PySpin API Reference

Release 1.11

FLIR Integrated Imaging Solutions, Inc

Aug 01, 2019

CONTENTS:

ONE

INTRODUCTION

PySpin is a wrapper for FLIR Integrated Imaging Solutions' Spinnaker library.

FLIR Integrated Imaging Solutions' website is located at https://https://www.flir.com/iis/machine-vision/

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

SOFTWARE LICENSING INFORMATION

Component	License
PySpin	Copyright (c) 2001-2019 FLIR Systems, Inc. All Rights Reserved. This software is the confidential and proprietary information of FLIR Integrated Imaging Solutions, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with FLIR Integrated Imaging Solutions, Inc. (FLIR). FLIR MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FLIR SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
GenICam	GenICam License 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	LGPv2.1 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/a9l9yA

EVENT CLASSES

3.1 PySpin.ArrivalEvent

class PySpin.**ArrivalEvent**
> An event handler for capturing the device arrival event.
>
> C++ includes: ArrivalEvent.h
>
> **OnDeviceArrival**(*self*, *serialNumber*)
>
>> **Parameters serialNumber** (*uint64_t*) –
>>
>> virtual void Spinnaker::ArrivalEvent::OnDeviceArrival(uint64_t serialNumber)=0
>>
>> Callback to the device arrival event.
>
> **thisown**
> > The membership flag

3.2 PySpin.DeviceEvent

class PySpin.**DeviceEvent**
> A handler to device events.
>
> C++ includes: DeviceEvent.h
>
> **GetDeviceEventId**(*self*) → uint64_t

> **Parameters self** (*Spinnaker::DeviceEvent const **) –

uint64_t Spinnaker::DeviceEvent::GetDeviceEventId() const

Get the ID of the device event.

The device event ID

GetDeviceEventName (*self*) → gcstring

> **Parameters self** (*Spinnaker::DeviceEvent const **) –

GenICam::gcstring Spinnaker::DeviceEvent::GetDeviceEventName() const

Get the name of the device event.

The device event name

OnDeviceEvent (*self*, *eventName*)

> **Parameters eventName** (*Spinnaker::GenICam::gcstring*) –

virtual void Spinnaker::DeviceEvent::OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0

Device event callback.

eventName: The name of the event

thisown
The membership flag

3.3 PySpin.Event

class PySpin.**Event** (**args*, ***kwargs*)
The base class for all event types.

C++ includes: Event.h

GetEventPayloadData (*self*) → PyObject *

> **Parameters self** (*Spinnaker::Event **) –

const uint8_t* Spinnaker::Event::GetEventPayloadData()

Gets the event payload data

The event payload data

GetEventPayloadDataSize (*self*) → size_t const

> **Parameters self** (*Spinnaker::Event **) –

const size_t Spinnaker::Event::GetEventPayloadDataSize()

Gets the event payload data size

The event payload data size

GetEventType (*self*) → Spinnaker::EventType

> **Parameters self** (*Spinnaker::Event **) –

EventType Spinnaker::Event::GetEventType()

Gets the event type

The event type

SetEventType(*self*, *eventType*)

>> Parameters **eventType** (*enum Spinnaker::EventType*) –

> void Spinnaker::Event::SetEventType(EventType eventType)

> Sets the event type

> eventType: The event type

thisown

> The membership flag

3.4 PySpin.ImageEvent

class PySpin.**ImageEvent**

> A handler for capturing image arrival events.

> C++ includes: ImageEvent.h

> **OnImageEvent**(*self*, *image*)

>> Parameters **image** (*Spinnaker::ImagePtr*) –

> virtual void Spinnaker::ImageEvent::OnImageEvent(ImagePtr image)=0

> Image event callback

> image: The ImagePtr object

thisown

> The membership flag

3.5 PySpin.InterfaceEvent

class PySpin.**InterfaceEvent**

> A handler to device arrival and removal events on all interfaces.

> C++ includes: InterfaceEvent.h

> **OnDeviceArrival**(*self*, *serialNumber*)

>> Parameters **serialNumber** (*uint64_t*) –

> virtual void Spinnaker::InterfaceEvent::OnDeviceArrival(uint64_t serialNumber)=0

> Device arrival event callback.

> **OnDeviceRemoval**(*self*, *serialNumber*)

>> Parameters **serialNumber** (*uint64_t*) –

> virtual void Spinnaker::InterfaceEvent::OnDeviceRemoval(uint64_t serialNumber)=0

> Callback to the device removal event.

> serialNumber: The serial number of the removed device

thisown

> The membership flag

3.6 PySpin.LoggingEvent

class PySpin.**LoggingEvent**
    An event handler for capturing the device logging event.

    C++ includes: LoggingEvent.h

    OnLogEvent(*self*, *eventPtr*)

            Parameters eventPtr(*Spinnaker::LoggingEventDataPtr*) –

        virtual void Spinnaker::LoggingEvent::OnLogEvent(LoggingEventDataPtr eventPtr)=0

        The callback for the log event.

        eventPtr: The logging event pointer

    thisown
        The membership flag

3.7 PySpin.LoggingEventDataPtr

class PySpin.**LoggingEventDataPtr**(**args*)
    A reference tracked pointer to the LoggingEvent object.

    C++ includes: LoggingEventDataPtr.h

    thisown
        The membership flag

3.8 PySpin.RemovalEvent

class PySpin.**RemovalEvent**
    An event handler for capturing the device removal event.

    C++ includes: RemovalEvent.h

    OnDeviceRemoval(*self*, *serialNumber*)

            Parameters serialNumber(*uint64_t*) –

        virtual void Spinnaker::RemovalEvent::OnDeviceRemoval(uint64_t serialNumber)=0

        Device removal event callback.

        serialNumber: The serial number of the device removed

    thisown
        The membership flag

PYSPIN CLASSES

4.1 PySpin.AVIRecorder

class PySpin.**ArrivalEvent**

An event handler for capturing the device arrival event.

C++ includes: ArrivalEvent.h

OnDeviceArrival(*self*, *serialNumber*)

> **Parameters** **serialNumber** (*uint64_t*) –

virtual void Spinnaker::ArrivalEvent::OnDeviceArrival(uint64_t serialNumber)=0

Callback to the device arrival event.

thisown
　　The membership flag

4.2 PySpin.BasePtr

4.3 PySpin.Camera

class PySpin.**Camera**(**args*, ***kwargs*)
　　The camera object class.

　　C++ includes: Camera.h

　　AasRoiEnable
　　　　Camera_AasRoiEnable_get(self) -> IBoolean

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AasRoiHeight
　　　　Camera_AasRoiHeight_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AasRoiOffsetX
　　　　Camera_AasRoiOffsetX_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AasRoiOffsetY
　　　　Camera_AasRoiOffsetY_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AasRoiWidth
　　　　Camera_AasRoiWidth_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AcquisitionAbort
　　　　Camera_AcquisitionAbort_get(self) -> ICommand

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AcquisitionArm
　　　　Camera_AcquisitionArm_get(self) -> ICommand

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AcquisitionBurstFrameCount
　　　　Camera_AcquisitionBurstFrameCount_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AcquisitionFrameCount
　　　　Camera_AcquisitionFrameCount_get(self) -> IInteger

　　　　　　Parameters self (*Spinnaker::Camera **) –

　　AcquisitionFrameRate
　　　　Camera_AcquisitionFrameRate_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionFrameRateEnable
Camera_AcquisitionFrameRateEnable_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionLineRate
Camera_AcquisitionLineRate_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionMode
Camera_AcquisitionMode_get(self) -> IEnumerationT_AcquisitionModeEnums

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionResultingFrameRate
Camera_AcquisitionResultingFrameRate_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionStart
Camera_AcquisitionStart_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionStatus
Camera_AcquisitionStatus_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionStatusSelector
Camera_AcquisitionStatusSelector_get(self) -> IEnumerationT_AcquisitionStatusSelectorEnums

> Parameters **self** (*Spinnaker::Camera* *) –

AcquisitionStop
Camera_AcquisitionStop_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera* *) –

ActionDeviceKey
Camera_ActionDeviceKey_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

ActionGroupKey
Camera_ActionGroupKey_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

ActionGroupMask
Camera_ActionGroupMask_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

ActionQueueSize
Camera_ActionQueueSize_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

ActionSelector
Camera_ActionSelector_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

ActionUnconditionalMode
    Camera_ActionUnconditionalMode_get(self) -> IEnumerationT_ActionUnconditionalModeEnums

        Parameters self(*Spinnaker::Camera* *) –

AdcBitDepth
    Camera_AdcBitDepth_get(self) -> IEnumerationT_AdcBitDepthEnums

        Parameters self(*Spinnaker::Camera* *) –

AutoAlgorithmSelector
    Camera_AutoAlgorithmSelector_get(self) -> IEnumerationT_AutoAlgorithmSelectorEnums

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureControlLoopDamping
    Camera_AutoExposureControlLoopDamping_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureControlPriority
    Camera_AutoExposureControlPriority_get(self) -> IEnumerationT_AutoExposureControlPriorityEnums

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureEVCompensation
    Camera_AutoExposureEVCompensation_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureExposureTimeLowerLimit
    Camera_AutoExposureExposureTimeLowerLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureExposureTimeUpperLimit
    Camera_AutoExposureExposureTimeUpperLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureGainLowerLimit
    Camera_AutoExposureGainLowerLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureGainUpperLimit
    Camera_AutoExposureGainUpperLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureGreyValueLowerLimit
    Camera_AutoExposureGreyValueLowerLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureGreyValueUpperLimit
    Camera_AutoExposureGreyValueUpperLimit_get(self) -> IFloat

