 HostAdapterDriverVersion (PySpin.PySpin.TransportLayerInterface property), 399
 HostAdapterDriverVersion (PySpin.TransportLayerInterface property), 78
 HostAdapterName (PySpin.PySpin.TransportLayerInterface property), 399
 HostAdapterName (PySpin.TransportLayerInterface property), 78
 HostAdapterVendor (PySpin.PySpin.TransportLayerInterface property), 399
 HostAdapterVendor (PySpin.TransportLayerInterface property), 78

I

IBase (class in PySpin.PySpin), 168
 IBoolean (class in PySpin.PySpin), 168
 ICameraBase (class in PySpin.PySpin), 168
 ICameraList (class in PySpin.PySpin), 171
 ICategory (class in PySpin.PySpin), 171
 IChunkData (class in PySpin.PySpin), 172
 ICommand (class in PySpin.PySpin), 173

IDestroy (class in PySpin.PySpin), 174
 IDeviceArrivalEventHandler (class in PySpin.PySpin), 174
 IDeviceEventHandler (class in PySpin.PySpin), 174
 IDeviceInfo (class in PySpin.PySpin), 174
 IDeviceRemovalEventHandler (class in PySpin.PySpin), 175
 IEnumEntry (class in PySpin.PySpin), 175
 IEnumeration (class in PySpin.PySpin), 176
 IEnumerationT_AcquisitionModeEnums (class in PySpin.PySpin), 176
 IEnumerationT_AcquisitionStatusSelectorEnums (class in PySpin.PySpin), 177
 IEnumerationT_ActionSelectorEnums (class in PySpin.PySpin), 178
 IEnumerationT_ActionUnconditionalModeEnums (class in PySpin.PySpin), 178
 IEnumerationT_AdcBitDepthEnums (class in PySpin.PySpin), 179
 IEnumerationT_AutoAlgorithmSelectorEnums (class in PySpin.PySpin), 180
 IEnumerationT_AutoExposureControlPriorityEnums (class in PySpin.PySpin), 180
 IEnumerationT_AutoExposureLightingModeEnums (class in PySpin.PySpin), 181
 IEnumerationT_AutoExposureMeteringModeEnums (class in PySpin.PySpin), 182
 IEnumerationT_AutoExposureTargetGreyValueAutoEnums (class in PySpin.PySpin), 182
 IEnumerationT_BalanceRatioSelectorEnums (class in PySpin.PySpin), 183
 IEnumerationT_BalanceWhiteAutoEnums (class in PySpin.PySpin), 184
 IEnumerationT_BalanceWhiteAutoProfileEnums (class in PySpin.PySpin), 184
 IEnumerationT_BinningHorizontalModeEnums (class in PySpin.PySpin), 185
 IEnumerationT_BinningSelectorEnums (class in PySpin.PySpin), 186
 IEnumerationT_BinningVerticalModeEnums (class in PySpin.PySpin), 186
 IEnumerationT_BlackLevelAutoBalanceEnums (class in PySpin.PySpin), 187
 IEnumerationT_BlackLevelAutoEnums (class in PySpin.PySpin), 188
 IEnumerationT_BlackLevelSelectorEnums (class in PySpin.PySpin), 188
 IEnumerationT_BsiFlatFieldCorrectionAutoEnums (class in PySpin.PySpin), 189
 IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums (class in PySpin.PySpin), 190
 IEnumerationT_ChunkBlackLevelSelectorEnums (class in PySpin.PySpin), 190
 IEnumerationT_ChunkCounterSelectorEnums (class

- in PySpin.PySpin*), 191
- `IEnumerationT_ChunkEncoderSelectorEnums` (class *in PySpin.PySpin*), 192
- `IEnumerationT_ChunkEncoderStatusEnums` (class *in PySpin.PySpin*), 192
- `IEnumerationT_ChunkExposureTimeSelectorEnums` (class *in PySpin.PySpin*), 193
- `IEnumerationT_ChunkGainSelectorEnums` (class *in PySpin.PySpin*), 194
- `IEnumerationT_ChunkImageComponentEnums` (class *in PySpin.PySpin*), 194
- `IEnumerationT_ChunkPixelFormatEnums` (class *in PySpin.PySpin*), 195
- `IEnumerationT_ChunkRegionIDEnums` (class *in PySpin.PySpin*), 196
- `IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums` (class *in PySpin.PySpin*), 196
- `IEnumerationT_ChunkScan3dCoordinateSelectorEnums` (class *in PySpin.PySpin*), 197
- `IEnumerationT_ChunkScan3dCoordinateSystemEnums` (class *in PySpin.PySpin*), 198
- `IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums` (class *in PySpin.PySpin*), 198
- `IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums` (class *in PySpin.PySpin*), 199
- `IEnumerationT_ChunkScan3dDistanceUnitEnums` (class *in PySpin.PySpin*), 200
- `IEnumerationT_ChunkScan3dOutputModeEnums` (class *in PySpin.PySpin*), 201
- `IEnumerationT_ChunkSelectorEnums` (class *in PySpin.PySpin*), 201
- `IEnumerationT_ChunkSourceIDEnums` (class *in PySpin.PySpin*), 202
- `IEnumerationT_ChunkTimerSelectorEnums` (class *in PySpin.PySpin*), 203
- `IEnumerationT_ChunkTransferStreamIDEnums` (class *in PySpin.PySpin*), 203
- `IEnumerationT_ClConfigurationEnums` (class *in PySpin.PySpin*), 204
- `IEnumerationT_ClTimeSlotsCountEnums` (class *in PySpin.PySpin*), 205
- `IEnumerationT_ColorTransformationSelectorEnums` (class *in PySpin.PySpin*), 205
- `IEnumerationT_ColorTransformationValueSelectorEnums` (class *in PySpin.PySpin*), 206
- `IEnumerationT_ComponentDestinationEnums` (class *in PySpin.PySpin*), 207
- `IEnumerationT_ComponentSelectorEnums` (class *in PySpin.PySpin*), 207
- `IEnumerationT_CompressionSaturationPriorityEnums` (class *in PySpin.PySpin*), 208
- `IEnumerationT_CounterEventActivationEnums` (class *in PySpin.PySpin*), 209
- `IEnumerationT_CounterEventSourceEnums` (class *in PySpin.PySpin*), 209
- `IEnumerationT_CounterResetActivationEnums` (class *in PySpin.PySpin*), 210
- `IEnumerationT_CounterResetSourceEnums` (class *in PySpin.PySpin*), 211
- `IEnumerationT_CounterSelectorEnums` (class *in PySpin.PySpin*), 211
- `IEnumerationT_CounterStatusEnums` (class *in PySpin.PySpin*), 212
- `IEnumerationT_CounterTriggerActivationEnums` (class *in PySpin.PySpin*), 213
- `IEnumerationT_CounterTriggerSourceEnums` (class *in PySpin.PySpin*), 213
- `IEnumerationT_CxpConnectionTestModeEnums` (class *in PySpin.PySpin*), 214
- `IEnumerationT_CxpLinkConfigurationEnums` (class *in PySpin.PySpin*), 215
- `IEnumerationT_CxpLinkConfigurationPreferredEnums` (class *in PySpin.PySpin*), 215
- `IEnumerationT_CxpLinkConfigurationStatusEnums` (class *in PySpin.PySpin*), 216
- `IEnumerationT_CxpPoCxpStatusEnums` (class *in PySpin.PySpin*), 217
- `IEnumerationT_DecimationHorizontalModeEnums` (class *in PySpin.PySpin*), 217
- `IEnumerationT_DecimationSelectorEnums` (class *in PySpin.PySpin*), 218
- `IEnumerationT_DecimationVerticalModeEnums` (class *in PySpin.PySpin*), 219
- `IEnumerationT_DefectCorrectionModeEnums` (class *in PySpin.PySpin*), 219
- `IEnumerationT_DeinterlacingEnums` (class *in PySpin.PySpin*), 220
- `IEnumerationT_DeviceAccessStatusEnum` (class *in PySpin.PySpin*), 221
- `IEnumerationT_DeviceCharacterSetEnums` (class *in PySpin.PySpin*), 221
- `IEnumerationT_DeviceClockSelectorEnums` (class *in PySpin.PySpin*), 222
- `IEnumerationT_DeviceConnectionStatusEnums` (class *in PySpin.PySpin*), 223
- `IEnumerationT_DeviceCurrentSpeedEnum` (class *in PySpin.PySpin*), 223
- `IEnumerationT_DeviceEndiannessMechanismEnum` (class *in PySpin.PySpin*), 224
- `IEnumerationT_DeviceIndicatorModeEnums` (class *in PySpin.PySpin*), 225
- `IEnumerationT_DeviceLinkHeartbeatModeEnums` (class *in PySpin.PySpin*), 225
- `IEnumerationT_DeviceLinkThroughputLimitModeEnums` (class *in PySpin.PySpin*), 226
- `IEnumerationT_DevicePowerSupplySelectorEnums` (class *in PySpin.PySpin*), 227
- `IEnumerationT_DeviceRegistersEndiannessEnums`

<i>(class in PySpin.PySpin)</i> , 227	<i>(class in PySpin.PySpin)</i> , 245
IEnumerationT_DeviceScanTypeEnums <i>(class in PySpin.PySpin)</i> , 228	IEnumerationT_FfcModeEnums <i>(class in PySpin.PySpin)</i> , 247
IEnumerationT_DeviceSensorChromaEnums <i>(class in PySpin.PySpin)</i> , 229	IEnumerationT_FileOpenModeEnums <i>(class in PySpin.PySpin)</i> , 247
IEnumerationT_DeviceSerialPortBaudRateEnums <i>(class in PySpin.PySpin)</i> , 229	IEnumerationT_FileOperationSelectorEnums <i>(class in PySpin.PySpin)</i> , 248
IEnumerationT_DeviceSerialPortSelectorEnums <i>(class in PySpin.PySpin)</i> , 230	IEnumerationT_FileOperationStatusEnums <i>(class in PySpin.PySpin)</i> , 249
IEnumerationT_DeviceStreamChannelEndiannessEnums <i>(class in PySpin.PySpin)</i> , 231	IEnumerationT_FileSelectorEnums <i>(class in PySpin.PySpin)</i> , 249
IEnumerationT_DeviceStreamChannelTypeEnums <i>(class in PySpin.PySpin)</i> , 231	IEnumerationT_FLIRFilterDriverStatusEnum <i>(class in PySpin.PySpin)</i> , 246
IEnumerationT_DeviceTapGeometryEnums <i>(class in PySpin.PySpin)</i> , 233	IEnumerationT_GainAutoBalanceEnums <i>(class in PySpin.PySpin)</i> , 251
IEnumerationT_DeviceTemperatureSelectorEnums <i>(class in PySpin.PySpin)</i> , 233	IEnumerationT_GainAutoEnums <i>(class in PySpin.PySpin)</i> , 251
IEnumerationT_DeviceTLTypeEnums <i>(class in PySpin.PySpin)</i> , 232	IEnumerationT_GainConversionEnums <i>(class in PySpin.PySpin)</i> , 252
IEnumerationT_DeviceTypeEnum <i>(class in PySpin.PySpin)</i> , 234	IEnumerationT_GainSelectorEnums <i>(class in PySpin.PySpin)</i> , 253
IEnumerationT_DeviceTypeEnums <i>(class in PySpin.PySpin)</i> , 235	IEnumerationT_GenICamXMLLocationEnum <i>(class in PySpin.PySpin)</i> , 253
IEnumerationT_EncoderModeEnums <i>(class in PySpin.PySpin)</i> , 235	IEnumerationT_GevCCPEnum <i>(class in PySpin.PySpin)</i> , 254
IEnumerationT_EncoderOutputModeEnums <i>(class in PySpin.PySpin)</i> , 236	IEnumerationT_GevCCPEnums <i>(class in PySpin.PySpin)</i> , 255
IEnumerationT_EncoderResetActivationEnums <i>(class in PySpin.PySpin)</i> , 237	IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums <i>(class in PySpin.PySpin)</i> , 255
IEnumerationT_EncoderResetSourceEnums <i>(class in PySpin.PySpin)</i> , 237	IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums <i>(class in PySpin.PySpin)</i> , 256
IEnumerationT_EncoderSelectorEnums <i>(class in PySpin.PySpin)</i> , 238	IEnumerationT_GevGVSPExtendedIDModeEnums <i>(class in PySpin.PySpin)</i> , 257
IEnumerationT_EncoderSourceAEnums <i>(class in PySpin.PySpin)</i> , 239	IEnumerationT_GevIEEE1588ClockAccuracyEnums <i>(class in PySpin.PySpin)</i> , 257
IEnumerationT_EncoderSourceBEnums <i>(class in PySpin.PySpin)</i> , 239	IEnumerationT_GevIEEE1588ModeEnums <i>(class in PySpin.PySpin)</i> , 258
IEnumerationT_EncoderStatusEnums <i>(class in PySpin.PySpin)</i> , 240	IEnumerationT_GevIEEE1588StatusEnums <i>(class in PySpin.PySpin)</i> , 259
IEnumerationT_EventNotificationEnums <i>(class in PySpin.PySpin)</i> , 241	IEnumerationT_GevIEEE1588StatusLatchedEnums <i>(class in PySpin.PySpin)</i> , 259
IEnumerationT_EventSelectorEnums <i>(class in PySpin.PySpin)</i> , 241	IEnumerationT_GevIPConfigurationStatusEnums <i>(class in PySpin.PySpin)</i> , 260
IEnumerationT_ExposureActiveModeEnums <i>(class in PySpin.PySpin)</i> , 242	IEnumerationT_GevPhysicalLinkConfigurationEnums <i>(class in PySpin.PySpin)</i> , 261
IEnumerationT_ExposureAutoEnums <i>(class in PySpin.PySpin)</i> , 243	IEnumerationT_GevSCPDDirectionEnums <i>(class in PySpin.PySpin)</i> , 261
IEnumerationT_ExposureModeEnums <i>(class in PySpin.PySpin)</i> , 243	IEnumerationT_GevSupportedOptionSelectorEnums <i>(class in PySpin.PySpin)</i> , 262
IEnumerationT_ExposureTimeModeEnums <i>(class in PySpin.PySpin)</i> , 244	IEnumerationT_GUIXMLLocationEnum <i>(class in PySpin.PySpin)</i> , 250
IEnumerationT_ExposureTimeSelectorEnums <i>(class in PySpin.PySpin)</i> , 245	IEnumerationT_ImageComponentSelectorEnums <i>(class in PySpin.PySpin)</i> , 263
IEnumerationT_ExternalVoltageSelectorEnums	IEnumerationT_ImageCompressionJPEGFormatOptionEnums

<i>(class in PySpin.PySpin)</i> , 263	<i>(class in PySpin.PySpin)</i> , 281
IEnumerationT_ImageCompressionModeEnums <i>(class in PySpin.PySpin)</i> , 264	IEnumerationT_Scan3dCoordinateReferenceSelectorEnums <i>(class in PySpin.PySpin)</i> , 282
IEnumerationT_ImageCompressionRateOptionEnums <i>(class in PySpin.PySpin)</i> , 265	IEnumerationT_Scan3dCoordinateSelectorEnums <i>(class in PySpin.PySpin)</i> , 283
IEnumerationT_InterfaceTypeEnum <i>(class in PySpin.PySpin)</i> , 265	IEnumerationT_Scan3dCoordinateSystemEnums <i>(class in PySpin.PySpin)</i> , 283
IEnumerationT_LensShadingCoefficientActiveSetEnums <i>(class in PySpin.PySpin)</i> , 267	IEnumerationT_Scan3dCoordinateSystemReferenceEnums <i>(class in PySpin.PySpin)</i> , 284
IEnumerationT_LensShadingCorrectionModeEnums <i>(class in PySpin.PySpin)</i> , 267	IEnumerationT_Scan3dCoordinateTransformSelectorEnums <i>(class in PySpin.PySpin)</i> , 285
IEnumerationT_LineFormatEnums <i>(class in PySpin.PySpin)</i> , 268	IEnumerationT_Scan3dDistanceUnitEnums <i>(class in PySpin.PySpin)</i> , 285
IEnumerationT_LineInputFilterSelectorEnums <i>(class in PySpin.PySpin)</i> , 269	IEnumerationT_Scan3dOutputModeEnums <i>(class in PySpin.PySpin)</i> , 286
IEnumerationT_LineModeEnums <i>(class in PySpin.PySpin)</i> , 269	IEnumerationT_SensorDigitizationTapsEnums <i>(class in PySpin.PySpin)</i> , 287
IEnumerationT_LineSelectorEnums <i>(class in PySpin.PySpin)</i> , 270	IEnumerationT_SensorShutterModeEnums <i>(class in PySpin.PySpin)</i> , 287
IEnumerationT_LineSourceEnums <i>(class in PySpin.PySpin)</i> , 271	IEnumerationT_SensorTapsEnums <i>(class in PySpin.PySpin)</i> , 288
IEnumerationT_LogicBlockLUTInputActivationEnums <i>(class in PySpin.PySpin)</i> , 271	IEnumerationT_SequencerConfigurationModeEnums <i>(class in PySpin.PySpin)</i> , 289
IEnumerationT_LogicBlockLUTInputSelectorEnums <i>(class in PySpin.PySpin)</i> , 272	IEnumerationT_SequencerConfigurationValidEnums <i>(class in PySpin.PySpin)</i> , 289
IEnumerationT_LogicBlockLUTInputSourceEnums <i>(class in PySpin.PySpin)</i> , 273	IEnumerationT_SequencerModeEnums <i>(class in PySpin.PySpin)</i> , 290
IEnumerationT_LogicBlockLUTSelectorEnums <i>(class in PySpin.PySpin)</i> , 273	IEnumerationT_SequencerSetValidEnums <i>(class in PySpin.PySpin)</i> , 291
IEnumerationT_LogicBlockSelectorEnums <i>(class in PySpin.PySpin)</i> , 274	IEnumerationT_SequencerTriggerActivationEnums <i>(class in PySpin.PySpin)</i> , 291
IEnumerationT_LUTSelectorEnums <i>(class in PySpin.PySpin)</i> , 266	IEnumerationT_SequencerTriggerSourceEnums <i>(class in PySpin.PySpin)</i> , 292
IEnumerationT_MultiRoiConfigurationInvalidReasonEnums <i>(class in PySpin.PySpin)</i> , 275	IEnumerationT_SerialPortBaudRateEnums <i>(class in PySpin.PySpin)</i> , 293
IEnumerationT_MultiRoiSelectorEnums <i>(class in PySpin.PySpin)</i> , 275	IEnumerationT_SerialPortParityEnums <i>(class in PySpin.PySpin)</i> , 293
IEnumerationT_PixelColorFilterEnums <i>(class in PySpin.PySpin)</i> , 277	IEnumerationT_SerialPortSelectorEnums <i>(class in PySpin.PySpin)</i> , 294
IEnumerationT_PixelFormatEnums <i>(class in PySpin.PySpin)</i> , 277	IEnumerationT_SerialPortSourceEnums <i>(class in PySpin.PySpin)</i> , 295
IEnumerationT_PixelFormatInfoSelectorEnums <i>(class in PySpin.PySpin)</i> , 278	IEnumerationT_SerialPortStopBitsEnums <i>(class in PySpin.PySpin)</i> , 295
IEnumerationT_PixelSizeEnums <i>(class in PySpin.PySpin)</i> , 279	IEnumerationT_SoftwareSignalSelectorEnums <i>(class in PySpin.PySpin)</i> , 296
IEnumerationT_POEStatusEnum <i>(class in PySpin.PySpin)</i> , 276	IEnumerationT_SourceSelectorEnums <i>(class in PySpin.PySpin)</i> , 297
IEnumerationT_RegionDestinationEnums <i>(class in PySpin.PySpin)</i> , 279	IEnumerationT_StereoResolutionEnums <i>(class in PySpin.PySpin)</i> , 297
IEnumerationT_RegionModeEnums <i>(class in PySpin.PySpin)</i> , 280	IEnumerationT_StreamBufferCountModeEnum <i>(class in PySpin.PySpin)</i> , 298
IEnumerationT_RegionSelectorEnums <i>(class in PySpin.PySpin)</i> , 281	IEnumerationT_StreamBufferHandlingModeEnum <i>(class in PySpin.PySpin)</i> , 299
IEnumerationT_RgbTransformLightSourceEnums	IEnumerationT_StreamModeEnum <i>(class in</i>