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureLightingMode
    Camera_AutoExposureLightingMode_get(self) -> IEnumerationT_AutoExposureLightingModeEnums

        Parameters self(*Spinnaker::Camera* *) –

AutoExposureMeteringMode
    Camera_AutoExposureMeteringMode_get(self) -> IEnumerationT_AutoExposureMeteringModeEnums

Parameters **self** (*Spinnaker::Camera* *) –

AutoExposureTargetGreyValue
Camera_AutoExposureTargetGreyValue_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

AutoExposureTargetGreyValueAuto
Camera_AutoExposureTargetGreyValueAuto_get(self) -> IEnumerationT_AutoExposureTargetGreyValueAutoEnums

Parameters **self** (*Spinnaker::Camera* *) –

BalanceRatio
Camera_BalanceRatio_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

BalanceRatioSelector
Camera_BalanceRatioSelector_get(self) -> IEnumerationT_BalanceRatioSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

BalanceWhiteAuto
Camera_BalanceWhiteAuto_get(self) -> IEnumerationT_BalanceWhiteAutoEnums

Parameters **self** (*Spinnaker::Camera* *) –

BalanceWhiteAutoDamping
Camera_BalanceWhiteAutoDamping_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

BalanceWhiteAutoLowerLimit
Camera_BalanceWhiteAutoLowerLimit_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

BalanceWhiteAutoProfile
Camera_BalanceWhiteAutoProfile_get(self) -> IEnumerationT_BalanceWhiteAutoProfileEnums

Parameters **self** (*Spinnaker::Camera* *) –

BalanceWhiteAutoUpperLimit
Camera_BalanceWhiteAutoUpperLimit_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

BinningHorizontal
Camera_BinningHorizontal_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

BinningHorizontalMode
Camera_BinningHorizontalMode_get(self) -> IEnumerationT_BinningHorizontalModeEnums

Parameters **self** (*Spinnaker::Camera* *) –

BinningSelector
Camera_BinningSelector_get(self) -> IEnumerationT_BinningSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

BinningVertical
Camera_BinningVertical_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

BinningVerticalMode
Camera_BinningVerticalMode_get(self) -> IEnumerationT_BinningVerticalModeEnums

    Parameters self (*Spinnaker::Camera **) –

BlackLevel
Camera_BlackLevel_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

BlackLevelAuto
Camera_BlackLevelAuto_get(self) -> IEnumerationT_BlackLevelAutoEnums

    Parameters self (*Spinnaker::Camera **) –

BlackLevelAutoBalance
Camera_BlackLevelAutoBalance_get(self) -> IEnumerationT_BlackLevelAutoBalanceEnums

    Parameters self (*Spinnaker::Camera **) –

BlackLevelClampingEnable
Camera_BlackLevelClampingEnable_get(self) -> IBoolean

    Parameters self (*Spinnaker::Camera **) –

BlackLevelRaw
Camera_BlackLevelRaw_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

BlackLevelSelector
Camera_BlackLevelSelector_get(self) -> IEnumerationT_BlackLevelSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

BsiFlatFieldCorrectionAuto
Camera_BsiFlatFieldCorrectionAuto_get(self) -> IEnumerationT_BsiFlatFieldCorrectionAutoEnums

    Parameters self (*Spinnaker::Camera **) –

BsiFlatFieldCorrectionAutoDamping
Camera_BsiFlatFieldCorrectionAutoDamping_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

BsiFlatFieldCorrectionEnable
Camera_BsiFlatFieldCorrectionEnable_get(self) -> IBoolean

    Parameters self (*Spinnaker::Camera **) –

BsiFlatFieldCorrectionGain
Camera_BsiFlatFieldCorrectionGain_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

BsiFlatFieldCorrectionGainSelector
Camera_BsiFlatFieldCorrectionGainSelector_get(self) -> IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

ChunkBlackLevel
Camera_ChunkBlackLevel_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

ChunkBlackLevelSelector
Camera_ChunkBlackLevelSelector_get(self) -> IEnumerationT_ChunkBlackLevelSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

ChunkCRC
   Camera_ChunkCRC_get(self) -> IInteger

   Parameters **self** (*Spinnaker::Camera **) –

ChunkCounterSelector
   Camera_ChunkCounterSelector_get(self) -> IEnumerationT_ChunkCounterSelectorEnums

   Parameters **self** (*Spinnaker::Camera **) –

ChunkCounterValue
   Camera_ChunkCounterValue_get(self) -> IInteger

   Parameters **self** (*Spinnaker::Camera **) –

ChunkEnable
   Camera_ChunkEnable_get(self) -> IBoolean

   Parameters **self** (*Spinnaker::Camera **) –

ChunkEncoderSelector
   Camera_ChunkEncoderSelector_get(self) -> IEnumerationT_ChunkEncoderSelectorEnums

   Parameters **self** (*Spinnaker::Camera **) –

ChunkEncoderStatus
   Camera_ChunkEncoderStatus_get(self) -> IEnumerationT_ChunkEncoderStatusEnums

   Parameters **self** (*Spinnaker::Camera **) –

ChunkEncoderValue
   Camera_ChunkEncoderValue_get(self) -> IInteger

   Parameters **self** (*Spinnaker::Camera **) –

ChunkExposureEndLineStatusAll
   Camera_ChunkExposureEndLineStatusAll_get(self) -> IInteger

   Parameters **self** (*Spinnaker::Camera **) –

ChunkExposureTime
   Camera_ChunkExposureTime_get(self) -> IFloat

   Parameters **self** (*Spinnaker::Camera **) –

ChunkExposureTimeSelector
   Camera_ChunkExposureTimeSelector_get(self) -> IEnumerationT_ChunkExposureTimeSelectorEnums

   Parameters **self** (*Spinnaker::Camera **) –

ChunkFrameID
   Camera_ChunkFrameID_get(self) -> IInteger

   Parameters **self** (*Spinnaker::Camera **) –

ChunkGain
   Camera_ChunkGain_get(self) -> IFloat

   Parameters **self** (*Spinnaker::Camera **) –

ChunkGainSelector
   Camera_ChunkGainSelector_get(self) -> IEnumerationT_ChunkGainSelectorEnums

   Parameters **self** (*Spinnaker::Camera **) –

ChunkHeight
Camera_ChunkHeight_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkImage
Camera_ChunkImage_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkImageComponent
Camera_ChunkImageComponent_get(self) -> IEnumerationT_ChunkImageComponentEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkLinePitch
Camera_ChunkLinePitch_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkLineStatusAll
Camera_ChunkLineStatusAll_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkModeActive
Camera_ChunkModeActive_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

ChunkOffsetX
Camera_ChunkOffsetX_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkOffsetY
Camera_ChunkOffsetY_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkPartSelector
Camera_ChunkPartSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkPixelDynamicRangeMax
Camera_ChunkPixelDynamicRangeMax_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkPixelDynamicRangeMin
Camera_ChunkPixelDynamicRangeMin_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkPixelFormat
Camera_ChunkPixelFormat_get(self) -> IEnumerationT_ChunkPixelFormatEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkRegionID
Camera_ChunkRegionID_get(self) -> IEnumerationT_ChunkRegionIDEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dAxisMax
Camera_ChunkScan3dAxisMax_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dAxisMin
Camera_ChunkScan3dAxisMin_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateOffset
Camera_ChunkScan3dCoordinateOffset_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateReferenceSelector
Camera_ChunkScan3dCoordinateReferenceSelector_get(self) -> IEnumera-
tionT_ChunkScan3dCoordinateReferenceSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateReferenceValue
Camera_ChunkScan3dCoordinateReferenceValue_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateScale
Camera_ChunkScan3dCoordinateScale_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateSelector
Camera_ChunkScan3dCoordinateSelector_get(self) -> IEnumerationT_ChunkScan3dCoordinateSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateSystem
Camera_ChunkScan3dCoordinateSystem_get(self) -> IEnumerationT_ChunkScan3dCoordinateSystemEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateSystemReference
Camera_ChunkScan3dCoordinateSystemReference_get(self) -> IEnumera-
tionT_ChunkScan3dCoordinateSystemReferenceEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateTransformSelector
Camera_ChunkScan3dCoordinateTransformSelector_get(self) -> IEnumera-
tionT_ChunkScan3dCoordinateTransformSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dDistanceUnit
Camera_ChunkScan3dDistanceUnit_get(self) -> IEnumerationT_ChunkScan3dDistanceUnitEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dInvalidDataFlag
Camera_ChunkScan3dInvalidDataFlag_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dInvalidDataValue
Camera_ChunkScan3dInvalidDataValue_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dOutputMode
    Camera_ChunkScan3dOutputMode_get(self) -> IEnumerationT_ChunkScan3dOutputModeEnums

        Parameters self (*Spinnaker::Camera **) –

ChunkScan3dTransformValue
    Camera_ChunkScan3dTransformValue_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

ChunkScanLineSelector
    Camera_ChunkScanLineSelector_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ChunkSelector
    Camera_ChunkSelector_get(self) -> IEnumerationT_ChunkSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

ChunkSequencerSetActive
    Camera_ChunkSequencerSetActive_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ChunkSerialData
    Camera_ChunkSerialData_get(self) -> IString