<i>PySpin.PySpin</i>), 299	<i>PySpin.PySpin</i>), 317
IEnumerationT_StreamTypeEnum (class in <i>PySpin.PySpin</i>), 300	IEnumerationT_UserSetSelectorEnums (class in <i>PySpin.PySpin</i>), 318
IEnumerationT_TeledyneGigeVisionFilterDriverStatusEnums (class in <i>PySpin.PySpin</i>), 301	IEnumerationT_WhiteClipSelectorEnums (class in <i>PySpin.PySpin</i>), 319
IEnumerationT_TestPatternEnums (class in <i>PySpin.PySpin</i>), 302	IEnumReference (class in <i>PySpin.PySpin</i>), 175
IEnumerationT_TestPatternGeneratorSelectorEnums (class in <i>PySpin.PySpin</i>), 303	IFloat (class in <i>PySpin.PySpin</i>), 319
IEnumerationT_TimerSelectorEnums (class in <i>PySpin.PySpin</i>), 303	IImage (class in <i>PySpin.PySpin</i>), 320
IEnumerationT_TimerStatusEnums (class in <i>PySpin.PySpin</i>), 304	IImageEventHandler (class in <i>PySpin.PySpin</i>), 324
IEnumerationT_TimerTriggerActivationEnums (class in <i>PySpin.PySpin</i>), 305	IImageList (class in <i>PySpin.PySpin</i>), 324
IEnumerationT_TimerTriggerSourceEnums (class in <i>PySpin.PySpin</i>), 305	IImageListEventHandler (class in <i>PySpin.PySpin</i>), 325
IEnumerationT_TLTypeEnum (class in <i>PySpin.PySpin</i>), 301	IImageProcessor (class in <i>PySpin.PySpin</i>), 326
IEnumerationT_TransferComponentSelectorEnums (class in <i>PySpin.PySpin</i>), 306	IInteger (class in <i>PySpin.PySpin</i>), 327
IEnumerationT_TransferControlModeEnums (class in <i>PySpin.PySpin</i>), 307	IInterface (class in <i>PySpin</i>), 66
IEnumerationT_TransferOperationModeEnums (class in <i>PySpin.PySpin</i>), 307	IInterface (class in <i>PySpin.PySpin</i>), 328
IEnumerationT_TransferQueueModeEnums (class in <i>PySpin.PySpin</i>), 308	IInterfaceArrivalEventHandler (class in <i>PySpin.PySpin</i>), 329
IEnumerationT_TransferSelectorEnums (class in <i>PySpin.PySpin</i>), 309	IInterfaceEventHandler (class in <i>PySpin.PySpin</i>), 329
IEnumerationT_TransferStatusSelectorEnums (class in <i>PySpin.PySpin</i>), 309	IInterfaceList (class in <i>PySpin.PySpin</i>), 329
IEnumerationT_TransferTriggerActivationEnums (class in <i>PySpin.PySpin</i>), 310	IInterfaceRemovalEventHandler (class in <i>PySpin.PySpin</i>), 330
IEnumerationT_TransferTriggerModeEnums (class in <i>PySpin.PySpin</i>), 311	ILoggingEventHandler (class in <i>PySpin.PySpin</i>), 330
IEnumerationT_TransferTriggerSelectorEnums (class in <i>PySpin.PySpin</i>), 311	Image (class in <i>PySpin</i>), 46
IEnumerationT_TransferTriggerSourceEnums (class in <i>PySpin.PySpin</i>), 312	Image (class in <i>PySpin.PySpin</i>), 339
IEnumerationT_TriggerActivationEnums (class in <i>PySpin.PySpin</i>), 313	ImageComponentEnable (<i>PySpin.Camera</i> property), 27
IEnumerationT_TriggerModeEnums (class in <i>PySpin.PySpin</i>), 313	ImageComponentEnable (<i>PySpin.PySpin.Camera</i> property), 133
IEnumerationT_TriggerOverlapEnums (class in <i>PySpin.PySpin</i>), 314	ImageComponentSelector (<i>PySpin.Camera</i> property), 27
IEnumerationT_TriggerSelectorEnums (class in <i>PySpin.PySpin</i>), 315	ImageComponentSelector (<i>PySpin.PySpin.Camera</i> property), 133
IEnumerationT_TriggerSourceEnums (class in <i>PySpin.PySpin</i>), 315	ImageCompressionBitrate (<i>PySpin.Camera</i> property), 27
IEnumerationT_U3VCurrentSpeedEnums (class in <i>PySpin.PySpin</i>), 316	ImageCompressionBitrate (<i>PySpin.PySpin.Camera</i> property), 133
IEnumerationT_UserOutputSelectorEnums (class in <i>PySpin.PySpin</i>), 317	ImageCompressionJPGFormatOption (<i>PySpin.Camera</i> property), 27
IEnumerationT_UserSetDefaultEnums (class in <i>PySpin.PySpin</i>), 317	ImageCompressionJPGFormatOption (<i>PySpin.PySpin.Camera</i> property), 133
	ImageCompressionMode (<i>PySpin.Camera</i> property), 27
	ImageCompressionMode (<i>PySpin.PySpin.Camera</i> property), 133
	ImageCompressionQuality (<i>PySpin.Camera</i> property), 27
	ImageCompressionQuality (<i>PySpin.PySpin.Camera</i> property), 133
	ImageCompressionRateOption (<i>PySpin.Camera</i> property), 27
	ImageCompressionRateOption (<i>PySpin.PySpin.Camera</i> property), 133
	ImageEventHandler (class in <i>PySpin</i>), 6
	ImageEventHandler (class in <i>PySpin.PySpin</i>), 348

`ImageList` (class in `PySpin`), 55
`ImageList` (class in `PySpin.PySpin`), 348
`ImageListEventHandler` (class in `PySpin`), 7
`ImageListEventHandler` (class in `PySpin.PySpin`), 350
`ImagePixel` (class in `PySpin.PySpin`), 350
`ImageProcessor` (class in `PySpin`), 56
`ImageProcessor` (class in `PySpin.PySpin`), 350
`ImagePtr` (class in `PySpin`), 58
`ImagePtr` (class in `PySpin.PySpin`), 351
`ImageUtility` (class in `PySpin`), 58
`ImageUtility` (class in `PySpin.PySpin`), 352
`ImageUtilityCCM` (class in `PySpin`), 59
`ImageUtilityCCM` (class in `PySpin.PySpin`), 353
`ImageUtilityHeatmap` (class in `PySpin`), 60
`ImageUtilityHeatmap` (class in `PySpin.PySpin`), 354
`ImageUtilityPolarization` (class in `PySpin`), 62
`ImageUtilityPolarization` (class in `PySpin.PySpin`), 355
`ImageUtilityStereo` (class in `PySpin`), 64
`ImageUtilityStereo` (class in `PySpin.PySpin`), 358
`ImposeAccessMode()` (`PySpin.PySpin.CBooleanPtr` method), 84
`ImposeAccessMode()` (`PySpin.PySpin.CCategoryPtr` method), 87
`ImposeAccessMode()` (`PySpin.PySpin.CCommandPtr` method), 90
`ImposeAccessMode()` (`PySpin.PySpin.CEnumEntryPtr` method), 93
`ImposeAccessMode()` (`PySpin.PySpin.CEnumerationPtr` method), 96
`ImposeAccessMode()` (`PySpin.PySpin.CIntegerPtr` method), 100
`ImposeAccessMode()` (`PySpin.PySpin.CNodePtr` method), 105
`ImposeAccessMode()` (`PySpin.PySpin.CRegisterPtr` method), 108
`ImposeAccessMode()` (`PySpin.PySpin.CStringPtr` method), 112
`ImposeAccessMode()` (`PySpin.PySpin.CValuePtr` method), 114
`ImposeAccessMode()` (`PySpin.PySpin.INode` method), 331
`ImposeAccessMode()` (`PySpin.PySpin.Node` method), 376
`ImposeMax()` (`PySpin.PySpin.CIntegerPtr` method), 100
`ImposeMax()` (`PySpin.PySpin.FloatNode` method), 165
`ImposeMax()` (`PySpin.PySpin.IFloat` method), 320
`ImposeMax()` (`PySpin.PySpin.IInteger` method), 327
`ImposeMax()` (`PySpin.PySpin.IntegerNode` method), 363
`ImposeMin()` (`PySpin.PySpin.CIntegerPtr` method), 100
`ImposeMin()` (`PySpin.PySpin.FloatNode` method), 165
`ImposeMin()` (`PySpin.PySpin.IFloat` method), 320
`ImposeMin()` (`PySpin.PySpin.IInteger` method), 327
`ImposeMin()` (`PySpin.PySpin.IntegerNode` method), 363
`ImposeVisibility()` (`PySpin.PySpin.CBooleanPtr` method), 85
`ImposeVisibility()` (`PySpin.PySpin.CCategoryPtr` method), 87
`ImposeVisibility()` (`PySpin.PySpin.CCommandPtr` method), 90
`ImposeVisibility()` (`PySpin.PySpin.CEnumEntryPtr` method), 93
`ImposeVisibility()` (`PySpin.PySpin.CEnumerationPtr` method), 96
`ImposeVisibility()` (`PySpin.PySpin.CIntegerPtr` method), 100
`ImposeVisibility()` (`PySpin.PySpin.CNodePtr` method), 105
`ImposeVisibility()` (`PySpin.PySpin.CRegisterPtr` method), 108
`ImposeVisibility()` (`PySpin.PySpin.CStringPtr` method), 112
`ImposeVisibility()` (`PySpin.PySpin.CValuePtr` method), 114
`ImposeVisibility()` (`PySpin.PySpin.INode` method), 331
`ImposeVisibility()` (`PySpin.PySpin.Node` method), 376
`IncompatibleDeviceCount` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleDeviceCount` (`PySpin.TransportLayerInterface` property), 78
`IncompatibleDeviceID` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleDeviceID` (`PySpin.TransportLayerInterface` property), 78
`IncompatibleDeviceModelName` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleDeviceModelName` (`PySpin.TransportLayerInterface` property), 78
`IncompatibleDeviceSelector` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleDeviceSelector` (`PySpin.TransportLayerInterface` property), 78
`IncompatibleDeviceVendorName` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleDeviceVendorName` (`PySpin.TransportLayerInterface` property), 78
`IncompatibleGevDeviceIPAddress` (`PySpin.PySpin.TransportLayerInterface` property), 399
`IncompatibleGevDeviceIPAddress` (`PySpin.TransportLayerInterface` property), 78

- IncompatibleGevDeviceMACAddress
(*PySpin.PySpin.TransportLayerInterface* property), 399
- IncompatibleGevDeviceMACAddress
(*PySpin.TransportLayerInterface* property), 78
- IncompatibleGevDeviceSubnetMask
(*PySpin.PySpin.TransportLayerInterface* property), 399
- IncompatibleGevDeviceSubnetMask
(*PySpin.TransportLayerInterface* property), 78
- indexedColor_8bit (*PySpin.PySpin.BMPOption* property), 81
- InferenceBoundingBox (class in *PySpin.PySpin*), 360
- InferenceBoundingBoxResult (class in *PySpin.PySpin*), 360
- InferenceBoxCircle (class in *PySpin.PySpin*), 360
- InferenceBoxRect (class in *PySpin.PySpin*), 361
- InferenceBoxRotatedRect (class in *PySpin.PySpin*), 361
- Init() (*PySpin.Camera* method), 27
- Init() (*PySpin.CameraBase* method), 38
- Init() (*PySpin.PySpin.Camera* method), 133
- Init() (*PySpin.PySpin.CameraBase* method), 143
- Init() (*PySpin.PySpin.ICameraBase* method), 169
- INode (class in *PySpin.PySpin*), 330
- INodeMap (class in *PySpin.PySpin*), 332
- INodeMapDyn (class in *PySpin.PySpin*), 332
- insert() (*PySpin.PySpin.node_vector* method), 409
- insert() (*PySpin.PySpin.value_vector* method), 410
- int64_autovector_t (class in *PySpin.PySpin*), 408
- IntegerNode (class in *PySpin.PySpin*), 362
- InterfaceArrivalEventHandler (class in *PySpin*), 7
- InterfaceArrivalEventHandler (class in *PySpin.PySpin*), 364
- InterfaceDisplayName
(*PySpin.PySpin.TransportLayerInterface* property), 399
- InterfaceDisplayName
(*PySpin.PySpin.TransportLayerSystem* property), 402
- InterfaceDisplayName
(*PySpin.TransportLayerInterface* property), 78
- InterfaceEventHandler (class in *PySpin*), 7
- InterfaceEventHandler (class in *PySpin.PySpin*), 364
- InterfaceID (*PySpin.PySpin.TransportLayerInterface* property), 399
- InterfaceID (*PySpin.PySpin.TransportLayerSystem* property), 402
- InterfaceID (*PySpin.TransportLayerInterface* property), 78
- InterfaceList (class in *PySpin*), 67
- InterfaceList (class in *PySpin.PySpin*), 364
- InterfacePtr (class in *PySpin*), 68
- InterfacePtr (class in *PySpin.PySpin*), 365
- InterfaceRemovalEventHandler (class in *PySpin*), 7
- InterfaceRemovalEventHandler (class in *PySpin.PySpin*), 365
- InterfaceSelector (*PySpin.PySpin.TransportLayerSystem* property), 402
- InterfaceType (*PySpin.PySpin.TransportLayerInterface* property), 399
- InterfaceType (*PySpin.TransportLayerInterface* property), 78
- InterfaceUpdateList
(*PySpin.PySpin.TransportLayerSystem* property), 402
- interlaced (*PySpin.PySpin.PNGOption* property), 383
- IntRegNode (class in *PySpin.PySpin*), 361
- InvalidateNode() (*PySpin.PySpin.CBooleanPtr* method), 85
- InvalidateNode() (*PySpin.PySpin.CCategoryPtr* method), 87
- InvalidateNode() (*PySpin.PySpin.CCommandPtr* method), 90
- InvalidateNode() (*PySpin.PySpin.CEnumEntryPtr* method), 93
- InvalidateNode() (*PySpin.PySpin.CEnumerationPtr* method), 96
- InvalidateNode() (*PySpin.PySpin.CIntegerPtr* method), 100
- InvalidateNode() (*PySpin.PySpin.CNodePtr* method), 105
- InvalidateNode() (*PySpin.PySpin.CRegisterPtr* method), 108
- InvalidateNode() (*PySpin.PySpin.CStringPtr* method), 112
- InvalidateNode() (*PySpin.PySpin.CValuePtr* method), 115
- InvalidateNode() (*PySpin.PySpin.INode* method), 331
- InvalidateNode() (*PySpin.PySpin.Node* method), 376
- InvalidateNodes() (*PySpin.PySpin.CNodeMapDynPtr* method), 102
- InvalidateNodes() (*PySpin.PySpin.CNodeMapPtr* method), 104
- InvalidateNodes() (*PySpin.PySpin.INodeMap* method), 332
- InvalidateNodes() (*PySpin.PySpin.NodeMap* method), 380
- invalidDataFlag (*PySpin.PySpin.StereoCameraParameters* property), 389
- invalidDataValue (*PySpin.PySpin.StereoCameraParameters* property), 389
- IPersistScript (class in *PySpin.PySpin*), 334
- IPointCloud (class in *PySpin.PySpin*), 334
- IReference (class in *PySpin.PySpin*), 335
- IRegister (class in *PySpin.PySpin*), 335
- IsAccessModeCacheable()
(*PySpin.PySpin.CBooleanPtr* method), 85

<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CCategoryPtr</code> method), 87	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CCategoryPtr</code> method), 88
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CCommandPtr</code> method), 90	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CCommandPtr</code> method), 90
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CEnumEntryPtr</code> method), 93	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CEnumEntryPtr</code> method), 93
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CEnumerationPtr</code> method), 96	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CEnumerationPtr</code> method), 96
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CIntegerPtr</code> method), 100	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CIntegerPtr</code> method), 100
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CNodePtr</code> method), 105	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CNodePtr</code> method), 106
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CRegisterPtr</code> method), 108	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CRegisterPtr</code> method), 108
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CStringPtr</code> method), 112	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CStringPtr</code> method), 112
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.CValuePtr</code> method), 115	<code>IsDeprecated()</code> (<code>PySpin.PySpin.CValuePtr</code> method), 115
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.INode</code> method), 331	<code>IsDeprecated()</code> (<code>PySpin.PySpin.INode</code> method), 331
<code>IsAccessModeCacheable()</code> (<code>PySpin.PySpin.Node</code> method), 376	<code>IsDeprecated()</code> (<code>PySpin.PySpin.Node</code> method), 376
<code>IsAvailable()</code> (in module <code>PySpin.PySpin</code>), 365	<code>IsDone()</code> (<code>PySpin.PySpin.CCommandPtr</code> method), 90
<code>IsCachable()</code> (<code>PySpin.PySpin.CBooleanPtr</code> method), 85	<code>IsDone()</code> (<code>PySpin.PySpin.CommandNode</code> method), 152
<code>IsCachable()</code> (<code>PySpin.PySpin.CCategoryPtr</code> method), 88	<code>IsDone()</code> (<code>PySpin.PySpin.ICommand</code> method), 174
<code>IsCachable()</code> (<code>PySpin.PySpin.CCommandPtr</code> method), 90	<code>ISelector</code> (class in <code>PySpin.PySpin</code>), 336
<code>IsCachable()</code> (<code>PySpin.PySpin.CEnumEntryPtr</code> method), 93	<code>ISelectorDigit</code> (class in <code>PySpin.PySpin</code>), 336
<code>IsCachable()</code> (<code>PySpin.PySpin.CEnumerationPtr</code> method), 96	<code>IsEmpty()</code> (<code>PySpin.PySpin.CSelectorSet</code> method), 110
<code>IsCachable()</code> (<code>PySpin.PySpin.CIntegerPtr</code> method), 100	<code>IsFeature()</code> (<code>PySpin.PySpin.CBooleanPtr</code> method), 85
<code>IsCachable()</code> (<code>PySpin.PySpin.CNodePtr</code> method), 105	<code>IsFeature()</code> (<code>PySpin.PySpin.CCategoryPtr</code> method), 88
<code>IsCachable()</code> (<code>PySpin.PySpin.CRegisterPtr</code> method), 108	<code>IsFeature()</code> (<code>PySpin.PySpin.CCommandPtr</code> method), 90
<code>IsCachable()</code> (<code>PySpin.PySpin.CStringPtr</code> method), 112	<code>IsFeature()</code> (<code>PySpin.PySpin.CEnumEntryPtr</code> method), 93
<code>IsCachable()</code> (<code>PySpin.PySpin.CValuePtr</code> method), 115	<code>IsFeature()</code> (<code>PySpin.PySpin.CEnumerationPtr</code> method), 96
<code>IsCachable()</code> (<code>PySpin.PySpin.INode</code> method), 331	<code>IsFeature()</code> (<code>PySpin.PySpin.CIntegerPtr</code> method), 101
<code>IsCachable()</code> (<code>PySpin.PySpin.Node</code> method), 376	<code>IsFeature()</code> (<code>PySpin.PySpin.CNodePtr</code> method), 106
<code>IsCacheable()</code> (in module <code>PySpin.PySpin</code>), 367	<code>IsFeature()</code> (<code>PySpin.PySpin.CRegisterPtr</code> method), 108
<code>IsCameraInUse()</code> (<code>PySpin.IInterface</code> method), 66	<code>IsFeature()</code> (<code>PySpin.PySpin.CStringPtr</code> method), 112
<code>IsCameraInUse()</code> (<code>PySpin.PySpin.IInterface</code> method), 328	<code>IsFeature()</code> (<code>PySpin.PySpin.CValuePtr</code> method), 115
<code>IsCompressed()</code> (<code>PySpin.Image</code> method), 51	<code>IsFeature()</code> (<code>PySpin.PySpin.INode</code> method), 331
<code>IsCompressed()</code> (<code>PySpin.PySpin.IImage</code> method), 322	<code>IsFeature()</code> (<code>PySpin.PySpin.Node</code> method), 376
<code>IsCompressed()</code> (<code>PySpin.PySpin.Image</code> method), 345	<code>IsImplemented()</code> (in module <code>PySpin.PySpin</code>), 367
<code>IsDeprecated()</code> (<code>PySpin.PySpin.CBooleanPtr</code> method), 85	<code>IsIncomplete()</code> (<code>PySpin.Image</code> method), 51
	<code>IsIncomplete()</code> (<code>PySpin.PySpin.IImage</code> method), 322
	<code>IsIncomplete()</code> (<code>PySpin.PySpin.Image</code> method), 345
	<code>IsInitialized()</code> (<code>PySpin.CameraBase</code> method), 38
	<code>IsInitialized()</code> (<code>PySpin.PySpin.CameraBase</code> method), 143
	<code>IsInitialized()</code> (<code>PySpin.PySpin.ICameraBase</code> method), 169
	<code>IsInUse()</code> (<code>PySpin.Image</code> method), 51
	<code>IsInUse()</code> (<code>PySpin.PySpin.IImage</code> method), 322

- [IsInUse\(\) \(PySpin.PySpin.Image method\), 345](#)
[IsInUse\(\) \(PySpin.PySpin.ISystem method\), 338](#)
[IsInUse\(\) \(PySpin.PySpin.System method\), 392](#)
[IsInUse\(\) \(PySpin.System method\), 72](#)
[IsPEnable \(PySpin.Camera property\), 27](#)
[IsPEnable \(PySpin.PySpin.Camera property\), 133](#)
[IsReadable\(\) \(in module PySpin.PySpin\), 368](#)
[IsSelector\(\) \(PySpin.PySpin.CBooleanPtr method\), 85](#)
[IsSelector\(\) \(PySpin.PySpin.CCategoryPtr method\), 88](#)
[IsSelector\(\) \(PySpin.PySpin.CCommandPtr method\), 90](#)
[IsSelector\(\) \(PySpin.PySpin.CEnumEntryPtr method\), 93](#)
[IsSelector\(\) \(PySpin.PySpin.CEnumerationPtr method\), 96](#)
[IsSelector\(\) \(PySpin.PySpin.CIntegerPtr method\), 101](#)
[IsSelector\(\) \(PySpin.PySpin.CNodePtr method\), 106](#)
[IsSelector\(\) \(PySpin.PySpin.CRegisterPtr method\), 109](#)
[IsSelector\(\) \(PySpin.PySpin.CSelectorPtr method\), 110](#)
[IsSelector\(\) \(PySpin.PySpin.CStringPtr method\), 112](#)
[IsSelector\(\) \(PySpin.PySpin.CValuePtr method\), 115](#)
[IsSelector\(\) \(PySpin.PySpin.ISelector method\), 336](#)
[IsSelector\(\) \(PySpin.PySpin.Node method\), 376](#)
[IsSelfClearing\(\) \(PySpin.PySpin.CEnumEntryPtr method\), 93](#)
[IsSelfClearing\(\) \(PySpin.PySpin.EnumEntryNode method\), 160](#)
[IsSelfClearing\(\) \(PySpin.PySpin.IEnumEntry method\), 175](#)
[IsStereoCamera\(\) \(PySpin.ImageUtilityStereo static method\), 66](#)
[IsStereoCamera\(\) \(PySpin.PySpin.ImageUtilityStereo static method\), 360](#)
[IsStreamable\(\) \(PySpin.PySpin.CBooleanPtr method\), 85](#)
[IsStreamable\(\) \(PySpin.PySpin.CCategoryPtr method\), 88](#)
[IsStreamable\(\) \(PySpin.PySpin.CCommandPtr method\), 90](#)
[IsStreamable\(\) \(PySpin.PySpin.CEnumEntryPtr method\), 93](#)
[IsStreamable\(\) \(PySpin.PySpin.CEnumerationPtr method\), 96](#)
[IsStreamable\(\) \(PySpin.PySpin.CIntegerPtr method\), 101](#)
[IsStreamable\(\) \(PySpin.PySpin.CNodePtr method\), 106](#)
[IsStreamable\(\) \(PySpin.PySpin.CRegisterPtr method\), 109](#)
[IsStreamable\(\) \(PySpin.PySpin.CStringPtr method\), 112](#)
[IsStreamable\(\) \(PySpin.PySpin.CValuePtr method\), 115](#)
[IsStreamable\(\) \(PySpin.PySpin.INode method\), 331](#)
[IsStreamable\(\) \(PySpin.PySpin.Node method\), 376](#)
[IsStreaming\(\) \(PySpin.CameraBase method\), 38](#)
[IsStreaming\(\) \(PySpin.PySpin.CameraBase method\), 143](#)
[IsStreaming\(\) \(PySpin.PySpin.ICameraBase method\), 169](#)
[IString \(class in PySpin.PySpin\), 337](#)
[IsValid\(\) \(PySpin.CameraBase method\), 38](#)
[IsValid\(\) \(PySpin.CBasePtr method\), 10](#)
[IsValid\(\) \(PySpin.IInterface method\), 66](#)
[IsValid\(\) \(PySpin.PySpin.CameraBase method\), 143](#)
[IsValid\(\) \(PySpin.PySpin.CBasePtr method\), 83](#)
[IsValid\(\) \(PySpin.PySpin.CBooleanPtr method\), 85](#)
[IsValid\(\) \(PySpin.PySpin.CCategoryPtr method\), 88](#)
[IsValid\(\) \(PySpin.PySpin.CCommandPtr method\), 90](#)
[IsValid\(\) \(PySpin.PySpin.CDeviceInfoPtr method\), 91](#)
[IsValid\(\) \(PySpin.PySpin.CEnumEntryPtr method\), 93](#)
[IsValid\(\) \(PySpin.PySpin.CEnumerationPtr method\), 96](#)
[IsValid\(\) \(PySpin.PySpin.CIntegerPtr method\), 101](#)
[IsValid\(\) \(PySpin.PySpin.CNodeMapDynPtr method\), 102](#)
[IsValid\(\) \(PySpin.PySpin.CNodeMapPtr method\), 104](#)
[IsValid\(\) \(PySpin.PySpin.CNodePtr method\), 106](#)
[IsValid\(\) \(PySpin.PySpin.CRegisterPtr method\), 109](#)
[IsValid\(\) \(PySpin.PySpin.CSelectorPtr method\), 110](#)
[IsValid\(\) \(PySpin.PySpin.CStringPtr method\), 112](#)
[IsValid\(\) \(PySpin.PySpin.CValuePtr method\), 115](#)
[IsValid\(\) \(PySpin.PySpin.ICameraBase method\), 169](#)
[IsValid\(\) \(PySpin.PySpin.IInterface method\), 328](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CBooleanPtr method\), 85](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CCategoryPtr method\), 88](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CCommandPtr method\), 90](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CEnumEntryPtr method\), 93](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CEnumerationPtr method\), 96](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CIntegerPtr method\), 101](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CRegisterPtr method\), 109](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CStringPtr method\), 112](#)
[IsValidCacheValid\(\) \(PySpin.PySpin.CValuePtr method\), 115](#)

IsValidCacheValid() (*PySpin.PySpin.IValue method*), 339

IsValidCacheValid() (*PySpin.PySpin.ValueNode method*), 403

IsVisible() (*in module PySpin.PySpin*), 369

IsWritable() (*in module PySpin.PySpin*), 369

ISystem (*class in PySpin.PySpin*), 337

ISystemEventHandler (*class in PySpin.PySpin*), 339

IValue (*class in PySpin.PySpin*), 339

J

JPEGOption (*class in PySpin.PySpin*), 370

JPG2Option (*class in PySpin.PySpin*), 370

L

LargePenalty (*PySpin.Camera property*), 28

LargePenalty (*PySpin.PySpin.Camera property*), 133

length() (*PySpin.PySpin.gcstring method*), 407

LensShadingCoefficientActiveSet

(*PySpin.Camera property*), 28

LensShadingCoefficientActiveSet

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionCalibration

(*PySpin.Camera property*), 28

LensShadingCorrectionCalibration

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionCalibrationGainLimit

(*PySpin.Camera property*), 28

LensShadingCorrectionCalibrationGainLimit

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionCalibrationSetup

(*PySpin.Camera property*), 28

LensShadingCorrectionCalibrationSetup

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionCalibrationStatus

(*PySpin.Camera property*), 28

LensShadingCorrectionCalibrationStatus

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionMode (*PySpin.Camera property*), 28

LensShadingCorrectionMode

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionStepSize (*PySpin.Camera property*), 28

LensShadingCorrectionStepSize

(*PySpin.PySpin.Camera property*), 133

LensShadingCorrectionVersion (*PySpin.Camera property*), 28

LensShadingCorrectionVersion

(*PySpin.PySpin.Camera property*), 133

LibraryVersion (*class in PySpin.PySpin*), 371

LineFilterWidth (*PySpin.Camera property*), 28

LineFilterWidth (*PySpin.PySpin.Camera property*), 133

LineFormat (*PySpin.Camera property*), 28

LineFormat (*PySpin.PySpin.Camera property*), 133

LineInputFilterSelector (*PySpin.Camera property*), 28

LineInputFilterSelector (*PySpin.PySpin.Camera property*), 133

LineInverter (*PySpin.Camera property*), 28

LineInverter (*PySpin.PySpin.Camera property*), 133

LineMode (*PySpin.Camera property*), 28

LineMode (*PySpin.PySpin.Camera property*), 133

LinePitch (*PySpin.Camera property*), 28

LinePitch (*PySpin.PySpin.Camera property*), 133

LineSelector (*PySpin.Camera property*), 28

LineSelector (*PySpin.PySpin.Camera property*), 134

LineSource (*PySpin.Camera property*), 28

LineSource (*PySpin.PySpin.Camera property*), 134

LineStatus (*PySpin.Camera property*), 28

LineStatus (*PySpin.PySpin.Camera property*), 134

LineStatusAll (*PySpin.Camera property*), 28

LineStatusAll (*PySpin.PySpin.Camera property*), 134

LinkErrorCount (*PySpin.Camera property*), 28

LinkErrorCount (*PySpin.PySpin.Camera property*), 134

LinkRecoveryCount (*PySpin.Camera property*), 28

LinkRecoveryCount (*PySpin.PySpin.Camera property*), 134

LinkUptime (*PySpin.Camera property*), 28

LinkUptime (*PySpin.PySpin.Camera property*), 134

Load() (*PySpin.Image static method*), 52

Load() (*PySpin.ImageList static method*), 55

Load() (*PySpin.PySpin.Image static method*), 345

Load() (*PySpin.PySpin.ImageList static method*), 349

LoadFromBag() (*PySpin.PySpin.CFeatureBag method*), 97

LoadPointCloudFromPly() (*PySpin.PointCloud method*), 68

LoadPointCloudFromPly() (*PySpin.PySpin.IPointCloud method*), 334

LoadPointCloudFromPly() (*PySpin.PySpin.PointCloud method*), 383

LoadXMLFromFile() (*PySpin.PySpin.CNodeMapDynPtr method*), 102

LoadXMLFromFile() (*PySpin.PySpin.INodeMapDyn method*), 333

LoadXMLFromFile() (*PySpin.PySpin.NodeMap method*), 381

LoadXMLFromFileInject() (*PySpin.PySpin.CNodeMapDynPtr method*), 102

LoadXMLFromFileInject() (*PySpin.PySpin.INodeMapDyn method*), 333

LoadXMLFromFileInject() (*PySpin.PySpin.NodeMap method*), 381

- LoadXMLFromString()
(*PySpin.PySpin.CNodeMapDynPtr* method), 102
- LoadXMLFromString() (*PySpin.PySpin.INodeMapDyn* method), 333
- LoadXMLFromString() (*PySpin.PySpin.NodeMap* method), 381
- LoadXMLFromStringInject()
(*PySpin.PySpin.CNodeMapDynPtr* method), 102
- LoadXMLFromStringInject()
(*PySpin.PySpin.INodeMapDyn* method), 333
- LoadXMLFromStringInject()
(*PySpin.PySpin.NodeMap* method), 381
- LoadXMLFromZIPData()
(*PySpin.PySpin.CNodeMapDynPtr* method), 102
- LoadXMLFromZIPData()
(*PySpin.PySpin.INodeMapDyn* method), 333
- LoadXMLFromZIPData() (*PySpin.PySpin.NodeMap* method), 382
- LoadXMLFromZIPFile()
(*PySpin.PySpin.CNodeMapDynPtr* method), 102
- LoadXMLFromZIPFile()
(*PySpin.PySpin.INodeMapDyn* method), 333
- LoadXMLFromZIPFile() (*PySpin.PySpin.NodeMap* method), 382
- LoggingEventData (class in *PySpin.PySpin*), 371
- LoggingEventDataPtr (class in *PySpin*), 8
- LoggingEventDataPtr (class in *PySpin.PySpin*), 372
- LoggingEventHandler (class in *PySpin*), 8
- LoggingEventHandler (class in *PySpin.PySpin*), 372
- LogicBlockLUTInputActivation (*PySpin.Camera* property), 28
- LogicBlockLUTInputActivation
(*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTInputSelector (*PySpin.Camera* property), 28
- LogicBlockLUTInputSelector
(*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTInputSource (*PySpin.Camera* property), 28
- LogicBlockLUTInputSource (*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTOutputValue (*PySpin.Camera* property), 28
- LogicBlockLUTOutputValue (*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTOutputValueAll (*PySpin.Camera* property), 28
- LogicBlockLUTOutputValueAll
(*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTRowIndex (*PySpin.Camera* property), 28
- LogicBlockLUTRowIndex (*PySpin.PySpin.Camera* property), 134
- LogicBlockLUTSelector (*PySpin.Camera* property), 29
- LogicBlockLUTSelector (*PySpin.PySpin.Camera* property), 134
- LogicBlockSelector (*PySpin.Camera* property), 29
- LogicBlockSelector (*PySpin.PySpin.Camera* property), 134
- LUTEnable (*PySpin.Camera* property), 27
- LUTEnable (*PySpin.PySpin.Camera* property), 133
- LUTIndex (*PySpin.Camera* property), 27
- LUTIndex (*PySpin.PySpin.Camera* property), 133
- LUTSelector (*PySpin.Camera* property), 28
- LUTSelector (*PySpin.PySpin.Camera* property), 133
- LUTValue (*PySpin.Camera* property), 28
- LUTValue (*PySpin.PySpin.Camera* property), 133
- LUTValueAll (*PySpin.Camera* property), 28
- LUTValueAll (*PySpin.PySpin.Camera* property), 133
- ## M
- major (*PySpin.PySpin.LibraryVersion* property), 371
- Major (*PySpin.PySpin.Version_t* property), 404
- max_size() (*PySpin.PySpin.gcstring* method), 407
- max_size() (*PySpin.PySpin.node_vector* method), 409
- max_size() (*PySpin.PySpin.value_vector* method), 411
- MaxDatarateThreshold (*PySpin.Camera* property), 29
- MaxDatarateThreshold (*PySpin.PySpin.Camera* property), 134
- maxDepthThresholdInMeter
(*PySpin.ImageUtilityStereo* property), 66
- maxDepthThresholdInMeter
(*PySpin.PySpin.ImageUtilityStereo* property), 360
- maxDepthThresholdInMm (*PySpin.ImageUtilityStereo* property), 66
- maxDepthThresholdInMm
(*PySpin.PySpin.ImageUtilityStereo* property), 360
- MaxDeviceResetTime (*PySpin.Camera* property), 29
- MaxDeviceResetTime (*PySpin.PySpin.Camera* property), 134
- MergeXMLFiles() (*PySpin.PySpin.CNodeMapDynPtr* method), 103
- MergeXMLFiles() (*PySpin.PySpin.INodeMapDyn* method), 333
- message (*PySpin.SpinnakerException* attribute), 69
- minor (*PySpin.PySpin.LibraryVersion* property), 371
- Minor (*PySpin.PySpin.Version_t* property), 404
- MJPEGOption (class in *PySpin.PySpin*), 372

module

PySpin.PySpin, 81

MultiRoiConfigurationInvalidReason
(PySpin.Camera property), 29MultiRoiConfigurationInvalidReason
(PySpin.PySpin.Camera property), 134MultiRoiConfigurationInvalidReasonAll
(PySpin.Camera property), 29MultiRoiConfigurationInvalidReasonAll
(PySpin.PySpin.Camera property), 134

MultiRoiEnable (PySpin.Camera property), 29

MultiRoiEnable (PySpin.PySpin.Camera property),
134MultiRoiFeatureEnable (PySpin.Camera property),
29MultiRoiFeatureEnable (PySpin.PySpin.Camera
property), 134

MultiRoiHeight (PySpin.Camera property), 29

MultiRoiHeight (PySpin.PySpin.Camera property),
134

MultiRoiOffsetX (PySpin.Camera property), 29

MultiRoiOffsetX (PySpin.PySpin.Camera property),
134

MultiRoiOffsetY (PySpin.Camera property), 29

MultiRoiOffsetY (PySpin.PySpin.Camera property),
134

MultiRoiSelector (PySpin.Camera property), 29

MultiRoiSelector (PySpin.PySpin.Camera property),
134

MultiRoiWidth (PySpin.Camera property), 29

MultiRoiWidth (PySpin.PySpin.Camera property), 134

MultiRoiWindows (PySpin.Camera property), 29

MultiRoiWindows (PySpin.PySpin.Camera property),
134

N

Node (class in PySpin.PySpin), 373

node_vector (class in PySpin.PySpin), 408

NodeCallback (class in PySpin.PySpin), 377

NodeMap (class in PySpin.PySpin), 378

npos (PySpin.PySpin.gcstring attribute), 407

num_pixel_values (PySpin.ChannelStatistics prop-
erty), 42num_pixel_values (PySpin.PySpin.ChannelStatistics
property), 148

NumDirections (PySpin.Camera property), 29

NumDirections (PySpin.PySpin.Camera property), 134

O

OffsetX (PySpin.Camera property), 29

OffsetX (PySpin.PySpin.Camera property), 134

OffsetY (PySpin.Camera property), 29

OffsetY (PySpin.PySpin.Camera property), 134

OnDeviceArrival() (PySpin.DeviceArrivalEventHandler
method), 5OnDeviceArrival() (PySpin.InterfaceEventHandler
method), 7OnDeviceArrival() (PySpin.PySpin.DeviceArrivalEventHandler
method), 153OnDeviceArrival() (PySpin.PySpin.IDeviceArrivalEventHandler
method), 174OnDeviceArrival() (PySpin.PySpin.IInterfaceEventHandler
method), 329OnDeviceArrival() (PySpin.PySpin.InterfaceEventHandler
method), 364OnDeviceEvent() (PySpin.DeviceEventHandler
method), 6OnDeviceEvent() (PySpin.PySpin.DeviceEventHandler
method), 153OnDeviceEvent() (PySpin.PySpin.IDeviceEventHandler
method), 174OnDeviceRemoval() (PySpin.DeviceRemovalEventHandler
method), 6OnDeviceRemoval() (PySpin.InterfaceEventHandler
method), 7OnDeviceRemoval() (PySpin.PySpin.DeviceRemovalEventHandler
method), 154OnDeviceRemoval() (PySpin.PySpin.IDeviceRemovalEventHandler
method), 175OnDeviceRemoval() (PySpin.PySpin.IInterfaceEventHandler
method), 329OnDeviceRemoval() (PySpin.PySpin.InterfaceEventHandler
method), 364OnImageEvent() (PySpin.ImageEventHandler method),
6OnImageEvent() (PySpin.PySpin.ImageEventHandler
method), 348OnImageListEvent() (PySpin.ImageListEventHandler
method), 7OnImageListEvent() (PySpin.PySpin.ImageListEventHandler
method), 350OnInterfaceArrival()
(PySpin.InterfaceArrivalEventHandler
method), 7OnInterfaceArrival()
(PySpin.PySpin.IInterfaceArrivalEventHandler
method), 329OnInterfaceArrival()
(PySpin.PySpin.InterfaceArrivalEventHandler
method), 364OnInterfaceArrival()
(PySpin.PySpin.ISystemEventHandler method),
339OnInterfaceArrival()
(PySpin.PySpin.SystemEventHandler method),
395

OnInterfaceArrival() (PySpin.SystemEventHandler

- method), 8
- OnInterfaceRemoval() (PySpin.InterfaceRemovalEventHandler method), 7
- OnInterfaceRemoval() (PySpin.PySpin.IInterfaceRemovalEventHandler method), 330
- OnInterfaceRemoval() (PySpin.PySpin.InterfaceRemovalEventHandler method), 365
- OnInterfaceRemoval() (PySpin.PySpin.ISystemEventHandler method), 339
- OnInterfaceRemoval() (PySpin.PySpin.SystemEventHandler method), 395
- OnInterfaceRemoval() (PySpin.SystemEventHandler method), 8
- OnLogEvent() (PySpin.LoggingEventHandler method), 8
- OnLogEvent() (PySpin.PySpin.ILoggingEventHandler method), 330
- OnLogEvent() (PySpin.PySpin.LoggingEventHandler method), 372
- Open() (PySpin.PySpin.SpinVideo method), 388
- Open() (PySpin.SpinVideo method), 69
- ## P
- PacketResendRequestCount (PySpin.Camera property), 29
- PacketResendRequestCount (PySpin.PySpin.Camera property), 134
- PacketResendRequestsDroppedCount (PySpin.Camera property), 29
- PacketResendRequestsDroppedCount (PySpin.PySpin.Camera property), 135
- PauseFrameCount (PySpin.Camera property), 29
- PauseFrameCount (PySpin.PySpin.Camera property), 135
- PayloadSize (PySpin.Camera property), 29
- PayloadSize (PySpin.PySpin.Camera property), 135
- PersistFeature() (PySpin.PySpin.CFeatureBag method), 97
- PersistFeature() (PySpin.PySpin.IPersistScript method), 334
- PGMOption (class in PySpin.PySpin), 382
- pixel (PySpin.PySpin.Stereo3DPoint property), 389
- pixel_value_max (PySpin.ChannelStatistics property), 42
- pixel_value_max (PySpin.PySpin.ChannelStatistics property), 148
- pixel_value_mean (PySpin.ChannelStatistics property), 42
- pixel_value_mean (PySpin.PySpin.ChannelStatistics property), 148
- pixel_value_min (PySpin.ChannelStatistics property), 42
- pixel_value_min (PySpin.PySpin.ChannelStatistics property), 148
- PixelColorFilter (PySpin.Camera property), 29
- PixelColorFilter (PySpin.PySpin.Camera property), 135
- PixelDynamicRangeMax (PySpin.Camera property), 29
- PixelDynamicRangeMax (PySpin.PySpin.Camera property), 135
- PixelDynamicRangeMin (PySpin.Camera property), 29
- PixelDynamicRangeMin (PySpin.PySpin.Camera property), 135
- PixelFormat (PySpin.Camera property), 29
- PixelFormat (PySpin.PySpin.Camera property), 135
- PixelFormatInfoID (PySpin.Camera property), 29
- PixelFormatInfoID (PySpin.PySpin.Camera property), 135
- PixelFormatInfoSelector (PySpin.Camera property), 29
- PixelFormatInfoSelector (PySpin.PySpin.Camera property), 135
- PixelFormatSize (PySpin.Camera property), 29
- PixelFormatSize (PySpin.PySpin.Camera property), 135
- PNGOption (class in PySpin.PySpin), 383
- POEStatus (PySpin.PySpin.TransportLayerInterface property), 399
- POEStatus (PySpin.TransportLayerInterface property), 78
- PointCloud (class in PySpin), 68
- PointCloud (class in PySpin.PySpin), 383
- PointCloudParameters (class in PySpin.PySpin), 384
- Poll() (PySpin.PySpin.CNodeMapDynPtr method), 103
- Poll() (PySpin.PySpin.CNodeMapPtr method), 104
- Poll() (PySpin.PySpin.INodeMap method), 332
- Poll() (PySpin.PySpin.NodeMap method), 382
- pop_back() (PySpin.PySpin.node_vector method), 409
- pop_back() (PySpin.PySpin.value_vector method), 411
- PowerSupplyCurrent (PySpin.Camera property), 29
- PowerSupplyCurrent (PySpin.PySpin.Camera property), 135
- PowerSupplyVoltage (PySpin.Camera property), 29
- PowerSupplyVoltage (PySpin.PySpin.Camera property), 135
- PPMOption (class in PySpin.PySpin), 383
- PreprocessXMLFromFile() (PySpin.PySpin.CNodeMapDynPtr method), 103
- PreprocessXMLFromFile() (PySpin.PySpin.INodeMapDyn method), 333
- PreprocessXMLFromZIPFile()