        Parameters self (*Spinnaker::Camera **) –

ChunkSerialDataLength
    Camera_ChunkSerialDataLength_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ChunkSerialReceiveOverflow
    Camera_ChunkSerialReceiveOverflow_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

ChunkSourceID
    Camera_ChunkSourceID_get(self) -> IEnumerationT_ChunkSourceIDEnums

        Parameters self (*Spinnaker::Camera **) –

ChunkStreamChannelID
    Camera_ChunkStreamChannelID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ChunkTimerSelector
    Camera_ChunkTimerSelector_get(self) -> IEnumerationT_ChunkTimerSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

ChunkTimerValue
    Camera_ChunkTimerValue_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

ChunkTimestamp
    Camera_ChunkTimestamp_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ChunkTimestampLatchValue
    Camera_ChunkTimestampLatchValue_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

ChunkTransferBlockID
Camera_ChunkTransferBlockID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

ChunkTransferQueueCurrentBlockCount
Camera_ChunkTransferQueueCurrentBlockCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

ChunkTransferStreamID
Camera_ChunkTransferStreamID_get(self) -> IEnumerationT_ChunkTransferStreamIDEnums

Parameters **self** (*Spinnaker::Camera **) –

ChunkWidth
Camera_ChunkWidth_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

ClConfiguration
Camera_ClConfiguration_get(self) -> IEnumerationT_ClConfigurationEnums

Parameters **self** (*Spinnaker::Camera **) –

ClTimeSlotsCount
Camera_ClTimeSlotsCount_get(self) -> IEnumerationT_ClTimeSlotsCountEnums

Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationEnable
Camera_ColorTransformationEnable_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationSelector
Camera_ColorTransformationSelector_get(self) -> IEnumerationT_ColorTransformationSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationValue
Camera_ColorTransformationValue_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationValueSelector
Camera_ColorTransformationValueSelector_get(self) -> IEnumerationT_ColorTransformationValueSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

CounterDelay
Camera_CounterDelay_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

CounterDuration
Camera_CounterDuration_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

CounterEventActivation
Camera_CounterEventActivation_get(self) -> IEnumerationT_CounterEventActivationEnums

Parameters **self** (*Spinnaker::Camera **) –

CounterEventSource
Camera_CounterEventSource_get(self) -> IEnumerationT_CounterEventSourceEnums

    Parameters self (*Spinnaker::Camera **) –

CounterReset
Camera_CounterReset_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

CounterResetActivation
Camera_CounterResetActivation_get(self) -> IEnumerationT_CounterResetActivationEnums

    Parameters self (*Spinnaker::Camera **) –

CounterResetSource
Camera_CounterResetSource_get(self) -> IEnumerationT_CounterResetSourceEnums

    Parameters self (*Spinnaker::Camera **) –

CounterSelector
Camera_CounterSelector_get(self) -> IEnumerationT_CounterSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

CounterStatus
Camera_CounterStatus_get(self) -> IEnumerationT_CounterStatusEnums

    Parameters self (*Spinnaker::Camera **) –

CounterTriggerActivation
Camera_CounterTriggerActivation_get(self) -> IEnumerationT_CounterTriggerActivationEnums

    Parameters self (*Spinnaker::Camera **) –

CounterTriggerSource
Camera_CounterTriggerSource_get(self) -> IEnumerationT_CounterTriggerSourceEnums

    Parameters self (*Spinnaker::Camera **) –

CounterValue
Camera_CounterValue_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

CounterValueAtReset
Camera_CounterValueAtReset_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

CxpConnectionSelector
Camera_CxpConnectionSelector_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

CxpConnectionTestErrorCount
Camera_CxpConnectionTestErrorCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

CxpConnectionTestMode
Camera_CxpConnectionTestMode_get(self) -> IEnumerationT_CxpConnectionTestModeEnums

    Parameters self (*Spinnaker::Camera **) –

CxpConnectionTestPacketCount
Camera_CxpConnectionTestPacketCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

CxpLinkConfiguration
Camera_CxpLinkConfiguration_get(self) -> IEnumerationT_CxpLinkConfigurationEnums

Parameters **self** (*Spinnaker::Camera* *) –

CxpLinkConfigurationPreferred
Camera_CxpLinkConfigurationPreferred_get(self) -> IEnumerationT_CxpLinkConfigurationPreferredEnums

Parameters **self** (*Spinnaker::Camera* *) –

CxpLinkConfigurationStatus
Camera_CxpLinkConfigurationStatus_get(self) -> IEnumerationT_CxpLinkConfigurationStatusEnums

Parameters **self** (*Spinnaker::Camera* *) –

CxpPoCxpAuto
Camera_CxpPoCxpAuto_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera* *) –

CxpPoCxpStatus
Camera_CxpPoCxpStatus_get(self) -> IEnumerationT_CxpPoCxpStatusEnums

Parameters **self** (*Spinnaker::Camera* *) –

CxpPoCxpTripReset
Camera_CxpPoCxpTripReset_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera* *) –

CxpPoCxpTurnOff
Camera_CxpPoCxpTurnOff_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera* *) –

DecimationHorizontal
Camera_DecimationHorizontal_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

DecimationHorizontalMode
Camera_DecimationHorizontalMode_get(self) -> IEnumerationT_DecimationHorizontalModeEnums

Parameters **self** (*Spinnaker::Camera* *) –

DecimationSelector
Camera_DecimationSelector_get(self) -> IEnumerationT_DecimationSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

DecimationVertical
Camera_DecimationVertical_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

DecimationVerticalMode
Camera_DecimationVerticalMode_get(self) -> IEnumerationT_DecimationVerticalModeEnums

Parameters **self** (*Spinnaker::Camera* *) –

DefectTableApply
Camera_DefectTableApply_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera* *) –

DefectTableCoordinateX
    Camera_DefectTableCoordinateX_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DefectTableCoordinateY
    Camera_DefectTableCoordinateY_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DefectTableFactoryRestore
    Camera_DefectTableFactoryRestore_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

DefectTableIndex
    Camera_DefectTableIndex_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DefectTablePixelCount
    Camera_DefectTablePixelCount_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DefectTableSave
    Camera_DefectTableSave_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

Deinterlacing
    Camera_Deinterlacing_get(self) -> IEnumerationT_DeinterlacingEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceCharacterSet
    Camera_DeviceCharacterSet_get(self) -> IEnumerationT_DeviceCharacterSetEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceClockFrequency
    Camera_DeviceClockFrequency_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

DeviceClockSelector
    Camera_DeviceClockSelector_get(self) -> IEnumerationT_DeviceClockSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionSelector
    Camera_DeviceConnectionSelector_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionSpeed
    Camera_DeviceConnectionSpeed_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionStatus
    Camera_DeviceConnectionStatus_get(self) -> IEnumerationT_DeviceConnectionStatusEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceEventChannelCount
    Camera_DeviceEventChannelCount_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceFamilyName
Camera_DeviceFamilyName_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceFeaturePersistenceEnd
Camera_DeviceFeaturePersistenceEnd_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceFeaturePersistenceStart
Camera_DeviceFeaturePersistenceStart_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceFirmwareVersion
Camera_DeviceFirmwareVersion_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceGenCPVersionMajor
Camera_DeviceGenCPVersionMajor_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceGenCPVersionMinor
Camera_DeviceGenCPVersionMinor_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceID
Camera_DeviceID_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceIndicatorMode
Camera_DeviceIndicatorMode_get(self) -> IEnumerationT_DeviceIndicatorModeEnums

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkBandwidthReserve
Camera_DeviceLinkBandwidthReserve_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkCommandTimeout
Camera_DeviceLinkCommandTimeout_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkConnectionCount
Camera_DeviceLinkConnectionCount_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkCurrentThroughput
Camera_DeviceLinkCurrentThroughput_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkHeartbeatMode
Camera_DeviceLinkHeartbeatMode_get(self) -> IEnumerationT_DeviceLinkHeartbeatModeEnums

> Parameters **self** (*Spinnaker::Camera* *) –

DeviceLinkHeartbeatTimeout
Camera_DeviceLinkHeartbeatTimeout_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

DeviceLinkSelector
Camera_DeviceLinkSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceLinkSpeed
Camera_DeviceLinkSpeed_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceLinkThroughputLimit
Camera_DeviceLinkThroughputLimit_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceLinkThroughputLimitMode
Camera_DeviceLinkThroughputLimitMode_get(self) -> IEnumerationT_DeviceLinkThroughputLimitModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestEntrySelector
Camera_DeviceManifestEntrySelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestPrimaryURL
Camera_DeviceManifestPrimaryURL_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestSchemaMajorVersion
Camera_DeviceManifestSchemaMajorVersion_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestSchemaMinorVersion
Camera_DeviceManifestSchemaMinorVersion_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestSecondaryURL
Camera_DeviceManifestSecondaryURL_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestXMLMajorVersion
Camera_DeviceManifestXMLMajorVersion_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestXMLMinorVersion
Camera_DeviceManifestXMLMinorVersion_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManifestXMLSubMinorVersion
Camera_DeviceManifestXMLSubMinorVersion_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceManufacturerInfo
Camera_DeviceManufacturerInfo_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera **) –

DeviceMaxThroughput
    Camera_DeviceMaxThroughput_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

DeviceModelName
    Camera_DeviceModelName_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera **) –