(*PySpin.PySpin.CNodeMapDynPtr* method), 103
 PreprocessXMLFromZIPFile() (*PySpin.PySpin.INodeMapDyn* method), 333
 principalPointU (*PySpin.PySpin.StereoCameraParameters* property), 389
 principalPointV (*PySpin.PySpin.StereoCameraParameters* property), 389
 PrintPoints() (*PySpin.PointCloud* method), 68
 PrintPoints() (*PySpin.PySpin.IPointCloud* method), 334
 PrintPoints() (*PySpin.PySpin.PointCloud* method), 384
 progressive (*PySpin.PySpin.JPEGOption* property), 370
 push_back() (*PySpin.PySpin.node_vector* method), 409
 push_back() (*PySpin.PySpin.value_vector* method), 411
 PySpin.PySpin module, 81

Q

quality (*PySpin.PySpin.JPEGOption* property), 370
 quality (*PySpin.PySpin.JPG2Option* property), 371
 quality (*PySpin.PySpin.MJPGOption* property), 372

R

r (*PySpin.PySpin.Stereo3DPoint* property), 389
 radius (*PySpin.PySpin.InferenceBoxCircle* property), 361
 range_max (*PySpin.ChannelStatistics* property), 42
 range_max (*PySpin.PySpin.ChannelStatistics* property), 148
 range_min (*PySpin.ChannelStatistics* property), 42
 range_min (*PySpin.PySpin.ChannelStatistics* property), 148
 ReadPort() (*PySpin.PySpin.ICameraBase* method), 169
 rect (*PySpin.PySpin.InferenceBoundingBox* property), 360
 RegionDestination (*PySpin.Camera* property), 29
 RegionDestination (*PySpin.PySpin.Camera* property), 135
 RegionMode (*PySpin.Camera* property), 30
 RegionMode (*PySpin.PySpin.Camera* property), 135
 RegionSelector (*PySpin.Camera* property), 30
 RegionSelector (*PySpin.PySpin.Camera* property), 135
 RegisterCallback() (*PySpin.PySpin.CBooleanPtr* method), 85
 RegisterCallback() (*PySpin.PySpin.CCategoryPtr* method), 88
 RegisterCallback() (*PySpin.PySpin.CCommandPtr* method), 90
 RegisterCallback() (*PySpin.PySpin.CEnumEntryPtr* method), 93
 RegisterCallback() (*PySpin.PySpin.CEnumerationPtr* method), 96
 RegisterCallback() (*PySpin.PySpin.CIntegerPtr* method), 101
 RegisterCallback() (*PySpin.PySpin.CNodePtr* method), 106
 RegisterCallback() (*PySpin.PySpin.CRegisterPtr* method), 109
 RegisterCallback() (*PySpin.PySpin.CStringPtr* method), 112
 RegisterCallback() (*PySpin.PySpin.CValuePtr* method), 115
 RegisterCallback() (*PySpin.PySpin.INode* method), 331
 RegisterCallback() (*PySpin.PySpin.Node* method), 377
 RegisterEventHandler() (*PySpin.CameraBase* method), 38
 RegisterEventHandler() (*PySpin.IInterface* method), 66
 RegisterEventHandler() (*PySpin.PySpin.CameraBase* method), 144
 RegisterEventHandler() (*PySpin.PySpin.ICameraBase* method), 169
 RegisterEventHandler() (*PySpin.PySpin.IInterface* method), 328
 RegisterEventHandler() (*PySpin.PySpin.ISystem* method), 338
 RegisterEventHandler() (*PySpin.PySpin.System* method), 393
 RegisterEventHandler() (*PySpin.System* method), 72
 RegisterLoggingEventHandler() (*PySpin.PySpin.ISystem* method), 338
 RegisterLoggingEventHandler() (*PySpin.PySpin.System* method), 393
 RegisterLoggingEventHandler() (*PySpin.System* method), 72
 RegisterNode (class in *PySpin.PySpin*), 384
 RegisterNodeCallback() (in module *PySpin.PySpin*), 386
 Release() (*PySpin.Image* method), 52
 Release() (*PySpin.ImageList* method), 56
 Release() (*PySpin.PySpin.IImage* method), 322
 Release() (*PySpin.PySpin.IImageList* method), 325
 Release() (*PySpin.PySpin.Image* method), 345
 Release() (*PySpin.PySpin.ImageList* method), 349
 ReleaseInstance() (*PySpin.PySpin.ISystem* method), 338
 ReleaseInstance() (*PySpin.PySpin.System* method), 393
 ReleaseInstance() (*PySpin.System* method), 72
 Remove() (*PySpin.CameraList* method), 40

- Remove() (*PySpin.InterfaceList* method), 68
- Remove() (*PySpin.PySpin.CameraList* method), 146
- Remove() (*PySpin.PySpin.ICameraList* method), 171
- Remove() (*PySpin.PySpin.IInterfaceList* method), 329
- Remove() (*PySpin.PySpin.InterfaceList* method), 365
- RemoveByDeviceID() (*PySpin.CameraList* method), 41
- RemoveByDeviceID() (*PySpin.PySpin.CameraList* method), 146
- RemoveByDeviceID() (*PySpin.PySpin.ICameraList* method), 171
- RemoveByIndex() (*PySpin.CameraList* method), 41
- RemoveByIndex() (*PySpin.ImageList* method), 56
- RemoveByIndex() (*PySpin.PySpin.CameraList* method), 146
- RemoveByIndex() (*PySpin.PySpin.ICameraList* method), 171
- RemoveByIndex() (*PySpin.PySpin.IImageList* method), 325
- RemoveByIndex() (*PySpin.PySpin.ImageList* method), 349
- RemoveByPayloadType() (*PySpin.ImageList* method), 56
- RemoveByPayloadType() (*PySpin.PySpin.IImageList* method), 325
- RemoveByPayloadType() (*PySpin.PySpin.ImageList* method), 349
- RemoveByPixelFormat() (*PySpin.ImageList* method), 56
- RemoveByPixelFormat() (*PySpin.PySpin.IImageList* method), 325
- RemoveByPixelFormat() (*PySpin.PySpin.ImageList* method), 349
- RemoveBySerial() (*PySpin.CameraList* method), 41
- RemoveBySerial() (*PySpin.PySpin.CameraList* method), 146
- RemoveBySerial() (*PySpin.PySpin.ICameraList* method), 171
- RemoveByStreamIndex() (*PySpin.ImageList* method), 56
- RemoveByStreamIndex() (*PySpin.PySpin.IImageList* method), 325
- RemoveByStreamIndex() (*PySpin.PySpin.ImageList* method), 349
- ReplaceEnvironmentVariables() (in module *PySpin.PySpin*), 386
- reserve() (*PySpin.PySpin.node_vector* method), 409
- reserve() (*PySpin.PySpin.value_vector* method), 411
- reserved (*PySpin.PySpin.AVIOption* property), 81
- reserved (*PySpin.PySpin.BMPOption* property), 81
- reserved (*PySpin.PySpin.H264Option* property), 168
- reserved (*PySpin.PySpin.JPEGOption* property), 370
- reserved (*PySpin.PySpin.JPG2Option* property), 371
- reserved (*PySpin.PySpin.MJPGOption* property), 372
- reserved (*PySpin.PySpin.PGMOption* property), 383
- reserved (*PySpin.PySpin.PNGOption* property), 383
- reserved (*PySpin.PySpin.PPMOption* property), 383
- reserved (*PySpin.PySpin.SIOption* property), 386
- reserved (*PySpin.PySpin.TIFFOption* property), 395
- ResetImage() (*PySpin.Image* method), 52
- ResetImage() (*PySpin.PySpin.IImage* method), 322
- ResetImage() (*PySpin.PySpin.Image* method), 345
- resize() (*PySpin.PySpin.gcstring* method), 407
- resize() (*PySpin.PySpin.node_vector* method), 409
- resize() (*PySpin.PySpin.value_vector* method), 411
- Restore() (*PySpin.PySpin.CSelectorSet* method), 110
- Restore() (*PySpin.PySpin.ISelectorDigit* method), 337
- result (*PySpin.PySpin.DeviceEventInferenceData* property), 154
- ReverseX (*PySpin.Camera* property), 30
- ReverseX (*PySpin.PySpin.Camera* property), 135
- ReverseY (*PySpin.Camera* property), 30
- ReverseY (*PySpin.PySpin.Camera* property), 135
- RgbTransformLightSource (*PySpin.Camera* property), 30
- RgbTransformLightSource (*PySpin.PySpin.Camera* property), 135
- ROIImageBottom (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIImageLeft (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIImageRight (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIImageTop (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesXMax (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesXMin (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesYMax (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesYMin (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesZMax (*PySpin.PySpin.PointCloudParameters* property), 384
- ROIWorldCoordinatesZMin (*PySpin.PySpin.PointCloudParameters* property), 384
- rotatedRect (*PySpin.PySpin.InferenceBoundingBox* property), 360
- rotationAngle (*PySpin.PySpin.InferenceBoxRotatedRect* property), 361

S

- Saturation (*PySpin.Camera* property), 30
- Saturation (*PySpin.PySpin.Camera* property), 135
- SaturationEnable (*PySpin.Camera* property), 30
- SaturationEnable (*PySpin.PySpin.Camera* property), 135
- Save() (*PySpin.Image* method), 53
- Save() (*PySpin.ImageList* method), 56
- Save() (*PySpin.PySpin.IImage* method), 323
- Save() (*PySpin.PySpin.IImageList* method), 325
- Save() (*PySpin.PySpin.Image* method), 347
- Save() (*PySpin.PySpin.ImageList* method), 350
- SavePointCloudAsPly() (*PySpin.PointCloud* method), 68
- SavePointCloudAsPly() (*PySpin.PySpin.IPointCloud* method), 334
- SavePointCloudAsPly() (*PySpin.PySpin.PointCloud* method), 384
- Scan3dAxisMax (*PySpin.Camera* property), 30
- Scan3dAxisMax (*PySpin.PySpin.Camera* property), 135
- Scan3dAxisMin (*PySpin.Camera* property), 30
- Scan3dAxisMin (*PySpin.PySpin.Camera* property), 135
- Scan3dBaseline (*PySpin.Camera* property), 30
- Scan3dBaseline (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateOffset (*PySpin.Camera* property), 30
- Scan3dCoordinateOffset (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateReferenceSelector (*PySpin.Camera* property), 30
- Scan3dCoordinateReferenceSelector (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateReferenceValue (*PySpin.Camera* property), 30
- Scan3dCoordinateReferenceValue (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateScale (*PySpin.Camera* property), 30
- Scan3dCoordinateScale (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateSelector (*PySpin.Camera* property), 30
- Scan3dCoordinateSelector (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateSystem (*PySpin.Camera* property), 30
- Scan3dCoordinateSystem (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateSystemReference (*PySpin.Camera* property), 30
- Scan3dCoordinateSystemReference (*PySpin.PySpin.Camera* property), 135
- Scan3dCoordinateTransformSelector (*PySpin.Camera* property), 30
- Scan3dCoordinateTransformSelector (*PySpin.PySpin.Camera* property), 135
- Scan3dDistanceUnit (*PySpin.Camera* property), 30
- Scan3dDistanceUnit (*PySpin.PySpin.Camera* property), 136
- Scan3dFocalLength (*PySpin.Camera* property), 30
- Scan3dFocalLength (*PySpin.PySpin.Camera* property), 136
- Scan3dInvalidDataFlag (*PySpin.Camera* property), 30
- Scan3dInvalidDataFlag (*PySpin.PySpin.Camera* property), 136
- Scan3dInvalidDataValue (*PySpin.Camera* property), 30
- Scan3dInvalidDataValue (*PySpin.PySpin.Camera* property), 136
- Scan3dOutputMode (*PySpin.Camera* property), 30
- Scan3dOutputMode (*PySpin.PySpin.Camera* property), 136
- Scan3dPrincipalPointU (*PySpin.Camera* property), 30
- Scan3dPrincipalPointU (*PySpin.PySpin.Camera* property), 136
- Scan3dPrincipalPointV (*PySpin.Camera* property), 30
- Scan3dPrincipalPointV (*PySpin.PySpin.Camera* property), 136
- Scan3dTransformValue (*PySpin.Camera* property), 30
- Scan3dTransformValue (*PySpin.PySpin.Camera* property), 136
- SendActionCommand() (*PySpin.IInterface* method), 66
- SendActionCommand() (*PySpin.PySpin.IInterface* method), 328
- SendActionCommand() (*PySpin.PySpin.ISystem* method), 338
- SendActionCommand() (*PySpin.PySpin.System* method), 393
- SendActionCommand() (*PySpin.System* method), 72
- Sensor (*PySpin.PySpin.CCMSettings* property), 86
- SensorDescription (*PySpin.Camera* property), 30
- SensorDescription (*PySpin.PySpin.Camera* property), 136
- SensorDigitizationTaps (*PySpin.Camera* property), 30
- SensorDigitizationTaps (*PySpin.PySpin.Camera* property), 136
- SensorHeight (*PySpin.Camera* property), 30
- SensorHeight (*PySpin.PySpin.Camera* property), 136
- SensorShutterMode (*PySpin.Camera* property), 30
- SensorShutterMode (*PySpin.PySpin.Camera* property), 136
- SensorTaps (*PySpin.Camera* property), 30

- SensorTaps (*PySpin.PySpin.Camera* property), 136
- SensorToString() (*PySpin.ImageUtilityCCM* static method), 60
- SensorToString() (*PySpin.PySpin.ImageUtilityCCM* static method), 354
- SensorWidth (*PySpin.Camera* property), 31
- SensorWidth (*PySpin.PySpin.Camera* property), 136
- SequencerConfigurationMode (*PySpin.Camera* property), 31
- SequencerConfigurationMode (*PySpin.PySpin.Camera* property), 136
- SequencerConfigurationReset (*PySpin.Camera* property), 31
- SequencerConfigurationReset (*PySpin.PySpin.Camera* property), 136
- SequencerConfigurationValid (*PySpin.Camera* property), 31
- SequencerConfigurationValid (*PySpin.PySpin.Camera* property), 136
- SequencerFeatureEnable (*PySpin.Camera* property), 31
- SequencerFeatureEnable (*PySpin.PySpin.Camera* property), 136
- SequencerMode (*PySpin.Camera* property), 31
- SequencerMode (*PySpin.PySpin.Camera* property), 136
- SequencerPathSelector (*PySpin.Camera* property), 31
- SequencerPathSelector (*PySpin.PySpin.Camera* property), 136
- SequencerSetActive (*PySpin.Camera* property), 31
- SequencerSetActive (*PySpin.PySpin.Camera* property), 136
- SequencerSetLoad (*PySpin.Camera* property), 31
- SequencerSetLoad (*PySpin.PySpin.Camera* property), 136
- SequencerSetNext (*PySpin.Camera* property), 31
- SequencerSetNext (*PySpin.PySpin.Camera* property), 136
- SequencerSetSave (*PySpin.Camera* property), 31
- SequencerSetSave (*PySpin.PySpin.Camera* property), 136
- SequencerSetSelector (*PySpin.Camera* property), 31
- SequencerSetSelector (*PySpin.PySpin.Camera* property), 136
- SequencerSetStart (*PySpin.Camera* property), 31
- SequencerSetStart (*PySpin.PySpin.Camera* property), 136
- SequencerSetValid (*PySpin.Camera* property), 31
- SequencerSetValid (*PySpin.PySpin.Camera* property), 136
- SequencerTriggerActivation (*PySpin.Camera* property), 31
- SequencerTriggerActivation (*PySpin.PySpin.Camera* property), 136
- SequencerTriggerSource (*PySpin.Camera* property), 31
- SequencerTriggerSource (*PySpin.PySpin.Camera* property), 136
- SerialPortBaudRate (*PySpin.Camera* property), 31
- SerialPortBaudRate (*PySpin.PySpin.Camera* property), 136
- SerialPortDataBits (*PySpin.Camera* property), 31
- SerialPortDataBits (*PySpin.PySpin.Camera* property), 136
- SerialPortParity (*PySpin.Camera* property), 31
- SerialPortParity (*PySpin.PySpin.Camera* property), 137
- SerialPortSelector (*PySpin.Camera* property), 31
- SerialPortSelector (*PySpin.PySpin.Camera* property), 137
- SerialPortSource (*PySpin.Camera* property), 31
- SerialPortSource (*PySpin.PySpin.Camera* property), 137
- SerialPortStopBits (*PySpin.Camera* property), 31
- SerialPortStopBits (*PySpin.PySpin.Camera* property), 137
- SerialReceiveFramingErrorCount (*PySpin.Camera* property), 31
- SerialReceiveFramingErrorCount (*PySpin.PySpin.Camera* property), 137
- SerialReceiveParityErrorCount (*PySpin.Camera* property), 31
- SerialReceiveParityErrorCount (*PySpin.PySpin.Camera* property), 137
- SerialReceiveQueueClear (*PySpin.Camera* property), 31
- SerialReceiveQueueClear (*PySpin.PySpin.Camera* property), 137
- SerialReceiveQueueCurrentCharacterCount (*PySpin.Camera* property), 31
- SerialReceiveQueueCurrentCharacterCount (*PySpin.PySpin.Camera* property), 137
- SerialReceiveQueueMaxCharacterCount (*PySpin.Camera* property), 31
- SerialReceiveQueueMaxCharacterCount (*PySpin.PySpin.Camera* property), 137
- SerialTransmitQueueCurrentCharacterCount (*PySpin.Camera* property), 31
- SerialTransmitQueueCurrentCharacterCount (*PySpin.PySpin.Camera* property), 137
- SerialTransmitQueueMaxCharacterCount (*PySpin.Camera* property), 31
- SerialTransmitQueueMaxCharacterCount (*PySpin.PySpin.Camera* property), 137
- Set() (*PySpin.PySpin.CRegisterPtr* method), 109
- Set() (*PySpin.PySpin.IRegister* method), 336
- Set() (*PySpin.PySpin.RegisterNode* method), 385
- SetBufferOwnership() (*PySpin.CameraBase* method),

39			
SetBufferOwnership()	(PySpin.PySpin.CameraBase method), 144	SetMaximumFileSize()	(PySpin.SpinVideo method), 388
SetBufferOwnership()	(PySpin.PySpin.ICameraBase method), 170	SetMessageCallback()	(in module PySpin.PySpin), 387
SetChunks()	(PySpin.ChunkData method), 46	SetNext()	(PySpin.PySpin.CSelectorSet method), 110
SetChunks()	(PySpin.PySpin.ChunkData method), 151	SetNext()	(PySpin.PySpin.ISelectorDigit method), 337
SetChunks()	(PySpin.PySpin.IChunkData method), 173	SetNodeHandle()	(PySpin.PySpin.Node method), 377
SetColorProcessing()	(PySpin.ImageProcessor method), 57	SetNodeMap()	(PySpin.PySpin.Node method), 377
SetColorProcessing()	(PySpin.PySpin.IImageProcessor method), 327	SetNumDecompressionThreads()	(PySpin.ImageProcessor method), 57
SetColorProcessing()	(PySpin.PySpin.ImageProcessor method), 351	SetNumDecompressionThreads()	(PySpin.PySpin.IImageProcessor method), 327
SetEnumReference()	(PySpin.PySpin.IEnumReference method), 175	SetNumDecompressionThreads()	(PySpin.PySpin.ImageProcessor method), 351
SetEventType()	(PySpin.EventHandler method), 6	SetNumEnums()	(PySpin.PySpin.IEnumReference method), 176
SetEventType()	(PySpin.PySpin.EventHandler method), 163	SetProgressCallback()	(in module PySpin.PySpin), 387
SetFirst()	(PySpin.PySpin.CSelectorSet method), 110	SetReference()	(PySpin.PySpin.BooleanNode method), 82
SetFirst()	(PySpin.PySpin.ISelectorDigit method), 337	SetReference()	(PySpin.PySpin.CategoryNode method), 147
SetGenICamCacheFolder()	(in module PySpin.PySpin), 386	SetReference()	(PySpin.PySpin.CBooleanPtr method), 85
SetGenICamCLProtocolFolder()	(in module PySpin.PySpin), 386	SetReference()	(PySpin.PySpin.CCategoryPtr method), 88
SetGenICamLogConfig()	(in module PySpin.PySpin), 386	SetReference()	(PySpin.PySpin.CCommandPtr method), 90
SetHeatmapColorGradient()	(PySpin.ImageUtilityHeatmap static method), 61	SetReference()	(PySpin.PySpin.CEnumEntryPtr method), 94
SetHeatmapColorGradient()	(PySpin.PySpin.ImageUtilityHeatmap static method), 355	SetReference()	(PySpin.PySpin.CEnumerationPtr method), 96
SetHeatmapRange()	(PySpin.ImageUtilityHeatmap static method), 61	SetReference()	(PySpin.PySpin.CIntegerPtr method), 101
SetHeatmapRange()	(PySpin.PySpin.ImageUtilityHeatmap static method), 355	SetReference()	(PySpin.PySpin.CNodePtr method), 106
SetInfo()	(PySpin.PySpin.CFeatureBag method), 97	SetReference()	(PySpin.PySpin.CommandNode method), 153
SetInfo()	(PySpin.PySpin.IPersistScript method), 334	SetReference()	(PySpin.PySpin.CRegisterPtr method), 109
SetIntValue()	(PySpin.PySpin.CEnumerationPtr method), 96	SetReference()	(PySpin.PySpin.CStringPtr method), 113
SetIntValue()	(PySpin.PySpin.EnumNode method), 162	SetReference()	(PySpin.PySpin.CValuePtr method), 115
SetIntValue()	(PySpin.PySpin.IEnumeration method), 176	SetReference()	(PySpin.PySpin.EnumEntryNode method), 161
SetLoggingEventPriorityLevel()	(PySpin.PySpin.ISystem method), 338	SetReference()	(PySpin.PySpin.EnumNode method), 162
SetLoggingEventPriorityLevel()	(PySpin.PySpin.System method), 394	SetReference()	(PySpin.PySpin.FloatNode method), 165
SetLoggingEventPriorityLevel()	(PySpin.System method), 73	SetReference()	(PySpin.PySpin.FloatRegNode method), 388
SetMaximumFileSize()	(PySpin.PySpin.SpinVideo method), 388		

- method*), 166
- SetReference() (PySpin.PySpin.IntegerNode *method*), 363
- SetReference() (PySpin.PySpin.IntRegNode *method*), 361
- SetReference() (PySpin.PySpin.IReference *method*), 335
- SetReference() (PySpin.PySpin.Node *method*), 377
- SetReference() (PySpin.PySpin.RegisterNode *method*), 385
- SetReference() (PySpin.PySpin.StringNode *method*), 390
- SetReference() (PySpin.PySpin.StringRegNode *method*), 391
- SetReference() (PySpin.PySpin.ValueNode *method*), 403
- SetUserBuffers() (PySpin.CameraBase *method*), 39
- SetUserBuffers() (PySpin.PySpin.CameraBase *method*), 144
- SetUserBuffers() (PySpin.PySpin.ICameraBase *method*), 170
- SetValue() (PySpin.PySpin.BooleanNode *method*), 82
- SetValue() (PySpin.PySpin.CBooleanPtr *method*), 85
- SetValue() (PySpin.PySpin.CIntegerPtr *method*), 101
- SetValue() (PySpin.PySpin.CStringPtr *method*), 113
- SetValue() (PySpin.PySpin.FloatNode *method*), 165
- SetValue() (PySpin.PySpin.IBoolean *method*), 168
- SetValue() (PySpin.PySpin.IEnumerationT_AcquisitionModeEnums *method*), 177
- SetValue() (PySpin.PySpin.IEnumerationT_AcquisitionStatusEnums *method*), 178
- SetValue() (PySpin.PySpin.IEnumerationT_ActionSelectorEnums *method*), 178
- SetValue() (PySpin.PySpin.IEnumerationT_ActionUnconditionalEnums *method*), 179
- SetValue() (PySpin.PySpin.IEnumerationT_AdcBitDepthEnums *method*), 180
- SetValue() (PySpin.PySpin.IEnumerationT_AutoAlgorithmEnums *method*), 180
- SetValue() (PySpin.PySpin.IEnumerationT_AutoExposureSelectorEnums *method*), 181
- SetValue() (PySpin.PySpin.IEnumerationT_AutoExposureSegmentEnums *method*), 182
- SetValue() (PySpin.PySpin.IEnumerationT_AutoExposureSegmentOrderEnums *method*), 182
- SetValue() (PySpin.PySpin.IEnumerationT_AutoExposureSegmentOrderEnums *method*), 183
- SetValue() (PySpin.PySpin.IEnumerationT_BalanceRatioEnums *method*), 184
- SetValue() (PySpin.PySpin.IEnumerationT_BalanceWhiteEnums *method*), 184
- SetValue() (PySpin.PySpin.IEnumerationT_BalanceWhiteEnums *method*), 185
- SetValue() (PySpin.PySpin.IEnumerationT_BinningHorizontalEnums *method*), 186
- SetValue() (PySpin.PySpin.IEnumerationT_BinningSelectorEnums *method*), 186
- SetValue() (PySpin.PySpin.IEnumerationT_BinningVerticalModeEnums *method*), 187
- SetValue() (PySpin.PySpin.IEnumerationT_BlackLevelAutoBalanceEnums *method*), 188
- SetValue() (PySpin.PySpin.IEnumerationT_BlackLevelAutoEnums *method*), 188
- SetValue() (PySpin.PySpin.IEnumerationT_BlackLevelSelectorEnums *method*), 189
- SetValue() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums *method*), 190
- SetValue() (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums *method*), 190
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums *method*), 191
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkCounterSelectorEnums *method*), 192
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkEncoderSelectorEnums *method*), 192
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkEncoderStatusEnums *method*), 193
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums *method*), 194
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkGainSelectorEnums *method*), 194
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkImageComponentEnums *method*), 195
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkPixelFormatEnums *method*), 196
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkRegionIDEnums *method*), 196
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateReferenceEnums *method*), 197
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSelectorEnums *method*), 198
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystemEnums *method*), 198
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateSystemEnums *method*), 199
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateTransformEnums *method*), 200
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dDistanceUnitEnums *method*), 200
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkScan3dOutputModeEnums *method*), 201
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkSelectorEnums *method*), 202
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkSourceIDEnums *method*), 202
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkTimerSelectorEnums *method*), 203
- SetValue() (PySpin.PySpin.IEnumerationT_ChunkTransferStreamIDEnums *method*), 203

- method), 204
- SetValue() (PySpin.PySpin.IEnumerationT_ClConfiguratorEnums method), 204
- SetValue() (PySpin.PySpin.IEnumerationT_ClTimeSlotsCEnums method), 205
- SetValue() (PySpin.PySpin.IEnumerationT_ColorTransformEnums method), 206
- SetValue() (PySpin.PySpin.IEnumerationT_ColorTransformEnums method), 206
- SetValue() (PySpin.PySpin.IEnumerationT_ComponentDescriptorEnums method), 207
- SetValue() (PySpin.PySpin.IEnumerationT_ComponentSelectorEnums method), 208
- SetValue() (PySpin.PySpin.IEnumerationT_CompressionEnums method), 208
- SetValue() (PySpin.PySpin.IEnumerationT_CounterEventEnums method), 209
- SetValue() (PySpin.PySpin.IEnumerationT_CounterEventEnums method), 210
- SetValue() (PySpin.PySpin.IEnumerationT_CounterResetEnums method), 210
- SetValue() (PySpin.PySpin.IEnumerationT_CounterResetEnums method), 211
- SetValue() (PySpin.PySpin.IEnumerationT_CounterSelectorEnums method), 212
- SetValue() (PySpin.PySpin.IEnumerationT_CounterStatusEnums method), 212
- SetValue() (PySpin.PySpin.IEnumerationT_CounterTriggerEnums method), 213
- SetValue() (PySpin.PySpin.IEnumerationT_CounterTriggerEnums method), 214
- SetValue() (PySpin.PySpin.IEnumerationT_CxpConnectionEnums method), 214
- SetValue() (PySpin.PySpin.IEnumerationT_CxpLinkConfigEnums method), 215
- SetValue() (PySpin.PySpin.IEnumerationT_CxpLinkConfigEnums method), 216
- SetValue() (PySpin.PySpin.IEnumerationT_CxpLinkConfigEnums method), 216
- SetValue() (PySpin.PySpin.IEnumerationT_CxpPoCxpStatusEnums method), 217
- SetValue() (PySpin.PySpin.IEnumerationT_DecimationHorizontalEnums method), 218
- SetValue() (PySpin.PySpin.IEnumerationT_DecimationSelectorEnums method), 218
- SetValue() (PySpin.PySpin.IEnumerationT_DecimationVerticalEnums method), 219
- SetValue() (PySpin.PySpin.IEnumerationT_DefectCorrectionEnums method), 220
- SetValue() (PySpin.PySpin.IEnumerationT_DeinterlacingEnums method), 220
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceAccessEnums method), 221
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceCharacteristicsEnums method), 222
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceClockSelectorEnums method), 222
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceConnectionStatusEnums method), 223
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceCurrentSpeedEnums method), 224
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceEndiannessMechanismEnums method), 224
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceIndicatorModeEnums method), 225
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnums method), 226
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceLinkThroughputLimitEnums method), 226
- SetValue() (PySpin.PySpin.IEnumerationT_DevicePowerSupplySelectorEnums method), 227
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceRegistersEndiannessEnums method), 228
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceScanTypeEnums method), 228
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceSensorChromaEnums method), 229
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceSerialPortBaudRateEnums method), 230
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceSerialPortSelectorEnums method), 230
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceStreamChannelEndiannessEnums method), 231
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceStreamChannelTypeEnums method), 232
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceTapGeometryEnums method), 233
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceTemperatureSelectorEnums method), 234
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceTLTypeEnums method), 232
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceTypeEnum method), 234
- SetValue() (PySpin.PySpin.IEnumerationT_DeviceTypeEnum method), 235
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderModeEnums method), 236
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderOutputModeEnums method), 236
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderResetActivationEnums method), 237
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderResetSourceEnums method), 238
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderSelectorEnums method), 238
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderSourceAEnums method), 239
- SetValue() (PySpin.PySpin.IEnumerationT_EncoderSourceBEnums method), 239

<i>method</i>), 240	<i>method</i>), 258
SetValue() (PySpin.PySpin.IEnumerationT_EncoderStatusEnums, <i>method</i>), 240	SetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusEnums, <i>method</i>), 259
SetValue() (PySpin.PySpin.IEnumerationT_EventNotificationEnums, <i>method</i>), 241	SetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums, <i>method</i>), 260
SetValue() (PySpin.PySpin.IEnumerationT_EventSelectorEnums, <i>method</i>), 242	SetValue() (PySpin.PySpin.IEnumerationT_GevIPConfigurationStatusEnums, <i>method</i>), 260
SetValue() (PySpin.PySpin.IEnumerationT_ExposureActiveEnums, <i>method</i>), 242	SetValue() (PySpin.PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnums, <i>method</i>), 261
SetValue() (PySpin.PySpin.IEnumerationT_ExposureAutoEnums, <i>method</i>), 243	SetValue() (PySpin.PySpin.IEnumerationT_GevSCPDirectionEnums, <i>method</i>), 262
SetValue() (PySpin.PySpin.IEnumerationT_ExposureModeEnums, <i>method</i>), 244	SetValue() (PySpin.PySpin.IEnumerationT_GevSupportedOptionSelectorEnums, <i>method</i>), 262
SetValue() (PySpin.PySpin.IEnumerationT_ExposureTimeEnums, <i>method</i>), 244	SetValue() (PySpin.PySpin.IEnumerationT_GUIXMLLocationEnums, <i>method</i>), 250
SetValue() (PySpin.PySpin.IEnumerationT_ExposureTimeUnits, <i>method</i>), 245	SetValue() (PySpin.PySpin.IEnumerationT_ImageComponentSelectorEnums, <i>method</i>), 263
SetValue() (PySpin.PySpin.IEnumerationT_ExternalVoltageEnums, <i>method</i>), 246	SetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionJPEGFormatEnums, <i>method</i>), 264
SetValue() (PySpin.PySpin.IEnumerationT_FfcModeEnums, <i>method</i>), 247	SetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionModeEnums, <i>method</i>), 264
SetValue() (PySpin.PySpin.IEnumerationT_FileOpenModeEnums, <i>method</i>), 248	SetValue() (PySpin.PySpin.IEnumerationT_ImageCompressionRateOptions, <i>method</i>), 265
SetValue() (PySpin.PySpin.IEnumerationT_FileOperationEnums, <i>method</i>), 248	SetValue() (PySpin.PySpin.IEnumerationT_InterfaceTypeEnum, <i>method</i>), 266
SetValue() (PySpin.PySpin.IEnumerationT_FileOperationUnits, <i>method</i>), 249	SetValue() (PySpin.PySpin.IEnumerationT_LensShadingCoefficientActiveEnums, <i>method</i>), 267
SetValue() (PySpin.PySpin.IEnumerationT_FileSelectorEnums, <i>method</i>), 250	SetValue() (PySpin.PySpin.IEnumerationT_LensShadingCorrectionModes, <i>method</i>), 268
SetValue() (PySpin.PySpin.IEnumerationT_FLIRFilterDriverEnums, <i>method</i>), 246	SetValue() (PySpin.PySpin.IEnumerationT_LineFormatEnums, <i>method</i>), 268
SetValue() (PySpin.PySpin.IEnumerationT_GainAutoBalanceEnums, <i>method</i>), 251	SetValue() (PySpin.PySpin.IEnumerationT_LineInputFilterSelectorEnums, <i>method</i>), 269
SetValue() (PySpin.PySpin.IEnumerationT_GainAutoEnums, <i>method</i>), 252	SetValue() (PySpin.PySpin.IEnumerationT_LineModeEnums, <i>method</i>), 270
SetValue() (PySpin.PySpin.IEnumerationT_GainConversionEnums, <i>method</i>), 252	SetValue() (PySpin.PySpin.IEnumerationT_LineSelectorEnums, <i>method</i>), 270
SetValue() (PySpin.PySpin.IEnumerationT_GainSelectorEnums, <i>method</i>), 253	SetValue() (PySpin.PySpin.IEnumerationT_LineSourceEnums, <i>method</i>), 271
SetValue() (PySpin.PySpin.IEnumerationT_GenICamXMLEnums, <i>method</i>), 254	SetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputActivationEnums, <i>method</i>), 272
SetValue() (PySpin.PySpin.IEnumerationT_GevCCPEnums, <i>method</i>), 254	SetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputSelectorEnums, <i>method</i>), 272
SetValue() (PySpin.PySpin.IEnumerationT_GevCCPEnums, <i>method</i>), 255	SetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTInputSourceEnums, <i>method</i>), 273
SetValue() (PySpin.PySpin.IEnumerationT_GevCurrentPhaseEnums, <i>method</i>), 256	SetValue() (PySpin.PySpin.IEnumerationT_LogicBlockLUTSelectorEnums, <i>method</i>), 274
SetValue() (PySpin.PySpin.IEnumerationT_GevGVCPEnums, <i>method</i>), 256	SetValue() (PySpin.PySpin.IEnumerationT_LogicBlockSelectorEnums, <i>method</i>), 274
SetValue() (PySpin.PySpin.IEnumerationT_GevGVSPExtEnums, <i>method</i>), 257	SetValue() (PySpin.PySpin.IEnumerationT_LUTSelectorEnums, <i>method</i>), 266
SetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusEnums, <i>method</i>), 258	SetValue() (PySpin.PySpin.IEnumerationT_MultiRoiConfigurationInvalidEnums, <i>method</i>), 275
SetValue() (PySpin.PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnums, <i>method</i>), 259	SetValue() (PySpin.PySpin.IEnumerationT_MultiRoiSelectorEnums, <i>method</i>), 275

- method), 276
- SetValue() (PySpin.PySpin.IEnumerationT_PixelColorFilterEnum), 277
- SetValue() (PySpin.PySpin.IEnumerationT_PixelFormatEnum), 278
- SetValue() (PySpin.PySpin.IEnumerationT_PixelFormatInEnum), 278
- SetValue() (PySpin.PySpin.IEnumerationT_PixelSizeEnum), 279
- SetValue() (PySpin.PySpin.IEnumerationT_POEStatusEnum), 276
- SetValue() (PySpin.PySpin.IEnumerationT_RegionDestinationEnum), 280
- SetValue() (PySpin.PySpin.IEnumerationT_RegionModeEnum), 280
- SetValue() (PySpin.PySpin.IEnumerationT_RegionSelectorEnum), 281
- SetValue() (PySpin.PySpin.IEnumerationT_RgbTransformEnum), 282
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnum), 282
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnum), 283
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnum), 284
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnum), 284
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dCoordinatesEnum), 285
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dDistanceEnum), 286
- SetValue() (PySpin.PySpin.IEnumerationT_Scan3dOutputsEnum), 286
- SetValue() (PySpin.PySpin.IEnumerationT_SensorDigitizationEnum), 287
- SetValue() (PySpin.PySpin.IEnumerationT_SensorShutterEnum), 288
- SetValue() (PySpin.PySpin.IEnumerationT_SensorTapsEnum), 288
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerConfigurationEnum), 289
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerConfigurationEnum), 290
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerModeEnum), 290
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerSetEnum), 291
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerTriggerEnum), 292
- SetValue() (PySpin.PySpin.IEnumerationT_SequencerTriggerEnum), 292
- SetValue() (PySpin.PySpin.IEnumerationT_SerialPortBaudRateEnum), 293
- SetValue() (PySpin.PySpin.IEnumerationT_SerialPortParametersEnum), 294
- SetValue() (PySpin.PySpin.IEnumerationT_SerialPortSelectorEnums), 294
- SetValue() (PySpin.PySpin.IEnumerationT_SerialPortSourceEnums), 295
- SetValue() (PySpin.PySpin.IEnumerationT_SerialPortStopBitsEnums), 296
- SetValue() (PySpin.PySpin.IEnumerationT_SoftwareSignalSelectorEnum), 296
- SetValue() (PySpin.PySpin.IEnumerationT_SourceSelectorEnums), 297
- SetValue() (PySpin.PySpin.IEnumerationT_StereoResolutionEnums), 298
- SetValue() (PySpin.PySpin.IEnumerationT_StreamBufferCountModeEnum), 298
- SetValue() (PySpin.PySpin.IEnumerationT_StreamBufferHandlingModeEnum), 299
- SetValue() (PySpin.PySpin.IEnumerationT_StreamModeEnum), 300
- SetValue() (PySpin.PySpin.IEnumerationT_StreamTypeEnum), 300
- SetValue() (PySpin.PySpin.IEnumerationT_TeledyneGigeVisionFilterDriverEnum), 302
- SetValue() (PySpin.PySpin.IEnumerationT_TestPatternEnums), 302
- SetValue() (PySpin.PySpin.IEnumerationT_TestPatternGeneratorSelectorEnum), 303
- SetValue() (PySpin.PySpin.IEnumerationT_TimerSelectorEnums), 304
- SetValue() (PySpin.PySpin.IEnumerationT_TimerStatusEnums), 304
- SetValue() (PySpin.PySpin.IEnumerationT_TimerTriggerActivationEnum), 305
- SetValue() (PySpin.PySpin.IEnumerationT_TimerTriggerSourceEnums), 306
- SetValue() (PySpin.PySpin.IEnumerationT_TLTypeEnum), 301
- SetValue() (PySpin.PySpin.IEnumerationT_TransferComponentSelectorEnum), 306
- SetValue() (PySpin.PySpin.IEnumerationT_TransferControlModeEnums), 307
- SetValue() (PySpin.PySpin.IEnumerationT_TransferOperationModeEnum), 308
- SetValue() (PySpin.PySpin.IEnumerationT_TransferQueueModeEnums), 308
- SetValue() (PySpin.PySpin.IEnumerationT_TransferSelectorEnums), 309
- SetValue() (PySpin.PySpin.IEnumerationT_TransferStatusSelectorEnums), 310
- SetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerActivationEnum), 310
- SetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerModeEnums), 311
- SetValue() (PySpin.PySpin.IEnumerationT_TransferTriggerSelectorEnum), 311