DevicePowerSupplySelector
    Camera_DevicePowerSupplySelector_get(self) -> IEnumerationT_DevicePowerSupplySelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

DeviceRegistersCheck
    Camera_DeviceRegistersCheck_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

DeviceRegistersEndianness
    Camera_DeviceRegistersEndianness_get(self) -> IEnumerationT_DeviceRegistersEndiannessEnums

> Parameters **self** (*Spinnaker::Camera **) –

DeviceRegistersStreamingEnd
    Camera_DeviceRegistersStreamingEnd_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

DeviceRegistersStreamingStart
    Camera_DeviceRegistersStreamingStart_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

DeviceRegistersValid
    Camera_DeviceRegistersValid_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

DeviceReset
    Camera_DeviceReset_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

DeviceSFNCVersionMajor
    Camera_DeviceSFNCVersionMajor_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

DeviceSFNCVersionMinor
    Camera_DeviceSFNCVersionMinor_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

DeviceSFNCVersionSubMinor
    Camera_DeviceSFNCVersionSubMinor_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

DeviceScanType
    Camera_DeviceScanType_get(self) -> IEnumerationT_DeviceScanTypeEnums

> Parameters **self** (*Spinnaker::Camera **) –

DeviceSerialNumber
Camera_DeviceSerialNumber_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera **) –

DeviceSerialPortBaudRate
Camera_DeviceSerialPortBaudRate_get(self) -> IEnumerationT_DeviceSerialPortBaudRateEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceSerialPortSelector
Camera_DeviceSerialPortSelector_get(self) -> IEnumerationT_DeviceSerialPortSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelCount
Camera_DeviceStreamChannelCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelEndianness
Camera_DeviceStreamChannelEndianness_get(self) -> IEnumerationT_DeviceStreamChannelEndiannessEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelLink
Camera_DeviceStreamChannelLink_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelPacketSize
Camera_DeviceStreamChannelPacketSize_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelSelector
Camera_DeviceStreamChannelSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelType
Camera_DeviceStreamChannelType_get(self) -> IEnumerationT_DeviceStreamChannelTypeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLType
Camera_DeviceTLType_get(self) -> IEnumerationT_DeviceTLTypeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionMajor
Camera_DeviceTLVersionMajor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionMinor
Camera_DeviceTLVersionMinor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionSubMinor
Camera_DeviceTLVersionSubMinor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTapGeometry
Camera_DeviceTapGeometry_get(self) -> IEnumerationT_DeviceTapGeometryEnums

Parameters **self** (*Spinnaker::Camera* *) –

DeviceTemperature
    Camera_DeviceTemperature_get(self) -> IFloat

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceTemperatureSelector
    Camera_DeviceTemperatureSelector_get(self) -> IEnumerationT_DeviceTemperatureSelectorEnums

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceType
    Camera_DeviceType_get(self) -> IEnumerationT_DeviceTypeEnums

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceUptime
    Camera_DeviceUptime_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceUserID
    Camera_DeviceUserID_get(self) -> IString

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceVendorName
    Camera_DeviceVendorName_get(self) -> IString

    Parameters **self** (*Spinnaker::Camera* *) –

DeviceVersion
    Camera_DeviceVersion_get(self) -> IString

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderDivider
    Camera_EncoderDivider_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderMode
    Camera_EncoderMode_get(self) -> IEnumerationT_EncoderModeEnums

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderOutputMode
    Camera_EncoderOutputMode_get(self) -> IEnumerationT_EncoderOutputModeEnums

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderReset
    Camera_EncoderReset_get(self) -> ICommand

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderResetActivation
    Camera_EncoderResetActivation_get(self) -> IEnumerationT_EncoderResetActivationEnums

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderResetSource
    Camera_EncoderResetSource_get(self) -> IEnumerationT_EncoderResetSourceEnums

    Parameters **self** (*Spinnaker::Camera* *) –

EncoderSelector
Camera_EncoderSelector_get(self) -> IEnumerationT_EncoderSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

EncoderSourceA
Camera_EncoderSourceA_get(self) -> IEnumerationT_EncoderSourceAEnums

> **Parameters self** (*Spinnaker::Camera **) –

EncoderSourceB
Camera_EncoderSourceB_get(self) -> IEnumerationT_EncoderSourceBEnums

> **Parameters self** (*Spinnaker::Camera **) –

EncoderStatus
Camera_EncoderStatus_get(self) -> IEnumerationT_EncoderStatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

EncoderTimeout
Camera_EncoderTimeout_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

EncoderValue
Camera_EncoderValue_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EncoderValueAtReset
Camera_EncoderValueAtReset_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EnumerationCount
Camera_EnumerationCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionEnd
Camera_EventAcquisitionEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionEndFrameID
Camera_EventAcquisitionEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionEndTimestamp
Camera_EventAcquisitionEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionError
Camera_EventAcquisitionError_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionErrorFrameID
Camera_EventAcquisitionErrorFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventAcquisitionErrorTimestamp
Camera_EventAcquisitionErrorTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionStart
Camera_EventAcquisitionStart_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionStartFrameID
Camera_EventAcquisitionStartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionStartTimestamp
Camera_EventAcquisitionStartTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferEnd
Camera_EventAcquisitionTransferEnd_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferEndFrameID
Camera_EventAcquisitionTransferEndFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferEndTimestamp
Camera_EventAcquisitionTransferEndTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferStart
Camera_EventAcquisitionTransferStart_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferStartFrameID
Camera_EventAcquisitionTransferStartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTransferStartTimestamp
Camera_EventAcquisitionTransferStartTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTrigger
Camera_EventAcquisitionTrigger_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTriggerFrameID
Camera_EventAcquisitionTriggerFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventAcquisitionTriggerTimestamp
Camera_EventAcquisitionTriggerTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventActionLate
Camera_EventActionLate_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventActionLateFrameID
Camera_EventActionLateFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventActionLateTimestamp
Camera_EventActionLateTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0End
Camera_EventCounter0End_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0EndFrameID
Camera_EventCounter0EndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0EndTimestamp
Camera_EventCounter0EndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0Start
Camera_EventCounter0Start_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0StartFrameID
Camera_EventCounter0StartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter0StartTimestamp
Camera_EventCounter0StartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1End
Camera_EventCounter1End_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1EndFrameID
Camera_EventCounter1EndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1EndTimestamp
Camera_EventCounter1EndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1Start
Camera_EventCounter1Start_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1StartFrameID
Camera_EventCounter1StartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera *) –

EventCounter1StartTimestamp
Camera_EventCounter1StartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0Restarted
    Camera_EventEncoder0Restarted_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0RestartedFrameID
    Camera_EventEncoder0RestartedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0RestartedTimestamp
    Camera_EventEncoder0RestartedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0Stopped
    Camera_EventEncoder0Stopped_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0StoppedFrameID
    Camera_EventEncoder0StoppedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0StoppedTimestamp
    Camera_EventEncoder0StoppedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1Restarted
    Camera_EventEncoder1Restarted_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1RestartedFrameID
    Camera_EventEncoder1RestartedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1RestartedTimestamp
    Camera_EventEncoder1RestartedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1Stopped
    Camera_EventEncoder1Stopped_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1StoppedFrameID
    Camera_EventEncoder1StoppedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1StoppedTimestamp
    Camera_EventEncoder1StoppedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventError
    Camera_EventError_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventErrorCode
Camera_EventErrorCode_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventErrorFrameID
Camera_EventErrorFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventErrorTimestamp
Camera_EventErrorTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureEnd
Camera_EventExposureEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureEndFrameID
Camera_EventExposureEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureEndTimestamp
Camera_EventExposureEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureStart
Camera_EventExposureStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureStartFrameID
Camera_EventExposureStartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventExposureStartTimestamp
Camera_EventExposureStartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventFrameBurstEnd
Camera_EventFrameBurstEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventFrameBurstEndFrameID
Camera_EventFrameBurstEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventFrameBurstEndTimestamp
Camera_EventFrameBurstEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventFrameBurstStart
Camera_EventFrameBurstStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EventFrameBurstStartFrameID
Camera_EventFrameBurstStartFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameBurstStartTimestamp
Camera_EventFrameBurstStartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameEnd
Camera_EventFrameEnd_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameEndFrameID
Camera_EventFrameEndFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameEndTimestamp
Camera_EventFrameEndTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameStart
Camera_EventFrameStart_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameStartFrameID
Camera_EventFrameStartFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameStartTimestamp
Camera_EventFrameStartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferEnd
Camera_EventFrameTransferEnd_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferEndFrameID
Camera_EventFrameTransferEndFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferEndTimestamp
Camera_EventFrameTransferEndTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferStart
Camera_EventFrameTransferStart_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferStartFrameID
Camera_EventFrameTransferStartFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTransferStartTimestamp
Camera_EventFrameTransferStartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

EventFrameTrigger
   Camera_EventFrameTrigger_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventFrameTriggerFrameID
   Camera_EventFrameTriggerFrameID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventFrameTriggerTimestamp
   Camera_EventFrameTriggerTimestamp_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0AnyEdge
   Camera_EventLine0AnyEdge_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0AnyEdgeFrameID
   Camera_EventLine0AnyEdgeFrameID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0AnyEdgeTimestamp
   Camera_EventLine0AnyEdgeTimestamp_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdge
   Camera_EventLine0FallingEdge_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdgeFrameID
   Camera_EventLine0FallingEdgeFrameID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdgeTimestamp
   Camera_EventLine0FallingEdgeTimestamp_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdge
   Camera_EventLine0RisingEdge_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdgeFrameID
   Camera_EventLine0RisingEdgeFrameID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdgeTimestamp
   Camera_EventLine0RisingEdgeTimestamp_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine1AnyEdge
   Camera_EventLine1AnyEdge_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