`method`), 312
`SetValue()` (`PySpin.PySpin.IEnumerationT_TransferTriggerSourceCount` `method`), 312
`SetValue()` (`PySpin.PySpin.IEnumerationT_TriggerActivationSourceCount` `method`), 313
`SetValue()` (`PySpin.PySpin.IEnumerationT_TriggerModeEnums` `method`), 314
`SetValue()` (`PySpin.PySpin.IEnumerationT_TriggerOverlapSpinUpdate` `method`), 314
`SetValue()` (`PySpin.PySpin.IEnumerationT_TriggerSelectorSpinUpdate` `method`), 315
`SetValue()` (`PySpin.PySpin.IEnumerationT_TriggerSourceSpinVideo` `method`), 316
`SetValue()` (`PySpin.PySpin.IEnumerationT_U3VCurrentStatus` `method`), 316
`SetValue()` (`PySpin.PySpin.IEnumerationT_UserOutputStereo53DPoint` `method`), 317
`SetValue()` (`PySpin.PySpin.IEnumerationT_UserSetDefaultEnums` `method`), 318
`SetValue()` (`PySpin.PySpin.IEnumerationT_UserSetSelectorStereoHeight` `method`), 318
`SetValue()` (`PySpin.PySpin.IEnumerationT_WhiteClipSelectorStereoResolution` `method`), 319
`SetValue()` (`PySpin.PySpin.IFloat` `method`), 320
`SetValue()` (`PySpin.PySpin.IInteger` `method`), 328
`SetValue()` (`PySpin.PySpin.IntegerNode` `method`), 363
`SetValue()` (`PySpin.PySpin.IString` `method`), 337
`SetValue()` (`PySpin.PySpin.StringNode` `method`), 390
`Sharpening` (`PySpin.Camera` `property`), 31
`Sharpening` (`PySpin.PySpin.Camera` `property`), 137
`SharpeningAuto` (`PySpin.Camera` `property`), 31
`SharpeningAuto` (`PySpin.PySpin.Camera` `property`), 137
`SharpeningEnable` (`PySpin.Camera` `property`), 32
`SharpeningEnable` (`PySpin.PySpin.Camera` `property`), 137
`SharpeningThreshold` (`PySpin.Camera` `property`), 32
`SharpeningThreshold` (`PySpin.PySpin.Camera` `property`), 137
`SIOption` (`class in PySpin.PySpin`), 386
`size()` (`PySpin.PySpin.double_autovector_t` `method`), 404
`size()` (`PySpin.PySpin.gcstring` `method`), 407
`size()` (`PySpin.PySpin.int64_autovector_t` `method`), 408
`size()` (`PySpin.PySpin.node_vector` `method`), 409
`size()` (`PySpin.PySpin.value_vector` `method`), 411
`SmallPenalty` (`PySpin.Camera` `property`), 32
`SmallPenalty` (`PySpin.PySpin.Camera` `property`), 137
`SoftwareSignalPulse` (`PySpin.Camera` `property`), 32
`SoftwareSignalPulse` (`PySpin.PySpin.Camera` `property`), 137
`SoftwareSignalSelector` (`PySpin.Camera` `property`), 32
`SoftwareSignalSelector` (`PySpin.PySpin.Camera` `property`), 137
`SourceCount` (`PySpin.PySpin.Camera` `property`), 32
`SourceCount` (`PySpin.PySpin.Camera` `property`), 137
`SourceSelector` (`PySpin.Camera` `property`), 32
`SourceSelector` (`PySpin.PySpin.Camera` `property`), 137
`SpinnakerException` (`class in PySpin`), 69
`SpinUpdate_SetMsgCallback()` (`in module PySpin.PySpin`), 387
`SpinUpdate_SetProgCallback()` (`in module PySpin.PySpin`), 387
`SpinVideo` (`class in PySpin`), 69
`SpinVideo` (`class in PySpin.PySpin`), 387
`Status` (`PySpin.PySpin.ActionCommandResult` `property`), 81
`Stereo53DPoint` (`class in PySpin.PySpin`), 389
`StereoCameraParameters` (`class in PySpin.PySpin`), 389
`StereoHeight` (`PySpin.Camera` `property`), 32
`StereoHeight` (`PySpin.PySpin.Camera` `property`), 137
`StereoResolution` (`PySpin.Camera` `property`), 32
`StereoResolution` (`PySpin.PySpin.Camera` `property`), 137
`StereoWidth` (`PySpin.Camera` `property`), 32
`StereoWidth` (`PySpin.PySpin.Camera` `property`), 137
`StoreToBag()` (`PySpin.PySpin.CFeatureBag` `method`), 98
`StreamAnnounceBufferMinimum` (`PySpin.PySpin.TransportLayerStream` `property`), 400
`StreamAnnounceBufferMinimum` (`PySpin.TransportLayerStream` `property`), 79
`StreamAnnouncedBufferCount` (`PySpin.PySpin.TransportLayerStream` `property`), 400
`StreamAnnouncedBufferCount` (`PySpin.TransportLayerStream` `property`), 79
`StreamBlocksProcessingTimeLast` (`PySpin.PySpin.TransportLayerStream` `property`), 400
`StreamBlocksProcessingTimeLast` (`PySpin.TransportLayerStream` `property`), 79
`StreamBlocksProcessingTimeMax` (`PySpin.PySpin.TransportLayerStream` `property`), 400
`StreamBlocksProcessingTimeMax` (`PySpin.TransportLayerStream` `property`), 79
`StreamBlocksProcessingTimeMin` (`PySpin.PySpin.TransportLayerStream` `property`), 400

StreamBlocksProcessingTimeMin (<i>PySpin.TransportLayerStream</i> property), 79	StreamBufferCountResult (<i>PySpin.TransportLayerStream</i> property), 79
StreamBlocksReceptionTimeLast (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamBufferHandlingMode (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBlocksReceptionTimeLast (<i>PySpin.TransportLayerStream</i> property), 79	StreamBufferHandlingMode (<i>PySpin.TransportLayerStream</i> property), 79
StreamBlocksReceptionTimeMax (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamChunkCountMaximum (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBlocksReceptionTimeMax (<i>PySpin.TransportLayerStream</i> property), 79	StreamChunkCountMaximum (<i>PySpin.TransportLayerStream</i> property), 79
StreamBlocksReceptionTimeMin (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamCRCCheckEnable (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBlocksReceptionTimeMin (<i>PySpin.TransportLayerStream</i> property), 79	StreamCRCCheckEnable (<i>PySpin.TransportLayerStream</i> property), 79
StreamBlockTransferSize (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamDeliveredFrameCount (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBlockTransferSize (<i>PySpin.TransportLayerStream</i> property), 79	StreamDeliveredFrameCount (<i>PySpin.TransportLayerStream</i> property), 79
StreamBufferAlignment (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamDroppedFrameCount (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBufferAlignment (<i>PySpin.TransportLayerStream</i> property), 79	StreamDroppedFrameCount (<i>PySpin.TransportLayerStream</i> property), 79
StreamBufferCountManual (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamID (<i>PySpin.PySpin.TransportLayerDevice</i> property), 397
StreamBufferCountManual (<i>PySpin.TransportLayerStream</i> property), 79	StreamID (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBufferCountMax (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamID (<i>PySpin.TransportLayerDevice</i> property), 76
StreamBufferCountMax (<i>PySpin.TransportLayerStream</i> property), 79	StreamID (<i>PySpin.TransportLayerStream</i> property), 79
StreamBufferCountMode (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamIncompleteFrameCount (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
StreamBufferCountMode (<i>PySpin.TransportLayerStream</i> property), 79	StreamIncompleteFrameCount (<i>PySpin.TransportLayerStream</i> property), 79
StreamBufferCountResult (<i>PySpin.PySpin.TransportLayerStream</i> property), 400	StreamInputBufferCount (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
	StreamInputBufferCount (<i>PySpin.TransportLayerStream</i> property), 79
	StreamIsGrabbing (<i>PySpin.PySpin.TransportLayerStream</i> property), 400
	StreamIsGrabbing (<i>PySpin.TransportLayerStream</i>

<i>property</i>), 79	<i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamLostFrameCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 400	StreamPacketResendRequestTimeoutCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamLostFrameCount <i>(PySpin.TransportLayerStream property)</i> , 79	StreamPacketResendTimeout <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamMissedPacketCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 400	StreamPacketResendTimeout <i>(PySpin.TransportLayerStream property)</i> , 80
StreamMissedPacketCount <i>(PySpin.TransportLayerStream property)</i> , 79	StreamPacketsDuplicatedCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamMode <i>(PySpin.PySpin.TransportLayerStream property)</i> , 400	StreamPacketsDuplicatedCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamMode <i>(PySpin.TransportLayerStream property)</i> , 79	StreamPacketsNotYetAvailableCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamOutputBufferCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 400	StreamPacketsNotYetAvailableCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamOutputBufferCount <i>(PySpin.TransportLayerStream property)</i> , 79	StreamPacketsPerFrameCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamPacketResendEnable <i>(PySpin.PySpin.TransportLayerStream property)</i> , 400	StreamPacketsPerFrameCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamPacketResendEnable <i>(PySpin.TransportLayerStream property)</i> , 79	StreamPacketsTemporarilyUnavailableCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamPacketResendMaxRequests <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401	StreamPacketsTemporarilyUnavailableCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamPacketResendMaxRequests <i>(PySpin.TransportLayerStream property)</i> , 80	StreamPacketsTimeoutCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamPacketResendReceivedPacketCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401	StreamPacketsTimeoutCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamPacketResendReceivedPacketCount <i>(PySpin.TransportLayerStream property)</i> , 80	StreamPacketsUnavailableCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamPacketResendRequestCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401	StreamPacketsUnavailableCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamPacketResendRequestCount <i>(PySpin.TransportLayerStream property)</i> , 80	StreamReceivedFrameCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401
StreamPacketResendRequestedPacketCount <i>(PySpin.PySpin.TransportLayerStream property)</i> , 401	StreamReceivedFrameCount <i>(PySpin.TransportLayerStream property)</i> , 80
StreamPacketResendRequestedPacketCount <i>(PySpin.TransportLayerStream property)</i> , 80	StreamReceivedPacketCount <i>(PySpin.PySpin.TransportLayerStream property)</i>
StreamPacketResendRequestTimeoutCount	

- erty), 401
- StreamReceivedPacketCount (PySpin.TransportLayerStream property), 80
- StreamSelector (PySpin.PySpin.TransportLayerDevice property), 397
- StreamSelector (PySpin.TransportLayerDevice property), 76
- StreamStartedFrameCount (PySpin.PySpin.TransportLayerStream property), 401
- StreamStartedFrameCount (PySpin.TransportLayerStream property), 80
- StreamType (PySpin.PySpin.TransportLayerStream property), 401
- StreamType (PySpin.TransportLayerStream property), 80
- StringNode (class in PySpin.PySpin), 389
- StringRegNode (class in PySpin.PySpin), 391
- SubMinor (PySpin.PySpin.Version_t property), 404
- substr() (PySpin.PySpin.gcstring method), 407
- swap() (PySpin.PySpin.gcstring method), 408
- System (class in PySpin), 70
- System (class in PySpin.PySpin), 391
- SystemEventHandler (class in PySpin), 8
- SystemEventHandler (class in PySpin.PySpin), 395
- SystemPtr (class in PySpin), 74
- SystemPtr (class in PySpin.PySpin), 395
- ## T
- TeledyneGigeVisionFilterDriverStatus (PySpin.PySpin.TransportLayerInterface property), 399
- TeledyneGigeVisionFilterDriverStatus (PySpin.TransportLayerInterface property), 78
- Test0001 (PySpin.Camera property), 32
- Test0001 (PySpin.PySpin.Camera property), 137
- TestEventGenerate (PySpin.Camera property), 32
- TestEventGenerate (PySpin.PySpin.Camera property), 137
- TestPattern (PySpin.Camera property), 32
- TestPattern (PySpin.PySpin.Camera property), 137
- TestPatternGeneratorSelector (PySpin.Camera property), 32
- TestPatternGeneratorSelector (PySpin.PySpin.Camera property), 137
- TestPendingAck (PySpin.Camera property), 32
- TestPendingAck (PySpin.PySpin.Camera property), 137
- thisown (PySpin.Camera property), 35
- thisown (PySpin.CameraBase property), 39
- thisown (PySpin.CameraList property), 41
- thisown (PySpin.CameraPtr property), 41
- thisown (PySpin.CBasePtr property), 10
- thisown (PySpin.ChannelStatistics property), 42
- thisown (PySpin.ChunkData property), 46
- thisown (PySpin.DeviceArrivalEventHandler property), 5
- thisown (PySpin.DeviceEventHandler property), 6
- thisown (PySpin.DeviceRemovalEventHandler property), 6
- thisown (PySpin.EventHandler property), 6
- thisown (PySpin.Interface property), 67
- thisown (PySpin.Image property), 55
- thisown (PySpin.ImageEventHandler property), 6
- thisown (PySpin.ImageList property), 56
- thisown (PySpin.ImageListEventHandler property), 7
- thisown (PySpin.ImageProcessor property), 57
- thisown (PySpin.ImagePtr property), 58
- thisown (PySpin.ImageUtility property), 59
- thisown (PySpin.ImageUtilityCCM property), 60
- thisown (PySpin.ImageUtilityHeatmap property), 61
- thisown (PySpin.ImageUtilityPolarization property), 64
- thisown (PySpin.ImageUtilityStereo property), 66
- thisown (PySpin.InterfaceArrivalEventHandler property), 7
- thisown (PySpin.InterfaceEventHandler property), 7
- thisown (PySpin.InterfaceList property), 68
- thisown (PySpin.InterfacePtr property), 68
- thisown (PySpin.InterfaceRemovalEventHandler property), 7
- thisown (PySpin.LoggingEventDataPtr property), 8
- thisown (PySpin.LoggingEventHandler property), 8
- thisown (PySpin.PointCloud property), 69
- thisown (PySpin.PySpin.ActionCommandResult property), 81
- thisown (PySpin.PySpin.AVIOption property), 81
- thisown (PySpin.PySpin.BMPOption property), 81
- thisown (PySpin.PySpin.BooleanNode property), 83
- thisown (PySpin.PySpin.Camera property), 140
- thisown (PySpin.PySpin.CameraBase property), 144
- thisown (PySpin.PySpin.CameraList property), 147
- thisown (PySpin.PySpin.CameraPtr property), 147
- thisown (PySpin.PySpin.CategoryNode property), 147
- thisown (PySpin.PySpin.CBasePtr property), 83
- thisown (PySpin.PySpin.CBooleanPtr property), 85
- thisown (PySpin.PySpin.CCategoryPtr property), 88
- thisown (PySpin.PySpin.CCMSettings property), 86
- thisown (PySpin.PySpin.CCommandPtr property), 91
- thisown (PySpin.PySpin.CDeviceInfoPtr property), 91
- thisown (PySpin.PySpin.CEnumEntryPtr property), 94
- thisown (PySpin.PySpin.CEnumerationPtr property), 97
- thisown (PySpin.PySpin.CFeatureBag property), 98
- thisown (PySpin.PySpin.CFloatPtr property), 98
- thisown (PySpin.PySpin.ChannelStatistics property), 148
- thisown (PySpin.PySpin.ChunkData property), 151

- thisown (PySpin.PySpin.CIntegerPtr property), 101
 thisown (PySpin.PySpin.CNodeMapDynPtr property), 103
 thisown (PySpin.PySpin.CNodeMapPtr property), 104
 thisown (PySpin.PySpin.CNodePtr property), 106
 thisown (PySpin.PySpin.CommandNode property), 153
 thisown (PySpin.PySpin.CRegisterPtr property), 109
 thisown (PySpin.PySpin.CSelectorPtr property), 110
 thisown (PySpin.PySpin.CSelectorSet property), 110
 thisown (PySpin.PySpin.CStringPtr property), 113
 thisown (PySpin.PySpin.CValuePtr property), 115
 thisown (PySpin.PySpin.DeviceArrivalEventHandler property), 153
 thisown (PySpin.PySpin.DeviceEventExposureEndData property), 153
 thisown (PySpin.PySpin.DeviceEventHandler property), 153
 thisown (PySpin.PySpin.DeviceEventInferenceData property), 154
 thisown (PySpin.PySpin.DeviceRemovalEventHandler property), 154
 thisown (PySpin.PySpin.double_autovector_t property), 404
 thisown (PySpin.PySpin.EAccessModeClass property), 155
 thisown (PySpin.PySpin.ECachingModeClass property), 155
 thisown (PySpin.PySpin.EDisplayNotationClass property), 155
 thisown (PySpin.PySpin.EEndianessClass property), 156
 thisown (PySpin.PySpin.ESchemaVersionClass property), 156
 thisown (PySpin.PySpin.EInputDirectionClass property), 157
 thisown (PySpin.PySpin.ENamespaceClass property), 157
 thisown (PySpin.PySpin.EnumEntryNode property), 161
 thisown (PySpin.PySpin.EnumNode property), 163
 thisown (PySpin.PySpin.ERepresentationClass property), 158
 thisown (PySpin.PySpin.ESignClass property), 158
 thisown (PySpin.PySpin.ESlopeClass property), 159
 thisown (PySpin.PySpin.EStandardNamespaceClass property), 159
 thisown (PySpin.PySpin.EventHandler property), 163
 thisown (PySpin.PySpin.EVisibilityClass property), 159
 thisown (PySpin.PySpin.EYesNoClass property), 160
 thisown (PySpin.PySpin.FloatNode property), 165
 thisown (PySpin.PySpin.FloatRegNode property), 166
 thisown (PySpin.PySpin.gcstring property), 408
 thisown (PySpin.PySpin.H264Option property), 168
 thisown (PySpin.PySpin.IBase property), 168
 thisown (PySpin.PySpin.IBoolean property), 168
 thisown (PySpin.PySpin.ICameraBase property), 170
 thisown (PySpin.PySpin.ICameraList property), 171
 thisown (PySpin.PySpin.ICategory property), 172
 thisown (PySpin.PySpin.IChunkData property), 173
 thisown (PySpin.PySpin.ICommand property), 174
 thisown (PySpin.PySpin.IDestroy property), 174
 thisown (PySpin.PySpin.IDeviceArrivalEventHandler property), 174
 thisown (PySpin.PySpin.IDeviceEventHandler property), 174
 thisown (PySpin.PySpin.IDeviceInfo property), 175
 thisown (PySpin.PySpin.IDeviceRemovalEventHandler property), 175
 thisown (PySpin.PySpin.IEnumEntry property), 175
 thisown (PySpin.PySpin.IEnumeration property), 176
 thisown (PySpin.PySpin.IEnumerationT_AcquisitionModeEnums property), 177
 thisown (PySpin.PySpin.IEnumerationT_AcquisitionStatusSelectorEnums property), 178
 thisown (PySpin.PySpin.IEnumerationT_ActionSelectorEnums property), 178
 thisown (PySpin.PySpin.IEnumerationT_ActionUnconditionalModeEnums property), 179
 thisown (PySpin.PySpin.IEnumerationT_AdcBitDepthEnums property), 180
 thisown (PySpin.PySpin.IEnumerationT_AutoAlgorithmSelectorEnums property), 180
 thisown (PySpin.PySpin.IEnumerationT_AutoExposureControlPriorityEnums property), 181
 thisown (PySpin.PySpin.IEnumerationT_AutoExposureLightingModeEnums property), 182
 thisown (PySpin.PySpin.IEnumerationT_AutoExposureMeteringModeEnums property), 182
 thisown (PySpin.PySpin.IEnumerationT_AutoExposureTargetGreyValueAuto property), 183
 thisown (PySpin.PySpin.IEnumerationT_BalanceRatioSelectorEnums property), 184
 thisown (PySpin.PySpin.IEnumerationT_BalanceWhiteAutoEnums property), 184
 thisown (PySpin.PySpin.IEnumerationT_BalanceWhiteAutoProfileEnums property), 185
 thisown (PySpin.PySpin.IEnumerationT_BinningHorizontalModeEnums property), 186
 thisown (PySpin.PySpin.IEnumerationT_BinningSelectorEnums property), 186
 thisown (PySpin.PySpin.IEnumerationT_BinningVerticalModeEnums property), 187
 thisown (PySpin.PySpin.IEnumerationT_BlackLevelAutoBalanceEnums property), 188
 thisown (PySpin.PySpin.IEnumerationT_BlackLevelAutoEnums property), 188
 thisown (PySpin.PySpin.IEnumerationT_BlackLevelSelectorEnums property), 189
 thisown (PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionAutoEnums property), 189

property), 190

thisown(PySpin.PySpin.IEnumerationT_BsiFlatFieldCorrectionModeEnums, property), 190

thisown(PySpin.PySpin.IEnumerationT_ChunkBlackLevelSelectorEnums, property), 191

thisown(PySpin.PySpin.IEnumerationT_ChunkCounterSelectorEnums, property), 192

thisown(PySpin.PySpin.IEnumerationT_ChunkEncoderSelectorEnums, property), 192

thisown(PySpin.PySpin.IEnumerationT_ChunkEncoderStatusEnums, property), 193

thisown(PySpin.PySpin.IEnumerationT_ChunkExposureTimeSelectorEnums, property), 194

thisown(PySpin.PySpin.IEnumerationT_ChunkGainSelectorEnums, property), 194

thisown(PySpin.PySpin.IEnumerationT_ChunkImageCompressionModeEnums, property), 195

thisown(PySpin.PySpin.IEnumerationT_ChunkPixelFormatEnums, property), 196

thisown(PySpin.PySpin.IEnumerationT_ChunkRegionIDEnums, property), 196

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateEnums, property), 197

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateEnums, property), 198

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateEnums, property), 198

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateEnums, property), 199

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dCoordinateEnums, property), 200