EventLine1AnyEdgeFrameID
   Camera_EventLine1AnyEdgeFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1AnyEdgeTimestamp
    Camera_EventLine1AnyEdgeTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1FallingEdge
    Camera_EventLine1FallingEdge_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1FallingEdgeFrameID
    Camera_EventLine1FallingEdgeFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1FallingEdgeTimestamp
    Camera_EventLine1FallingEdgeTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdge
    Camera_EventLine1RisingEdge_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdgeFrameID
    Camera_EventLine1RisingEdgeFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdgeTimestamp
    Camera_EventLine1RisingEdgeTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChange
    Camera_EventLinkSpeedChange_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChangeFrameID
    Camera_EventLinkSpeedChangeFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChangeTimestamp
    Camera_EventLinkSpeedChangeTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0
    Camera_EventLinkTrigger0_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0FrameID
    Camera_EventLinkTrigger0FrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0Timestamp
    Camera_EventLinkTrigger0Timestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger1
Camera_EventLinkTrigger1_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventLinkTrigger1FrameID
Camera_EventLinkTrigger1FrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventLinkTrigger1Timestamp
Camera_EventLinkTrigger1Timestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventNotification
Camera_EventNotification_get(self) -> IEnumerationT_EventNotificationEnums

> **Parameters self** (*Spinnaker::Camera **) –

EventSelector
Camera_EventSelector_get(self) -> IEnumerationT_EventSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChange
Camera_EventSequencerSetChange_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChangeFrameID
Camera_EventSequencerSetChangeFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChangeTimestamp
Camera_EventSequencerSetChangeTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSerialData
Camera_EventSerialData_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera **) –

EventSerialDataLength
Camera_EventSerialDataLength_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSerialPortReceive
Camera_EventSerialPortReceive_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSerialPortReceiveTimestamp
Camera_EventSerialPortReceiveTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventSerialReceiveOverflow
Camera_EventSerialReceiveOverflow_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEnd
Camera_EventStream0TransferBlockEnd_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEndFrameID
Camera_EventStream0TransferBlockEndFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEndTimestamp
Camera_EventStream0TransferBlockEndTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStart
Camera_EventStream0TransferBlockStart_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStartFrameID
Camera_EventStream0TransferBlockStartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStartTimestamp
Camera_EventStream0TransferBlockStartTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTrigger
Camera_EventStream0TransferBlockTrigger_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTriggerFrameID
Camera_EventStream0TransferBlockTriggerFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTriggerTimestamp
Camera_EventStream0TransferBlockTriggerTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEnd
Camera_EventStream0TransferBurstEnd_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEndFrameID
Camera_EventStream0TransferBurstEndFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEndTimestamp
Camera_EventStream0TransferBurstEndTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStart
Camera_EventStream0TransferBurstStart_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStartFrameID
Camera_EventStream0TransferBurstStartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStartTimestamp
Camera_EventStream0TransferBurstStartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEnd
Camera_EventStream0TransferEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEndFrameID
Camera_EventStream0TransferEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEndTimestamp
Camera_EventStream0TransferEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflow
Camera_EventStream0TransferOverflow_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflowFrameID
Camera_EventStream0TransferOverflowFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflowTimestamp
Camera_EventStream0TransferOverflowTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPause
Camera_EventStream0TransferPause_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPauseFrameID
Camera_EventStream0TransferPauseFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPauseTimestamp
Camera_EventStream0TransferPauseTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResume
Camera_EventStream0TransferResume_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResumeFrameID
Camera_EventStream0TransferResumeFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResumeTimestamp
Camera_EventStream0TransferResumeTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferStart
Camera_EventStream0TransferStart_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferStartFrameID
Camera_EventStream0TransferStartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventStream0TransferStartTimestamp
Camera_EventStream0TransferStartTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTest
Camera_EventTest_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTestTimestamp
Camera_EventTestTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0End
Camera_EventTimer0End_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0EndFrameID
Camera_EventTimer0EndFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0EndTimestamp
Camera_EventTimer0EndTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0Start
Camera_EventTimer0Start_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0StartFrameID
Camera_EventTimer0StartFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer0StartTimestamp
Camera_EventTimer0StartTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer1End
Camera_EventTimer1End_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer1EndFrameID
Camera_EventTimer1EndFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer1EndTimestamp
Camera_EventTimer1EndTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventTimer1Start
>   Camera_EventTimer1Start_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::Camera **) –

EventTimer1StartFrameID
>   Camera_EventTimer1StartFrameID_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::Camera **) –

EventTimer1StartTimestamp
>   Camera_EventTimer1StartTimestamp_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureActiveMode
>   Camera_ExposureActiveMode_get(self) -> IEnumerationT_ExposureActiveModeEnums

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureAuto
>   Camera_ExposureAuto_get(self) -> IEnumerationT_ExposureAutoEnums

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureMode
>   Camera_ExposureMode_get(self) -> IEnumerationT_ExposureModeEnums

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureTime
>   Camera_ExposureTime_get(self) -> IFloat

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureTimeMode
>   Camera_ExposureTimeMode_get(self) -> IEnumerationT_ExposureTimeModeEnums

>>   **Parameters self** (*Spinnaker::Camera **) –

ExposureTimeSelector
>   Camera_ExposureTimeSelector_get(self) -> IEnumerationT_ExposureTimeSelectorEnums

>>   **Parameters self** (*Spinnaker::Camera **) –

FactoryReset
>   Camera_FactoryReset_get(self) -> ICommand

>>   **Parameters self** (*Spinnaker::Camera **) –

FfcUserGain
>   Camera_FfcUserGain_get(self) -> IFloat

>>   **Parameters self** (*Spinnaker::Camera **) –

FfcUserGainAll
>   Camera_FfcUserGainAll_get(self) -> IRegister

>>   **Parameters self** (*Spinnaker::Camera **) –

FfcUserOffset
>   Camera_FfcUserOffset_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::Camera **) –

FfcUserOffsetAll
>   Camera_FfcUserOffsetAll_get(self) -> IRegister

Parameters **self** (*Spinnaker::Camera **) –

FfcUserTableReset
    Camera_FfcUserTableReset_get(self) -> ICommand

    Parameters **self** (*Spinnaker::Camera **) –

FfcUserTableSave
    Camera_FfcUserTableSave_get(self) -> ICommand

    Parameters **self** (*Spinnaker::Camera **) –

FfcUserTableXCoordinate
    Camera_FfcUserTableXCoordinate_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

FileAccessBuffer
    Camera_FileAccessBuffer_get(self) -> IRegister

    Parameters **self** (*Spinnaker::Camera **) –

FileAccessLength
    Camera_FileAccessLength_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

FileAccessOffset
    Camera_FileAccessOffset_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

FileOpenMode
    Camera_FileOpenMode_get(self) -> IEnumerationT_FileOpenModeEnums

    Parameters **self** (*Spinnaker::Camera **) –

FileOperationExecute
    Camera_FileOperationExecute_get(self) -> ICommand

    Parameters **self** (*Spinnaker::Camera **) –

FileOperationResult
    Camera_FileOperationResult_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

FileOperationSelector
    Camera_FileOperationSelector_get(self) -> IEnumerationT_FileOperationSelectorEnums

    Parameters **self** (*Spinnaker::Camera **) –

FileOperationStatus
    Camera_FileOperationStatus_get(self) -> IEnumerationT_FileOperationStatusEnums

    Parameters **self** (*Spinnaker::Camera **) –

FileSelector
    Camera_FileSelector_get(self) -> IEnumerationT_FileSelectorEnums

    Parameters **self** (*Spinnaker::Camera **) –

FileSize
    Camera_FileSize_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

Gain
> Camera_Gain_get(self) -> IFloat
>
>> **Parameters self** (*Spinnaker::Camera **) –

GainAuto
> Camera_GainAuto_get(self) -> IEnumerationT_GainAutoEnums
>
>> **Parameters self** (*Spinnaker::Camera **) –

GainAutoBalance
> Camera_GainAutoBalance_get(self) -> IEnumerationT_GainAutoBalanceEnums
>
>> **Parameters self** (*Spinnaker::Camera **) –

GainSelector
> Camera_GainSelector_get(self) -> IEnumerationT_GainSelectorEnums
>
>> **Parameters self** (*Spinnaker::Camera **) –

Gamma
> Camera_Gamma_get(self) -> IFloat
>
>> **Parameters self** (*Spinnaker::Camera **) –

GammaEnable
> Camera_GammaEnable_get(self) -> IBoolean
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevActiveLinkCount
> Camera_GevActiveLinkCount_get(self) -> IInteger
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCCP
> Camera_GevCCP_get(self) -> IEnumerationT_GevCCPEnums
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentDefaultGateway
> Camera_GevCurrentDefaultGateway_get(self) -> IInteger
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentIPAddress
> Camera_GevCurrentIPAddress_get(self) -> IInteger
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentIPConfigurationDHCP
> Camera_GevCurrentIPConfigurationDHCP_get(self) -> IBoolean
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentIPConfigurationLLA
> Camera_GevCurrentIPConfigurationLLA_get(self) -> IBoolean
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentIPConfigurationPersistentIP
> Camera_GevCurrentIPConfigurationPersistentIP_get(self) -> IBoolean
>
>> **Parameters self** (*Spinnaker::Camera **) –

GevCurrentPhysicalLinkConfiguration
> Camera_GevCurrentPhysicalLinkConfiguration_get(self)   ->   IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums

> > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentSubnetMask
> Camera_GevCurrentSubnetMask_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera* *) –

GevDiscoveryAckDelay
> Camera_GevDiscoveryAckDelay_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera* *) –

GevFirstURL
> Camera_GevFirstURL_get(self) -> IString

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPExtendedStatusCodes
> Camera_GevGVCPExtendedStatusCodes_get(self) -> IBoolean

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPExtendedStatusCodesSelector
> Camera_GevGVCPExtendedStatusCodesSelector_get(self)   ->   IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPHeartbeatDisable
> Camera_GevGVCPHeartbeatDisable_get(self) -> IBoolean

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPPendingAck
> Camera_GevGVCPPendingAck_get(self) -> IBoolean

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPPendingTimeout
> Camera_GevGVCPPendingTimeout_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera* *) –

GevGVSPExtendedIDMode
> Camera_GevGVSPExtendedIDMode_get(self) -> IEnumerationT_GevGVSPExtendedIDModeEnums

> > **Parameters self** (*Spinnaker::Camera* *) –

GevHeartbeatTimeout
> Camera_GevHeartbeatTimeout_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588
> Camera_GevIEEE1588_get(self) -> IBoolean

> > **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588ClockAccuracy
> Camera_GevIEEE1588ClockAccuracy_get(self) -> IEnumerationT_GevIEEE1588ClockAccuracyEnums

> > **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588Mode
Camera_GevIEEE1588Mode_get(self) -> IEnumerationT_GevIEEE1588ModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

GevIEEE1588Status
Camera_GevIEEE1588Status_get(self) -> IEnumerationT_GevIEEE1588StatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

GevIPConfigurationStatus
Camera_GevIPConfigurationStatus_get(self) -> IEnumerationT_GevIPConfigurationStatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

GevInterfaceSelector
Camera_GevInterfaceSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMACAddress
Camera_GevMACAddress_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMCDA
Camera_GevMCDA_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMCPHostPort
Camera_GevMCPHostPort_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMCRC
Camera_GevMCRC_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMCSP
Camera_GevMCSP_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevMCTT
Camera_GevMCTT_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevNumberOfInterfaces
Camera_GevNumberOfInterfaces_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

GevPAUSEFrameReception
Camera_GevPAUSEFrameReception_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

GevPAUSEFrameTransmission
Camera_GevPAUSEFrameTransmission_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

GevPersistentDefaultGateway
Camera_GevPersistentDefaultGateway_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevPersistentIPAddress
Camera_GevPersistentIPAddress_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevPersistentSubnetMask
Camera_GevPersistentSubnetMask_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevPhysicalLinkConfiguration
Camera_GevPhysicalLinkConfiguration_get(self) -> IEnumerationT_GevPhysicalLinkConfigurationEnums

Parameters **self** (*Spinnaker::Camera* *) –

GevPrimaryApplicationIPAddress
Camera_GevPrimaryApplicationIPAddress_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevPrimaryApplicationSocket
Camera_GevPrimaryApplicationSocket_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevPrimaryApplicationSwitchoverKey
Camera_GevPrimaryApplicationSwitchoverKey_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevSCCFGAllInTransmission
Camera_GevSCCFGAllInTransmission_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

GevSCCFGExtendedChunkData
Camera_GevSCCFGExtendedChunkData_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

GevSCCFGPacketResendDestination
Camera_GevSCCFGPacketResendDestination_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

GevSCCFGUnconditionalStreaming
Camera_GevSCCFGUnconditionalStreaming_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

GevSCDA
Camera_GevSCDA_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevSCPD
Camera_GevSCPD_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevSCPDirection
Camera_GevSCPDirection_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

GevSCPHostPort
Camera_GevSCPHostPort_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCPInterfaceIndex
Camera_GevSCPInterfaceIndex_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCPSBigEndian
Camera_GevSCPSBigEndian_get(self) -> IBoolean

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCPSDoNotFragment
Camera_GevSCPSDoNotFragment_get(self) -> IBoolean

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCPSFireTestPacket
Camera_GevSCPSFireTestPacket_get(self) -> IBoolean

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCPSPacketSize
Camera_GevSCPSPacketSize_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCSP
Camera_GevSCSP_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCZoneConfigurationLock
Camera_GevSCZoneConfigurationLock_get(self) -> IBoolean

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCZoneCount
Camera_GevSCZoneCount_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSCZoneDirectionAll
Camera_GevSCZoneDirectionAll_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSecondURL
Camera_GevSecondURL_get(self) -> IString

>   **Parameters self** (*Spinnaker::Camera **) –

GevStreamChannelSelector
Camera_GevStreamChannelSelector_get(self) -> IInteger

>   **Parameters self** (*Spinnaker::Camera **) –

GevSupportedOption
Camera_GevSupportedOption_get(self) -> IBoolean

>   **Parameters self** (*Spinnaker::Camera **) –

GevSupportedOptionSelector
Camera_GevSupportedOptionSelector_get(self) -> IEnumerationT_GevSupportedOptionSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

GevTimestampTickFrequency
    Camera_GevTimestampTickFrequency_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera **) –

GuiXmlManifestAddress
    Camera_GuiXmlManifestAddress_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera **) –

Height
    Camera_Height_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera **) –

HeightMax
    Camera_HeightMax_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera **) –

ImageComponentEnable
    Camera_ImageComponentEnable_get(self) -> IBoolean

        Parameters **self** (*Spinnaker::Camera **) –

ImageComponentSelector
    Camera_ImageComponentSelector_get(self) -> IEnumerationT_ImageComponentSelectorEnums

        Parameters **self** (*Spinnaker::Camera **) –

ImageCompressionBitrate
    Camera_ImageCompressionBitrate_get(self) -> IFloat

        Parameters **self** (*Spinnaker::Camera **) –

ImageCompressionJPEGFormatOption
    Camera_ImageCompressionJPEGFormatOption_get(self) -> IEnumerationT_ImageCompressionJPEGFormatOptionEnums

        Parameters **self** (*Spinnaker::Camera **) –

ImageCompressionMode
    Camera_ImageCompressionMode_get(self) -> IEnumerationT_ImageCompressionModeEnums

        Parameters **self** (*Spinnaker::Camera **) –

ImageCompressionQuality
    Camera_ImageCompressionQuality_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera **) –

ImageCompressionRateOption
    Camera_ImageCompressionRateOption_get(self) -> IEnumerationT_ImageCompressionRateOptionEnums

        Parameters **self** (*Spinnaker::Camera **) –

Init (*self*)

        Parameters **self** (*Spinnaker::Camera **) –

    void Spinnaker::Camera::Init()

IspEnable
    Camera_IspEnable_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera **) –

LUTEnable
> Camera_LUTEnable_get(self) -> IBoolean

> > Parameters **self** (*Spinnaker::Camera **) –

LUTIndex
> Camera_LUTIndex_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

LUTSelector
> Camera_LUTSelector_get(self) -> IEnumerationT_LUTSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LUTValue
> Camera_LUTValue_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

LUTValueAll
> Camera_LUTValueAll_get(self) -> IRegister

> > Parameters **self** (*Spinnaker::Camera **) –

LineFilterWidth
> Camera_LineFilterWidth_get(self) -> IFloat

> > Parameters **self** (*Spinnaker::Camera **) –

LineFormat
> Camera_LineFormat_get(self) -> IEnumerationT_LineFormatEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LineInputFilterSelector
> Camera_LineInputFilterSelector_get(self) -> IEnumerationT_LineInputFilterSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LineInverter
> Camera_LineInverter_get(self) -> IBoolean

> > Parameters **self** (*Spinnaker::Camera **) –

LineMode
> Camera_LineMode_get(self) -> IEnumerationT_LineModeEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LinePitch
> Camera_LinePitch_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

LineSelector
> Camera_LineSelector_get(self) -> IEnumerationT_LineSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LineSource
> Camera_LineSource_get(self) -> IEnumerationT_LineSourceEnums

> > Parameters **self** (*Spinnaker::Camera **) –

LineStatus
Camera_LineStatus_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

LineStatusAll
Camera_LineStatusAll_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LinkErrorCount
Camera_LinkErrorCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LinkRecoveryCount
Camera_LinkRecoveryCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LinkUptime
Camera_LinkUptime_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTInputActivation
Camera_LogicBlockLUTInputActivation_get(self) -> IEnumerationT_LogicBlockLUTInputActivationEnums

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTInputSelector
Camera_LogicBlockLUTInputSelector_get(self) -> IEnumerationT_LogicBlockLUTInputSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTInputSource
Camera_LogicBlockLUTInputSource_get(self) -> IEnumerationT_LogicBlockLUTInputSourceEnums

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTOutputValue
Camera_LogicBlockLUTOutputValue_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTOutputValueAll
Camera_LogicBlockLUTOutputValueAll_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTRowIndex
Camera_LogicBlockLUTRowIndex_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockLUTSelector
Camera_LogicBlockLUTSelector_get(self) -> IEnumerationT_LogicBlockLUTSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