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dDistanceEnums, property), 201

thisown(PySpin.PySpin.IEnumerationT_ChunkScan3dOutputEnums, property), 201

thisown(PySpin.PySpin.IEnumerationT_ChunkSelectorEnums, property), 202

thisown(PySpin.PySpin.IEnumerationT_ChunkSourceIDEnums, property), 203

thisown(PySpin.PySpin.IEnumerationT_ChunkTimerSelectorEnums, property), 203

thisown(PySpin.PySpin.IEnumerationT_ChunkTransferStrategyEnums, property), 204

thisown(PySpin.PySpin.IEnumerationT_ClConfigurationEnums, property), 205

thisown(PySpin.PySpin.IEnumerationT_ClTimeSlotsCountEnums, property), 205

thisown(PySpin.PySpin.IEnumerationT_ColorTransformationEnums, property), 206

thisown(PySpin.PySpin.IEnumerationT_ColorTransformationEnums, property), 207

thisown(PySpin.PySpin.IEnumerationT_ComponentDestinationEnums, property), 207

thisown(PySpin.PySpin.IEnumerationT_ComponentSelectorEnums, property), 208

thisown(PySpin.PySpin.IEnumerationT_CompressionSaturationPriorityEnums, property), 209

thisown(PySpin.PySpin.IEnumerationT_CounterEventActivationEnums, property), 209

thisown(PySpin.PySpin.IEnumerationT_CounterEventSourceEnums, property), 210

thisown(PySpin.PySpin.IEnumerationT_CounterResetActivationEnums, property), 211

thisown(PySpin.PySpin.IEnumerationT_CounterResetSourceEnums, property), 211

thisown(PySpin.PySpin.IEnumerationT_CounterSelectorEnums, property), 212

thisown(PySpin.PySpin.IEnumerationT_CounterStatusEnums, property), 213

thisown(PySpin.PySpin.IEnumerationT_CounterTriggerActivationEnums, property), 213

thisown(PySpin.PySpin.IEnumerationT_CounterTriggerSourceEnums, property), 214

thisown(PySpin.PySpin.IEnumerationT_CxpConnectionTestModeEnums, property), 215

thisown(PySpin.PySpin.IEnumerationT_CxpLinkConfigurationEnums, property), 215

thisown(PySpin.PySpin.IEnumerationT_CxpLinkConfigurationPreferredEnums, property), 216

thisown(PySpin.PySpin.IEnumerationT_CxpLinkConfigurationStatusEnums, property), 217

thisown(PySpin.PySpin.IEnumerationT_CxpPoCxpStatusEnums, property), 217

thisown(PySpin.PySpin.IEnumerationT_DecimationHorizontalModeEnums, property), 218

thisown(PySpin.PySpin.IEnumerationT_DecimationSelectorEnums, property), 219

thisown(PySpin.PySpin.IEnumerationT_DecimationVerticalModeEnums, property), 219

thisown(PySpin.PySpin.IEnumerationT_DefectCorrectionModeEnums, property), 220

thisown(PySpin.PySpin.IEnumerationT_DeinterlacingEnums, property), 221

thisown(PySpin.PySpin.IEnumerationT_DeviceAccessStatusEnum, property), 221

thisown(PySpin.PySpin.IEnumerationT_DeviceCharacterSetEnums, property), 222

thisown(PySpin.PySpin.IEnumerationT_DeviceClockSelectorEnums, property), 223

thisown(PySpin.PySpin.IEnumerationT_DeviceConnectionStatusEnums, property), 223

thisown(PySpin.PySpin.IEnumerationT_DeviceCurrentSpeedEnum, property), 224

thisown(PySpin.PySpin.IEnumerationT_DeviceEndianessMechanismEnum, property), 225

thisown(PySpin.PySpin.IEnumerationT_DeviceIndicatorModeEnums, property), 225

thisown(PySpin.PySpin.IEnumerationT_DeviceLinkHeartbeatModeEnum, property), 225

property), 226

thisown(PySpin.PySpin.IEnumerationT_DeviceLinkThroughputEnum property), 227

thisown(PySpin.PySpin.IEnumerationT_DevicePowerSupplyEnum property), 227

thisown(PySpin.PySpin.IEnumerationT_DeviceRegistersEnum property), 228

thisown(PySpin.PySpin.IEnumerationT_DeviceScanTypeEnum property), 229

thisown(PySpin.PySpin.IEnumerationT_DeviceSensorCharacteristicsEnum property), 229

thisown(PySpin.PySpin.IEnumerationT_DeviceSerialPortEnum property), 230

thisown(PySpin.PySpin.IEnumerationT_DeviceSerialPortSpeedEnum property), 231

thisown(PySpin.PySpin.IEnumerationT_DeviceStreamChannelEnum property), 231

thisown(PySpin.PySpin.IEnumerationT_DeviceStreamChannelTypeEnum property), 232

thisown(PySpin.PySpin.IEnumerationT_DeviceTapGeometryEnum property), 233

thisown(PySpin.PySpin.IEnumerationT_DeviceTemperatureSensorEnum property), 234

thisown(PySpin.PySpin.IEnumerationT_DeviceTLTypeEnum property), 233

thisown(PySpin.PySpin.IEnumerationT_DeviceTypeEnum property), 235

thisown(PySpin.PySpin.IEnumerationT_DeviceTypeEnum property), 235

thisown(PySpin.PySpin.IEnumerationT_EncoderModeEnum property), 236

thisown(PySpin.PySpin.IEnumerationT_EncoderOutputModeEnum property), 237

thisown(PySpin.PySpin.IEnumerationT_EncoderResetActiveEnum property), 237

thisown(PySpin.PySpin.IEnumerationT_EncoderResetSourceEnum property), 238

thisown(PySpin.PySpin.IEnumerationT_EncoderSelectorEnum property), 239

thisown(PySpin.PySpin.IEnumerationT_EncoderSourceAEnum property), 239

thisown(PySpin.PySpin.IEnumerationT_EncoderSourceBEnum property), 240

thisown(PySpin.PySpin.IEnumerationT_EncoderStatusEnum property), 241

thisown(PySpin.PySpin.IEnumerationT_EventNotificationEnum property), 241

thisown(PySpin.PySpin.IEnumerationT_EventSelectorEnum property), 242

thisown(PySpin.PySpin.IEnumerationT_ExposureActiveModeEnum property), 243

thisown(PySpin.PySpin.IEnumerationT_ExposureAutoEnum property), 243

thisown(PySpin.PySpin.IEnumerationT_ExposureModeEnum property), 244

thisown(PySpin.PySpin.IEnumerationT_ExposureTimeModeEnums property), 245

thisown(PySpin.PySpin.IEnumerationT_ExposureTimeSelectorEnums property), 245

thisown(PySpin.PySpin.IEnumerationT_ExternalVoltageSelectorEnums property), 246

thisown(PySpin.PySpin.IEnumerationT_FfcModeEnums property), 247

thisown(PySpin.PySpin.IEnumerationT_FileOpenModeEnums property), 248

thisown(PySpin.PySpin.IEnumerationT_FileOperationSelectorEnums property), 249

thisown(PySpin.PySpin.IEnumerationT_FileOperationStatusEnums property), 249

thisown(PySpin.PySpin.IEnumerationT_FileSelectorEnums property), 250

thisown(PySpin.PySpin.IEnumerationT_FLIRFilterDriverStatusEnum property), 247

thisown(PySpin.PySpin.IEnumerationT_GainAutoBalanceEnums property), 251

thisown(PySpin.PySpin.IEnumerationT_GainAutoEnums property), 252

thisown(PySpin.PySpin.IEnumerationT_GainConversionEnums property), 253

thisown(PySpin.PySpin.IEnumerationT_GainSelectorEnums property), 253

thisown(PySpin.PySpin.IEnumerationT_GenICamXMLLocationEnum property), 254

thisown(PySpin.PySpin.IEnumerationT_GevCCPEnum property), 255

thisown(PySpin.PySpin.IEnumerationT_GevCCPEnums property), 255

thisown(PySpin.PySpin.IEnumerationT_GevCurrentPhysicalLinkConfigurationEnum property), 256

thisown(PySpin.PySpin.IEnumerationT_GevGVCPEExtendedStatusCodesEnum property), 257

thisown(PySpin.PySpin.IEnumerationT_GevGVSPEExtendedIDModeEnum property), 257

thisown(PySpin.PySpin.IEnumerationT_GevIEEE1588ClockAccuracyEnum property), 258

thisown(PySpin.PySpin.IEnumerationT_GevIEEE1588ModeEnums property), 259

thisown(PySpin.PySpin.IEnumerationT_GevIEEE1588StatusEnums property), 259

thisown(PySpin.PySpin.IEnumerationT_GevIEEE1588StatusLatchedEnum property), 260

thisown(PySpin.PySpin.IEnumerationT_GevIPConfigurationStatusEnums property), 261

thisown(PySpin.PySpin.IEnumerationT_GevPhysicalLinkConfigurationEnum property), 261

thisown(PySpin.PySpin.IEnumerationT_GevSCPDirectionEnums property), 262

thisown(PySpin.PySpin.IEnumerationT_GevSupportedOptionSelectorEnum property), 262

property), 263

thisown(PySpin.PySpin.IEnumerationT_GUIXMLLocation property), 251

thisown(PySpin.PySpin.IEnumerationT_ImageComponent property), 263

thisown(PySpin.PySpin.IEnumerationT_ImageCompression property), 264

thisown(PySpin.PySpin.IEnumerationT_ImageCompression property), 265

thisown(PySpin.PySpin.IEnumerationT_ImageCompression property), 265

thisown(PySpin.PySpin.IEnumerationT_InterfaceTypeEnum property), 266

thisown(PySpin.PySpin.IEnumerationT_LensShadingCoefficient property), 267

thisown(PySpin.PySpin.IEnumerationT_LensShadingCorrection property), 268

thisown(PySpin.PySpin.IEnumerationT_LineFormatEnum property), 269

thisown(PySpin.PySpin.IEnumerationT_LineInputFilterSelector property), 269

thisown(PySpin.PySpin.IEnumerationT_LineModeEnums property), 270

thisown(PySpin.PySpin.IEnumerationT_LineSelectorEnum property), 271

thisown(PySpin.PySpin.IEnumerationT_LineSourceEnum property), 271

thisown(PySpin.PySpin.IEnumerationT_LogicBlockLUTInput property), 272

thisown(PySpin.PySpin.IEnumerationT_LogicBlockLUTInput property), 273

thisown(PySpin.PySpin.IEnumerationT_LogicBlockLUTInput property), 273

thisown(PySpin.PySpin.IEnumerationT_LogicBlockLUTSelector property), 274

thisown(PySpin.PySpin.IEnumerationT_LogicBlockSelector property), 275

thisown(PySpin.PySpin.IEnumerationT_LUTSelectorEnum property), 267

thisown(PySpin.PySpin.IEnumerationT_MultiRoiConfiguration property), 275

thisown(PySpin.PySpin.IEnumerationT_MultiRoiSelectorEnum property), 276

thisown(PySpin.PySpin.IEnumerationT_PixelColorFilterEnum property), 277

thisown(PySpin.PySpin.IEnumerationT_PixelFormatEnum property), 278

thisown(PySpin.PySpin.IEnumerationT_PixelFormatInfoSelector property), 279

thisown(PySpin.PySpin.IEnumerationT_PixelSizeEnums property), 279

thisown(PySpin.PySpin.IEnumerationT_POEStatusEnum property), 277

thisown(PySpin.PySpin.IEnumerationT_RegionDestination property), 280

thisown(PySpin.PySpin.IEnumerationT_RegionModeEnums property), 281

thisown(PySpin.PySpin.IEnumerationT_RegionSelectorEnums property), 281

thisown(PySpin.PySpin.IEnumerationT_RgbTransformLightSourceEnums property), 282

thisown(PySpin.PySpin.IEnumerationT_Scan3dCoordinateReferenceSelector property), 283

thisown(PySpin.PySpin.IEnumerationT_Scan3dCoordinateSelectorEnums property), 283

thisown(PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemEnums property), 284

thisown(PySpin.PySpin.IEnumerationT_Scan3dCoordinateSystemReferenceSelector property), 285

thisown(PySpin.PySpin.IEnumerationT_Scan3dCoordinateTransformSelector property), 285

thisown(PySpin.PySpin.IEnumerationT_Scan3dDistanceUnitEnums property), 286

thisown(PySpin.PySpin.IEnumerationT_Scan3dOutputModeEnums property), 287

thisown(PySpin.PySpin.IEnumerationT_SensorDigitizationTapsEnums property), 287

thisown(PySpin.PySpin.IEnumerationT_SensorShutterModeEnums property), 288

thisown(PySpin.PySpin.IEnumerationT_SensorTapsEnums property), 289

thisown(PySpin.PySpin.IEnumerationT_SequencerConfigurationModeEnum property), 289

thisown(PySpin.PySpin.IEnumerationT_SequencerConfigurationValidEnum property), 290

thisown(PySpin.PySpin.IEnumerationT_SequencerModeEnums property), 291

thisown(PySpin.PySpin.IEnumerationT_SequencerSetValidEnums property), 291

thisown(PySpin.PySpin.IEnumerationT_SequencerTriggerActivationEnum property), 292

thisown(PySpin.PySpin.IEnumerationT_SequencerTriggerSourceEnums property), 293

thisown(PySpin.PySpin.IEnumerationT_SerialPortBaudRateEnums property), 293

thisown(PySpin.PySpin.IEnumerationT_SerialPortParityEnums property), 294

thisown(PySpin.PySpin.IEnumerationT_SerialPortSelectorEnums property), 295

thisown(PySpin.PySpin.IEnumerationT_SerialPortSourceEnums property), 295

thisown(PySpin.PySpin.IEnumerationT_SerialPortStopBitsEnums property), 296

thisown(PySpin.PySpin.IEnumerationT_SoftwareSignalSelectorEnums property), 297

thisown(PySpin.PySpin.IEnumerationT_SourceSelectorEnums property), 297

thisown(PySpin.PySpin.IEnumerationT_StereoResolutionEnums property), 297

property), 298

thisown (PySpin.PySpin.IEnumerationT_StreamBufferCountEnums property), 299

thisown (PySpin.PySpin.IEnumerationT_StreamBufferHandleEnums property), 299

thisown (PySpin.PySpin.IEnumerationT_StreamModeEnums property), 300

thisown (PySpin.PySpin.IEnumerationT_StreamTypeEnum property), 301

thisown (PySpin.PySpin.IEnumerationT_TeledyneGigeVisionEnums property), 302

thisown (PySpin.PySpin.IEnumerationT_TestPatternEnums property), 303

thisown (PySpin.PySpin.IEnumerationT_TestPatternGenerationEnums property), 303

thisown (PySpin.PySpin.IEnumerationT_TimerSelectorEnums property), 304

thisown (PySpin.PySpin.IEnumerationT_TimerStatusEnums property), 305

thisown (PySpin.PySpin.IEnumerationT_TimerTriggerActivationEnums property), 305

thisown (PySpin.PySpin.IEnumerationT_TimerTriggerSourceEnums property), 306

thisown (PySpin.PySpin.IEnumerationT_TLTypeEnum property), 301

thisown (PySpin.PySpin.IEnumerationT_TransferComponentEnums property), 307

thisown (PySpin.PySpin.IEnumerationT_TransferControlModeEnums property), 307

thisown (PySpin.PySpin.IEnumerationT_TransferOperationModeEnums property), 308

thisown (PySpin.PySpin.IEnumerationT_TransferQueueModeEnums property), 309

thisown (PySpin.PySpin.IEnumerationT_TransferSelectorEnums property), 309

thisown (PySpin.PySpin.IEnumerationT_TransferStatusSelectorEnums property), 310

thisown (PySpin.PySpin.IEnumerationT_TransferTriggerActivationEnums property), 311

thisown (PySpin.PySpin.IEnumerationT_TransferTriggerModeEnums property), 311

thisown (PySpin.PySpin.IEnumerationT_TransferTriggerSetEnums property), 312

thisown (PySpin.PySpin.IEnumerationT_TransferTriggerSourceEnums property), 313

thisown (PySpin.PySpin.IEnumerationT_TriggerActivationEnums property), 313

thisown (PySpin.PySpin.IEnumerationT_TriggerModeEnums property), 314

thisown (PySpin.PySpin.IEnumerationT_TriggerOverlapEnums property), 315

thisown (PySpin.PySpin.IEnumerationT_TriggerSelectorEnums property), 315

thisown (PySpin.PySpin.IEnumerationT_TriggerSourceEnums property), 316

thisown (PySpin.PySpin.IEnumerationT_U3VCurrentSpeedEnums property), 317

thisown (PySpin.PySpin.IEnumerationT_UserOutputSelectorEnums property), 317

thisown (PySpin.PySpin.IEnumerationT_UserSetDefaultEnums property), 318

thisown (PySpin.PySpin.IEnumerationT_UserSetSelectorEnums property), 319

thisown (PySpin.PySpin.IEnumerationT_WhiteClipSelectorEnums property), 319

thisown (PySpin.PySpin.IEnumReference property), 176

thisown (PySpin.PySpin.IFloat property), 320

thisown (PySpin.PySpin.IImage property), 324

thisown (PySpin.PySpin.IImageEventHandler property), 324

thisown (PySpin.PySpin.IImageList property), 325

thisown (PySpin.PySpin.IImageListEventHandler property), 325

thisown (PySpin.PySpin.IImageProcessor property), 327

thisown (PySpin.PySpin.IInteger property), 328

thisown (PySpin.PySpin.IInterface property), 328

thisown (PySpin.PySpin.IInterfaceArrivalEventHandler property), 329

thisown (PySpin.PySpin.IInterfaceEventHandler property), 329

thisown (PySpin.PySpin.IInterfaceList property), 329

thisown (PySpin.PySpin.IInterfaceRemovalEventHandler property), 330

thisown (PySpin.PySpin.ILoggingEventHandler property), 330

thisown (PySpin.PySpin.Image property), 348

thisown (PySpin.PySpin.ImageEventHandler property), 348

thisown (PySpin.PySpin.ImageList property), 350

thisown (PySpin.PySpin.ImageListEventHandler property), 350

thisown (PySpin.PySpin.ImagePixel property), 350

thisown (PySpin.PySpin.ImageProcessor property), 351

thisown (PySpin.PySpin.ImagePtr property), 352

thisown (PySpin.PySpin.ImageUtility property), 353

thisown (PySpin.PySpin.ImageUtilityCCM property), 354

thisown (PySpin.PySpin.ImageUtilityHeatmap property), 355

thisown (PySpin.PySpin.ImageUtilityPolarization property), 358

thisown (PySpin.PySpin.ImageUtilityStereo property), 360

thisown (PySpin.PySpin.InferenceBoundingBox property), 360

thisown (PySpin.PySpin.InferenceBoundingBoxResult property), 360

- `thisown` (`PySpin.PySpin.InferenceBoxCircle` property), 361
- `thisown` (`PySpin.PySpin.InferenceBoxRect` property), 361
- `thisown` (`PySpin.PySpin.InferenceBoxRotatedRect` property), 361
- `thisown` (`PySpin.PySpin.INode` property), 332
- `thisown` (`PySpin.PySpin.INodeMap` property), 332
- `thisown` (`PySpin.PySpin.INodeMapDyn` property), 334
- `thisown` (`PySpin.PySpin.int64_autovector_t` property), 408
- `thisown` (`PySpin.PySpin.IntegerNode` property), 364
- `thisown` (`PySpin.PySpin.InterfaceArrivalEventHandler` property), 364
- `thisown` (`PySpin.PySpin.InterfaceEventHandler` property), 364
- `thisown` (`PySpin.PySpin.InterfaceList` property), 365
- `thisown` (`PySpin.PySpin.InterfacePtr` property), 365
- `thisown` (`PySpin.PySpin.InterfaceRemovalEventHandler` property), 365
- `thisown` (`PySpin.PySpin.IntRegNode` property), 361
- `thisown` (`PySpin.PySpin.IPersistScript` property), 334
- `thisown` (`PySpin.PySpin.IPointCloud` property), 335
- `thisown` (`PySpin.PySpin.IReference` property), 335
- `thisown` (`PySpin.PySpin.IRegister` property), 336
- `thisown` (`PySpin.PySpin.ISelector` property), 336
- `thisown` (`PySpin.PySpin.ISelectorDigit` property), 337
- `thisown` (`PySpin.PySpin.IString` property), 337
- `thisown` (`PySpin.PySpin.ISystem` property), 339
- `thisown` (`PySpin.PySpin.ISystemEventHandler` property), 339
- `thisown` (`PySpin.PySpin.IValue` property), 339
- `thisown` (`PySpin.PySpin.JPEGOption` property), 370
- `thisown` (`PySpin.PySpin.JPG2Option` property), 371
- `thisown` (`PySpin.PySpin.LibraryVersion` property), 371
- `thisown` (`PySpin.PySpin.LoggingEventData` property), 372
- `thisown` (`PySpin.PySpin.LoggingEventDataPtr` property), 372
- `thisown` (`PySpin.PySpin.LoggingEventHandler` property), 372
- `thisown` (`PySpin.PySpin.MJPGOption` property), 372
- `thisown` (`PySpin.PySpin.Node` property), 377
- `thisown` (`PySpin.PySpin.node_vector` property), 409
- `thisown` (`PySpin.PySpin.NodeCallback` property), 378
- `thisown` (`PySpin.PySpin.NodeMap` property), 382
- `thisown` (`PySpin.PySpin.PGMOption` property), 383
- `thisown` (`PySpin.PySpin.PNGOption` property), 383
- `thisown` (`PySpin.PySpin.PointCloud` property), 384
- `thisown` (`PySpin.PySpin.PointCloudParameters` property), 384
- `thisown` (`PySpin.PySpin.PPMOption` property), 383
- `thisown` (`PySpin.PySpin.RegisterNode` property), 386
- `thisown` (`PySpin.PySpin.SIOption` property), 386
- `thisown` (`PySpin.PySpin.SpinVideo` property), 388
- `thisown` (`PySpin.PySpin.Stereo3DPoint` property), 389
- `thisown` (`PySpin.PySpin.StereoCameraParameters` property), 389
- `thisown` (`PySpin.PySpin.StringNode` property), 390
- `thisown` (`PySpin.PySpin.StringRegNode` property), 391
- `thisown` (`PySpin.PySpin.System` property), 395
- `thisown` (`PySpin.PySpin.SystemEventHandler` property), 395
- `thisown` (`PySpin.PySpin.SystemPtr` property), 395
- `thisown` (`PySpin.PySpin.TIFFOption` property), 395
- `thisown` (`PySpin.PySpin.TransportLayerDevice` property), 397
- `thisown` (`PySpin.PySpin.TransportLayerInterface` property), 399
- `thisown` (`PySpin.PySpin.TransportLayerStream` property), 401
- `thisown` (`PySpin.PySpin.TransportLayerSystem` property), 402
- `thisown` (`PySpin.PySpin.value_vector` property), 411
- `thisown` (`PySpin.PySpin.ValueNode` property), 404
- `thisown` (`PySpin.PySpin.Version_t` property), 404
- `thisown` (`PySpin.SpinVideo` property), 70
- `thisown` (`PySpin.System` property), 74
- `thisown` (`PySpin.SystemEventHandler` property), 8
- `thisown` (`PySpin.SystemPtr` property), 74
- `thisown` (`PySpin.TransportLayerDevice` property), 76
- `thisown` (`PySpin.TransportLayerInterface` property), 78
- `thisown` (`PySpin.TransportLayerStream` property), 80
- `ThrowBadAlloc()` (in module `PySpin.PySpin`), 395
- `TIFFOption` (class in `PySpin.PySpin`), 395
- `TimerDelay` (`PySpin.Camera` property), 32
- `TimerDelay` (`PySpin.PySpin.Camera` property), 137
- `TimerDuration` (`PySpin.Camera` property), 32
- `TimerDuration` (`PySpin.PySpin.Camera` property), 137
- `TimerReset` (`PySpin.Camera` property), 32
- `TimerReset` (`PySpin.PySpin.Camera` property), 138
- `TimerSelector` (`PySpin.Camera` property), 32
- `TimerSelector` (`PySpin.PySpin.Camera` property), 138
- `TimerStatus` (`PySpin.Camera` property), 32
- `TimerStatus` (`PySpin.PySpin.Camera` property), 138
- `TimerTriggerActivation` (`PySpin.Camera` property), 32
- `TimerTriggerActivation` (`PySpin.PySpin.Camera` property), 138
- `TimerTriggerSource` (`PySpin.Camera` property), 32
- `TimerTriggerSource` (`PySpin.PySpin.Camera` property), 138
- `TimerValue` (`PySpin.Camera` property), 32
- `TimerValue` (`PySpin.PySpin.Camera` property), 138
- `Timestamp` (`PySpin.Camera` property), 32
- `Timestamp` (`PySpin.PySpin.Camera` property), 138
- `TimestampIncrement` (`PySpin.Camera` property), 32

- TimestampIncrement (*PySpin.PySpin.Camera property*), 138
- TimestampLatch (*PySpin.Camera property*), 32
- TimestampLatch (*PySpin.PySpin.Camera property*), 138
- TimestampLatchValue (*PySpin.Camera property*), 32
- TimestampLatchValue (*PySpin.PySpin.Camera property*), 138
- TimestampReset (*PySpin.Camera property*), 32
- TimestampReset (*PySpin.PySpin.Camera property*), 138
- TLDevice (*PySpin.PySpin.ICameraBase property*), 170
- TLDisplayName (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLFileName (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLID (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLInterface (*PySpin.IInterface property*), 67
- TLInterface (*PySpin.PySpin.IInterface property*), 328
- TLModelName (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLParamsLocked (*PySpin.Camera property*), 32
- TLParamsLocked (*PySpin.PySpin.Camera property*), 137
- TLPath (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLStream (*PySpin.PySpin.ICameraBase property*), 170
- TLSystem (*PySpin.PySpin.ISystem property*), 338
- TLType (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLVendorName (*PySpin.PySpin.TransportLayerSystem property*), 402
- TLVersion (*PySpin.PySpin.TransportLayerSystem property*), 402
- Tokenize() (*in module PySpin.PySpin*), 395
- topLeftXCoord (*PySpin.PySpin.InferenceBoxRect property*), 361
- topLeftXCoord (*PySpin.PySpin.InferenceBoxRotatedRect property*), 361
- topLeftYCoord (*PySpin.PySpin.InferenceBoxRect property*), 361
- topLeftYCoord (*PySpin.PySpin.InferenceBoxRotatedRect property*), 361
- ToString() (*PySpin.PySpin.CBooleanPtr method*), 85
- ToString() (*PySpin.PySpin.CCategoryPtr method*), 88
- ToString() (*PySpin.PySpin.CCommandPtr method*), 90
- ToString() (*PySpin.PySpin.CEnumEntryPtr method*), 94
- ToString() (*PySpin.PySpin.CEnumerationPtr method*), 97
- ToString() (*PySpin.PySpin.CIntegerPtr method*), 101
- ToString() (*PySpin.PySpin.CRegisterPtr method*), 109
- ToString() (*PySpin.PySpin.CSelectorSet method*), 110
- ToString() (*PySpin.PySpin.CStringPtr method*), 113
- ToString() (*PySpin.PySpin.CValuePtr method*), 115
- ToString() (*PySpin.PySpin.EAccessModeClass static method*), 154
- ToString() (*PySpin.PySpin.ECachingModeClass static method*), 155
- ToString() (*PySpin.PySpin.EDisplayNotationClass static method*), 155
- ToString() (*PySpin.PySpin.EEndianessClass static method*), 156
- ToString() (*PySpin.PySpin.EGenApiSchemaVersionClass static method*), 156
- ToString() (*PySpin.PySpin.EInputDirectionClass static method*), 157
- ToString() (*PySpin.PySpin.ENamespaceClass static method*), 157
- ToString() (*PySpin.PySpin.ERepresentationClass static method*), 157
- ToString() (*PySpin.PySpin.ESignClass static method*), 158
- ToString() (*PySpin.PySpin.ESlopeClass static method*), 158
- ToString() (*PySpin.PySpin.EStandardNameSpaceClass static method*), 159
- ToString() (*PySpin.PySpin.EVisibilityClass static method*), 159
- ToString() (*PySpin.PySpin.EYesNoClass static method*), 160
- ToString() (*PySpin.PySpin.ISelectorDigit method*), 337
- ToString() (*PySpin.PySpin.IValue method*), 339
- ToString() (*PySpin.PySpin.ValueNode method*), 403
- TotalDisparity (*PySpin.Camera property*), 32
- TotalDisparity (*PySpin.PySpin.Camera property*), 138
- TransferAbort (*PySpin.Camera property*), 32
- TransferAbort (*PySpin.PySpin.Camera property*), 138
- TransferBlockCount (*PySpin.Camera property*), 33
- TransferBlockCount (*PySpin.PySpin.Camera property*), 138
- TransferBurstCount (*PySpin.Camera property*), 33
- TransferBurstCount (*PySpin.PySpin.Camera property*), 138
- TransferComponentSelector (*PySpin.Camera property*), 33
- TransferComponentSelector (*PySpin.PySpin.Camera property*), 138
- TransferControlMode (*PySpin.Camera property*), 33
- TransferControlMode (*PySpin.PySpin.Camera property*), 138
- TransferOperationMode (*PySpin.Camera property*), 33
- TransferOperationMode (*PySpin.PySpin.Camera property*), 138
- TransferPause (*PySpin.Camera property*), 33

- TransferPause (*PySpin.PySpin.Camera* property), 138
- TransferQueueCurrentBlockCount (*PySpin.Camera* property), 33
- TransferQueueCurrentBlockCount (*PySpin.PySpin.Camera* property), 138
- TransferQueueMaxBlockCount (*PySpin.Camera* property), 33
- TransferQueueMaxBlockCount (*PySpin.PySpin.Camera* property), 138
- TransferQueueMode (*PySpin.Camera* property), 33
- TransferQueueMode (*PySpin.PySpin.Camera* property), 138
- TransferQueueOverflowCount (*PySpin.Camera* property), 33
- TransferQueueOverflowCount (*PySpin.PySpin.Camera* property), 138
- TransferResume (*PySpin.Camera* property), 33
- TransferResume (*PySpin.PySpin.Camera* property), 138
- TransferSelector (*PySpin.Camera* property), 33
- TransferSelector (*PySpin.PySpin.Camera* property), 138
- TransferStart (*PySpin.Camera* property), 33
- TransferStart (*PySpin.PySpin.Camera* property), 138
- TransferStatus (*PySpin.Camera* property), 33
- TransferStatus (*PySpin.PySpin.Camera* property), 138
- TransferStatusSelector (*PySpin.Camera* property), 33
- TransferStatusSelector (*PySpin.PySpin.Camera* property), 138
- TransferStop (*PySpin.Camera* property), 33
- TransferStop (*PySpin.PySpin.Camera* property), 138
- TransferStreamChannel (*PySpin.Camera* property), 33
- TransferStreamChannel (*PySpin.PySpin.Camera* property), 138
- TransferTriggerActivation (*PySpin.Camera* property), 33
- TransferTriggerActivation (*PySpin.PySpin.Camera* property), 138
- TransferTriggerMode (*PySpin.Camera* property), 33
- TransferTriggerMode (*PySpin.PySpin.Camera* property), 139
- TransferTriggerSelector (*PySpin.Camera* property), 33
- TransferTriggerSelector (*PySpin.PySpin.Camera* property), 139
- TransferTriggerSource (*PySpin.Camera* property), 33
- TransferTriggerSource (*PySpin.PySpin.Camera* property), 139
- TransmissionDelay (*PySpin.Camera* property), 33
- TransmissionDelay (*PySpin.PySpin.Camera* property), 139
- TransmissionDelayAverage (*PySpin.Camera* property), 33
- TransmissionDelayAverage (*PySpin.PySpin.Camera* property), 139
- TransmissionDelayMax (*PySpin.Camera* property), 33
- TransmissionDelayMax (*PySpin.PySpin.Camera* property), 139
- TransportLayerDevice (class in *PySpin*), 75
- TransportLayerDevice (class in *PySpin.PySpin*), 396
- TransportLayerInterface (class in *PySpin*), 77
- TransportLayerInterface (class in *PySpin.PySpin*), 397
- TransportLayerStream (class in *PySpin*), 79
- TransportLayerStream (class in *PySpin.PySpin*), 399
- TransportLayerSystem (class in *PySpin.PySpin*), 401
- TriggerActivation (*PySpin.Camera* property), 33
- TriggerActivation (*PySpin.PySpin.Camera* property), 139
- TriggerDelay (*PySpin.Camera* property), 33
- TriggerDelay (*PySpin.PySpin.Camera* property), 139
- TriggerDivider (*PySpin.Camera* property), 33
- TriggerDivider (*PySpin.PySpin.Camera* property), 139
- TriggerEventTest (*PySpin.Camera* property), 33
- TriggerEventTest (*PySpin.PySpin.Camera* property), 139
- TriggerMode (*PySpin.Camera* property), 33
- TriggerMode (*PySpin.PySpin.Camera* property), 139
- TriggerMultiplier (*PySpin.Camera* property), 33
- TriggerMultiplier (*PySpin.PySpin.Camera* property), 139
- TriggerOverlap (*PySpin.Camera* property), 33
- TriggerOverlap (*PySpin.PySpin.Camera* property), 139
- TriggerSelector (*PySpin.Camera* property), 34
- TriggerSelector (*PySpin.PySpin.Camera* property), 139
- TriggerSoftware (*PySpin.Camera* property), 34
- TriggerSoftware (*PySpin.PySpin.Camera* property), 139
- TriggerSource (*PySpin.Camera* property), 34
- TriggerSource (*PySpin.PySpin.Camera* property), 139
- Type (*PySpin.PySpin.CCMSettings* property), 86
- type (*PySpin.PySpin.LibraryVersion* property), 371
- TypeToString() (*PySpin.ImageUtilityCCM* static method), 60
- TypeToString() (*PySpin.PySpin.ImageUtilityCCM* static method), 354
- ## U
- u (*PySpin.PySpin.ImagePixel* property), 350
- U3VAccessPrivilege (*PySpin.Camera* property), 34

- U3VAccessPrivilege (*PySpin.PySpin.Camera* property), 139
- U3VCPCapability (*PySpin.Camera* property), 34
- U3VCPCapability (*PySpin.PySpin.Camera* property), 139
- U3VCPEIRMAvailable (*PySpin.Camera* property), 34
- U3VCPEIRMAvailable (*PySpin.PySpin.Camera* property), 139
- U3VCPIIDC2Available (*PySpin.Camera* property), 34
- U3VCPIIDC2Available (*PySpin.PySpin.Camera* property), 139
- U3VCPSIRMAvailable (*PySpin.Camera* property), 34
- U3VCPSIRMAvailable (*PySpin.PySpin.Camera* property), 139
- U3VCurrentSpeed (*PySpin.Camera* property), 34
- U3VCurrentSpeed (*PySpin.PySpin.Camera* property), 139
- U3VMaxAcknowledgeTransferLength (*PySpin.Camera* property), 34
- U3VMaxAcknowledgeTransferLength (*PySpin.PySpin.Camera* property), 139
- U3VMaxCommandTransferLength (*PySpin.Camera* property), 34
- U3VMaxCommandTransferLength (*PySpin.PySpin.Camera* property), 139
- U3VMaxDeviceResponseTime (*PySpin.Camera* property), 34
- U3VMaxDeviceResponseTime (*PySpin.PySpin.Camera* property), 139
- U3VMessageChannelID (*PySpin.Camera* property), 34
- U3VMessageChannelID (*PySpin.PySpin.Camera* property), 139
- U3VNumberOfStreamChannels (*PySpin.Camera* property), 34
- U3VNumberOfStreamChannels (*PySpin.PySpin.Camera* property), 139
- U3VVersionMajor (*PySpin.Camera* property), 34
- U3VVersionMajor (*PySpin.PySpin.Camera* property), 139
- U3VVersionMinor (*PySpin.Camera* property), 34
- U3VVersionMinor (*PySpin.PySpin.Camera* property), 139
- UniquenessRatio (*PySpin.Camera* property), 34
- UniquenessRatio (*PySpin.PySpin.Camera* property), 139
- UnregisterAllLoggingEventHandlers() (*PySpin.PySpin.ISystem* method), 338
- UnregisterAllLoggingEventHandlers() (*PySpin.PySpin.System* method), 394
- UnregisterAllLoggingEventHandlers() (*PySpin.System* method), 73
- UnregisterEventHandler() (*PySpin.CameraBase* method), 39
- UnregisterEventHandler() (*PySpin.IInterface* method), 67
- UnregisterEventHandler() (*PySpin.PySpin.CameraBase* method), 144
- UnregisterEventHandler() (*PySpin.PySpin.ICameraBase* method), 170
- UnregisterEventHandler() (*PySpin.PySpin.IInterface* method), 328
- UnregisterEventHandler() (*PySpin.PySpin.ISystem* method), 338
- UnregisterEventHandler() (*PySpin.PySpin.System* method), 394
- UnregisterEventHandler() (*PySpin.System* method), 74
- UnregisterLoggingEventHandler() (*PySpin.PySpin.ISystem* method), 338
- UnregisterLoggingEventHandler() (*PySpin.PySpin.System* method), 394
- UnregisterLoggingEventHandler() (*PySpin.System* method), 74
- UpdateCameras() (*PySpin.IInterface* method), 67
- UpdateCameras() (*PySpin.PySpin.IInterface* method), 328
- UpdateCameras() (*PySpin.PySpin.ISystem* method), 338
- UpdateCameras() (*PySpin.PySpin.System* method), 394
- UpdateCameras() (*PySpin.System* method), 74
- UpdateFirmware() (in module *PySpin.PySpin*), 402
- UpdateFirmwareConsole() (in module *PySpin.PySpin*), 402
- UpdateFirmwareGUI() (in module *PySpin.PySpin*), 402
- UpdateInterfaceList() (*PySpin.PySpin.ISystem* method), 338
- UpdateInterfaceList() (*PySpin.PySpin.System* method), 395
- UpdateInterfaceList() (*PySpin.System* method), 74
- UrlDecode() (in module *PySpin.PySpin*), 402
- UrlEncode() (in module *PySpin.PySpin*), 403
- useMP4 (*PySpin.PySpin.H264Option* property), 168
- UserOutputSelector (*PySpin.Camera* property), 34
- UserOutputSelector (*PySpin.PySpin.Camera* property), 139
- UserOutputValue (*PySpin.Camera* property), 34
- UserOutputValue (*PySpin.PySpin.Camera* property), 140
- UserOutputValueAll (*PySpin.Camera* property), 34
- UserOutputValueAll (*PySpin.PySpin.Camera* property), 140
- UserOutputValueAllMask (*PySpin.Camera* property), 34
- UserOutputValueAllMask (*PySpin.PySpin.Camera* property), 140
- UserSetDefault (*PySpin.Camera* property), 34
- UserSetDefault (*PySpin.PySpin.Camera* property), 140

UserSetFeatureEnable (*PySpin.Camera* property), 34
UserSetFeatureEnable (*PySpin.PySpin.Camera* property), 140
UserSetLoad (*PySpin.Camera* property), 34
UserSetLoad (*PySpin.PySpin.Camera* property), 140
UserSetSave (*PySpin.Camera* property), 34
UserSetSave (*PySpin.PySpin.Camera* property), 140
UserSetSelector (*PySpin.Camera* property), 34
UserSetSelector (*PySpin.PySpin.Camera* property), 140

V

v (*PySpin.PySpin.ImagePixel* property), 350
V3_3Enable (*PySpin.Camera* property), 34
V3_3Enable (*PySpin.PySpin.Camera* property), 140
value_vector (class in *PySpin.PySpin*), 410
ValueNode (class in *PySpin.PySpin*), 403
Version_t (class in *PySpin.PySpin*), 404

W

WhiteClip (*PySpin.Camera* property), 34
WhiteClip (*PySpin.PySpin.Camera* property), 140
WhiteClipSelector (*PySpin.Camera* property), 34
WhiteClipSelector (*PySpin.PySpin.Camera* property), 140
Width (*PySpin.Camera* property), 34
width (*PySpin.PySpin.AVIOption* property), 81
Width (*PySpin.PySpin.Camera* property), 140
width (*PySpin.PySpin.H264Option* property), 168
width (*PySpin.PySpin.MJPGOption* property), 373
WidthMax (*PySpin.Camera* property), 34
WidthMax (*PySpin.PySpin.Camera* property), 140
WindowSizeH (*PySpin.Camera* property), 35
WindowSizeH (*PySpin.PySpin.Camera* property), 140
WindowSizeW (*PySpin.Camera* property), 35
WindowSizeW (*PySpin.PySpin.Camera* property), 140
WritePort() (*PySpin.PySpin.ICameraBase* method), 170

X

x (*PySpin.PySpin.Stereo3DPoint* property), 389

Y

y (*PySpin.PySpin.Stereo3DPoint* property), 389

Z

z (*PySpin.PySpin.Stereo3DPoint* property), 389