LogicBlockSelector
Camera_LogicBlockSelector_get(self) -> IEnumerationT_LogicBlockSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

MaxDeviceResetTime
Camera_MaxDeviceResetTime_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

OffsetX
    Camera_OffsetX_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

OffsetY
    Camera_OffsetY_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PacketResendRequestCount
    Camera_PacketResendRequestCount_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PayloadSize
    Camera_PayloadSize_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelColorFilter
    Camera_PixelColorFilter_get(self) -> IEnumerationT_PixelColorFilterEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelDynamicRangeMax
    Camera_PixelDynamicRangeMax_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelDynamicRangeMin
    Camera_PixelDynamicRangeMin_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormat
    Camera_PixelFormat_get(self) -> IEnumerationT_PixelFormatEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormatInfoID
    Camera_PixelFormatInfoID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormatInfoSelector
    Camera_PixelFormatInfoSelector_get(self) -> IEnumerationT_PixelFormatInfoSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelSize
    Camera_PixelSize_get(self) -> IEnumerationT_PixelSizeEnums

> Parameters **self** (*Spinnaker::Camera **) –

PowerSupplyCurrent
    Camera_PowerSupplyCurrent_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

PowerSupplyVoltage
    Camera_PowerSupplyVoltage_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

RegionDestination
    Camera_RegionDestination_get(self) -> IEnumerationT_RegionDestinationEnums

        Parameters self (*Spinnaker::Camera* *) –

RegionMode
    Camera_RegionMode_get(self) -> IEnumerationT_RegionModeEnums

        Parameters self (*Spinnaker::Camera* *) –

RegionSelector
    Camera_RegionSelector_get(self) -> IEnumerationT_RegionSelectorEnums

        Parameters self (*Spinnaker::Camera* *) –

ReverseX
    Camera_ReverseX_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera* *) –

ReverseY
    Camera_ReverseY_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera* *) –

RgbTransformLightSource
    Camera_RgbTransformLightSource_get(self) -> IEnumerationT_RgbTransformLightSourceEnums

        Parameters self (*Spinnaker::Camera* *) –

Saturation
    Camera_Saturation_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera* *) –

SaturationEnable
    Camera_SaturationEnable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera* *) –

Scan3dAxisMax
    Camera_Scan3dAxisMax_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera* *) –

Scan3dAxisMin
    Camera_Scan3dAxisMin_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera* *) –

Scan3dCoordinateOffset
    Camera_Scan3dCoordinateOffset_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera* *) –

Scan3dCoordinateReferenceSelector
    Camera_Scan3dCoordinateReferenceSelector_get(self) -> IEnumerationT_Scan3dCoordinateReferenceSelectorEnums

        Parameters self (*Spinnaker::Camera* *) –

Scan3dCoordinateReferenceValue
    Camera_Scan3dCoordinateReferenceValue_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera* *) –

Scan3dCoordinateScale
    Camera_Scan3dCoordinateScale_get(self) -> IFloat

> **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dCoordinateSelector
> Camera_Scan3dCoordinateSelector_get(self) -> IEnumerationT_Scan3dCoordinateSelectorEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dCoordinateSystem
> Camera_Scan3dCoordinateSystem_get(self) -> IEnumerationT_Scan3dCoordinateSystemEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dCoordinateSystemReference
> Camera_Scan3dCoordinateSystemReference_get(self) -> IEnumerationT_Scan3dCoordinateSystemReferenceEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dCoordinateTransformSelector
> Camera_Scan3dCoordinateTransformSelector_get(self) -> IEnumerationT_Scan3dCoordinateTransformSelectorEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dDistanceUnit
> Camera_Scan3dDistanceUnit_get(self) -> IEnumerationT_Scan3dDistanceUnitEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dInvalidDataFlag
> Camera_Scan3dInvalidDataFlag_get(self) -> IBoolean
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dInvalidDataValue
> Camera_Scan3dInvalidDataValue_get(self) -> IFloat
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dOutputMode
> Camera_Scan3dOutputMode_get(self) -> IEnumerationT_Scan3dOutputModeEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

Scan3dTransformValue
> Camera_Scan3dTransformValue_get(self) -> IFloat
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

SensorDescription
> Camera_SensorDescription_get(self) -> IString
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

SensorDigitizationTaps
> Camera_SensorDigitizationTaps_get(self) -> IEnumerationT_SensorDigitizationTapsEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

SensorHeight
> Camera_SensorHeight_get(self) -> IInteger
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

SensorShutterMode
> Camera_SensorShutterMode_get(self) -> IEnumerationT_SensorShutterModeEnums
>
> > **Parameters** **self** (*Spinnaker::Camera* *) –

SensorTaps
Camera_SensorTaps_get(self) -> IEnumerationT_SensorTapsEnums

> **Parameters self** (*Spinnaker::Camera **) –

SensorWidth
Camera_SensorWidth_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationMode
Camera_SequencerConfigurationMode_get(self) -> IEnumerationT_SequencerConfigurationModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationReset
Camera_SequencerConfigurationReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationValid
Camera_SequencerConfigurationValid_get(self) -> IEnumerationT_SequencerConfigurationValidEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerFeatureEnable
Camera_SequencerFeatureEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

SequencerFeatureSelector
Camera_SequencerFeatureSelector_get(self) -> IEnumerationT_SequencerFeatureSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerMode
Camera_SequencerMode_get(self) -> IEnumerationT_SequencerModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerPathSelector
Camera_SequencerPathSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetActive
Camera_SequencerSetActive_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetLoad
Camera_SequencerSetLoad_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetNext
Camera_SequencerSetNext_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetSave
Camera_SequencerSetSave_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetSelector
Camera_SequencerSetSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetStart
Camera_SequencerSetStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetValid
Camera_SequencerSetValid_get(self) -> IEnumerationT_SequencerSetValidEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerTriggerActivation
Camera_SequencerTriggerActivation_get(self) -> IEnumerationT_SequencerTriggerActivationEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerTriggerSource
Camera_SequencerTriggerSource_get(self) -> IEnumerationT_SequencerTriggerSourceEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortBaudRate
Camera_SerialPortBaudRate_get(self) -> IEnumerationT_SerialPortBaudRateEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortDataBits
Camera_SerialPortDataBits_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortParity
Camera_SerialPortParity_get(self) -> IEnumerationT_SerialPortParityEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortSelector
Camera_SerialPortSelector_get(self) -> IEnumerationT_SerialPortSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortSource
Camera_SerialPortSource_get(self) -> IEnumerationT_SerialPortSourceEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialPortStopBits
Camera_SerialPortStopBits_get(self) -> IEnumerationT_SerialPortStopBitsEnums

> **Parameters self** (*Spinnaker::Camera **) –

SerialReceiveFramingErrorCount
Camera_SerialReceiveFramingErrorCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SerialReceiveParityErrorCount
Camera_SerialReceiveParityErrorCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SerialReceiveQueueClear
Camera_SerialReceiveQueueClear_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SerialReceiveQueueCurrentCharacterCount
Camera_SerialReceiveQueueCurrentCharacterCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

SerialReceiveQueueMaxCharacterCount
Camera_SerialReceiveQueueMaxCharacterCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

SerialTransmitQueueCurrentCharacterCount
Camera_SerialTransmitQueueCurrentCharacterCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

SerialTransmitQueueMaxCharacterCount
Camera_SerialTransmitQueueMaxCharacterCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

Sharpening
Camera_Sharpening_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

SharpeningAuto
Camera_SharpeningAuto_get(self) -> IBoolean

    Parameters self (*Spinnaker::Camera **) –

SharpeningEnable
Camera_SharpeningEnable_get(self) -> IBoolean

    Parameters self (*Spinnaker::Camera **) –

SharpeningThreshold
Camera_SharpeningThreshold_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

SoftwareSignalPulse
Camera_SoftwareSignalPulse_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

SoftwareSignalSelector
Camera_SoftwareSignalSelector_get(self) -> IEnumerationT_SoftwareSignalSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

SourceCount
Camera_SourceCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

SourceSelector
Camera_SourceSelector_get(self) -> IEnumerationT_SourceSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

TLParamsLocked
Camera_TLParamsLocked_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

Test0001
Camera_Test0001_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

TestEventGenerate
    Camera_TestEventGenerate_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

TestPattern
    Camera_TestPattern_get(self) -> IEnumerationT_TestPatternEnums

> Parameters **self** (*Spinnaker::Camera **) –

TestPatternGeneratorSelector
    Camera_TestPatternGeneratorSelector_get(self) -> IEnumerationT_TestPatternGeneratorSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

TestPendingAck
    Camera_TestPendingAck_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

TimerDelay
    Camera_TimerDelay_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

TimerDuration
    Camera_TimerDuration_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

TimerReset
    Camera_TimerReset_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera **) –

TimerSelector
    Camera_TimerSelector_get(self) -> IEnumerationT_TimerSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

TimerStatus
    Camera_TimerStatus_get(self) -> IEnumerationT_TimerStatusEnums

> Parameters **self** (*Spinnaker::Camera **) –

TimerTriggerActivation
    Camera_TimerTriggerActivation_get(self) -> IEnumerationT_TimerTriggerActivationEnums

> Parameters **self** (*Spinnaker::Camera **) –

TimerTriggerSource
    Camera_TimerTriggerSource_get(self) -> IEnumerationT_TimerTriggerSourceEnums

> Parameters **self** (*Spinnaker::Camera **) –

TimerValue
    Camera_TimerValue_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

Timestamp
    Camera_Timestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

TimestampIncrement
Camera_TimestampIncrement_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TimestampLatch
Camera_TimestampLatch_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TimestampLatchValue
Camera_TimestampLatchValue_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TimestampReset
Camera_TimestampReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferAbort
Camera_TransferAbort_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferBlockCount
Camera_TransferBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferBurstCount
Camera_TransferBurstCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferComponentSelector
Camera_TransferComponentSelector_get(self) -> IEnumerationT_TransferComponentSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferControlMode
Camera_TransferControlMode_get(self) -> IEnumerationT_TransferControlModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferOperationMode
Camera_TransferOperationMode_get(self) -> IEnumerationT_TransferOperationModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferPause
Camera_TransferPause_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueCurrentBlockCount
Camera_TransferQueueCurrentBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueMaxBlockCount
Camera_TransferQueueMaxBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueMode
Camera_TransferQueueMode_get(self) -> IEnumerationT_TransferQueueModeEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferQueueOverflowCount
Camera_TransferQueueOverflowCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

TransferResume
Camera_TransferResume_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera **) –

TransferSelector
Camera_TransferSelector_get(self) -> IEnumerationT_TransferSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferStart
Camera_TransferStart_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera **) –

TransferStatus
Camera_TransferStatus_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera **) –

TransferStatusSelector
Camera_TransferStatusSelector_get(self) -> IEnumerationT_TransferStatusSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferStop
Camera_TransferStop_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera **) –

TransferStreamChannel
Camera_TransferStreamChannel_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerActivation
Camera_TransferTriggerActivation_get(self) -> IEnumerationT_TransferTriggerActivationEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerMode
Camera_TransferTriggerMode_get(self) -> IEnumerationT_TransferTriggerModeEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerSelector
Camera_TransferTriggerSelector_get(self) -> IEnumerationT_TransferTriggerSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerSource
Camera_TransferTriggerSource_get(self) -> IEnumerationT_TransferTriggerSourceEnums

Parameters **self** (*Spinnaker::Camera **) –

TriggerActivation
Camera_TriggerActivation_get(self) -> IEnumerationT_TriggerActivationEnums

Parameters **self** (*Spinnaker::Camera **) –

TriggerDelay
    Camera_TriggerDelay_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

TriggerDivider
    Camera_TriggerDivider_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

TriggerEventTest
    Camera_TriggerEventTest_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

TriggerMode
    Camera_TriggerMode_get(self) -> IEnumerationT_TriggerModeEnums

        Parameters self (*Spinnaker::Camera **) –

TriggerMultiplier
    Camera_TriggerMultiplier_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

TriggerOverlap
    Camera_TriggerOverlap_get(self) -> IEnumerationT_TriggerOverlapEnums

        Parameters self (*Spinnaker::Camera **) –

TriggerSelector
    Camera_TriggerSelector_get(self) -> IEnumerationT_TriggerSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

TriggerSoftware
    Camera_TriggerSoftware_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

TriggerSource
    Camera_TriggerSource_get(self) -> IEnumerationT_TriggerSourceEnums

        Parameters self (*Spinnaker::Camera **) –

U3VAccessPrivilege
    Camera_U3VAccessPrivilege_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

U3VCPCapability
    Camera_U3VCPCapability_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

U3VCPEIRMAvailable
    Camera_U3VCPEIRMAvailable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

U3VCPIIDC2Available
    Camera_U3VCPIIDC2Available_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

U3VCPSIRMAvailable
    Camera_U3VCPSIRMAvailable_get(self) -> IBoolean

> > > **Parameters self** (*Spinnaker::Camera* *) –

U3VCurrentSpeed
> Camera_U3VCurrentSpeed_get(self) -> IEnumerationT_U3VCurrentSpeedEnums
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VMaxAcknowledgeTransferLength
> Camera_U3VMaxAcknowledgeTransferLength_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VMaxCommandTransferLength
> Camera_U3VMaxCommandTransferLength_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VMaxDeviceResponseTime
> Camera_U3VMaxDeviceResponseTime_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VMessageChannelID
> Camera_U3VMessageChannelID_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VNumberOfStreamChannels
> Camera_U3VNumberOfStreamChannels_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VVersionMajor
> Camera_U3VVersionMajor_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

U3VVersionMinor
> Camera_U3VVersionMinor_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserOutputSelector
> Camera_UserOutputSelector_get(self) -> IEnumerationT_UserOutputSelectorEnums
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserOutputValue
> Camera_UserOutputValue_get(self) -> IBoolean
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserOutputValueAll
> Camera_UserOutputValueAll_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserOutputValueAllMask
> Camera_UserOutputValueAllMask_get(self) -> IInteger
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserSetDefault
> Camera_UserSetDefault_get(self) -> IEnumerationT_UserSetDefaultEnums
>
> > **Parameters self** (*Spinnaker::Camera* *) –

UserSetFeatureEnable
Camera_UserSetFeatureEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

UserSetFeatureSelector
Camera_UserSetFeatureSelector_get(self) -> IEnumerationT_UserSetFeatureSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

UserSetLoad
Camera_UserSetLoad_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

UserSetSave
Camera_UserSetSave_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

UserSetSelector
Camera_UserSetSelector_get(self) -> IEnumerationT_UserSetSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

V3_3Enable
Camera_V3_3Enable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

WhiteClip
Camera_WhiteClip_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

WhiteClipSelector
Camera_WhiteClipSelector_get(self) -> IEnumerationT_WhiteClipSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

Width
Camera_Width_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

WidthMax
Camera_WidthMax_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

aPAUSEMACCtrlFramesReceived
Camera_aPAUSEMACCtrlFramesReceived_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

aPAUSEMACCtrlFramesTransmitted
Camera_aPAUSEMACCtrlFramesTransmitted_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

thisown
The membership flag

4.4 PySpin.CameraBase

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

> The base class for the camera object.
>
> C++ includes: CameraBase.h
>
> **BeginAcquisition**(*self*)
>
> > **Parameters self** (*Spinnaker::CameraBase **) –
> >
> > 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*)
>
> > **Parameters self** (*Spinnaker::CameraBase **) –
> >
> > 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
>
> > **Parameters self** (*Spinnaker::CameraBase **) –
> >
> > 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*)
>
> > **Parameters self** (*Spinnaker::CameraBase **) –
> >
> > 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()
>
> **GetAccessMode**(*self*) → Spinnaker::GenApi::EAccessMode
>
> > **Parameters self** (*Spinnaker::CameraBase const **) –

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

GetGuiXml (*self*) → gcstring

> **Parameters self** (*Spinnaker::CameraBase const **) –

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*, *streamID=0*) → ImagePtr

> **Parameters**
>
> - **grabTimeout** (*uint64_t*) –
> - **streamID** (*uint64_t*) –
> - **grabTimeout) -> ImagePtr** (GetNextImage *(self,)*) –
> - **grabTimeout** –
> - **-> ImagePtr** (GetNextImage *(self))* –
> - **self** (*Spinnaker::CameraBase **) –

ImagePtr Spinnaker::CameraBase::GetNextImage(uint64_t grabTimeout=EVENT_TIMEOUT_INFINITE, uint64_t streamID=0)

Gets the next image that was received by the transport layer. This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID to select from which stream to grab images

See: Init()

See: BeginAcquisition()

See: EndAcquisition()

grabTimeout: a 64bit value that represents a timeout in milliseconds

streamID: The stream to grab the image.

pointer to an Image object

GetNodeMap (*self*) → INodeMap

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

unsigned int Spinnaker::CameraBase::GetNumDataStreams()

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse (*self*) → unsigned int

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase const **) –

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*) → INodeMap

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

Gets a reference to the node map that is generated from a GenICam XML file for the GenTL Stream module. The camera does not need to be initialized before acquiring this node map.

A reference to the INodeMap.

GetUniqueID (*self*) → gcstring

> **Parameters self** (*Spinnaker::CameraBase **) –

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)

Init (*self*)

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

bool Spinnaker::CameraBase::IsValid()

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

If camera is valid or not

RegisterEvent (*self*, *evtToRegister*)

> **Parameters**
>
> > - **evtToRegister** (*Spinnaker::Event &*) –
> > - **evtToRegister, eventName)** ([RegisterEvent](self,)) –
> > - **evtToRegister** –
> > - **eventName** (*Spinnaker::GenICam::gcstring const &*) –

void Spinnaker::CameraBase::RegisterEvent(Event &evtToRegister, const GenICam::gcstring &eventName)

Registers a specific event for the camera

See: Init()

evtToRegister: The event to register for the camera

eventName: The event name to register

TLDevice
CameraBase_TLDevice_get(self) -> TransportLayerDevice

> **Parameters self** (*Spinnaker::CameraBase **) –

TLStream
CameraBase_TLStream_get(self) -> TransportLayerStream

> **Parameters self** (*Spinnaker::CameraBase **) –

UnregisterEvent (*self*, *evtToUnregister*)

> **Parameters evtToUnregister** (*Spinnaker::Event &*) –

void Spinnaker::CameraBase::UnregisterEvent(Event &evtToUnregister)

Unregisters an event for the camera Events should be unregistered first before calling camera DeInit(). Otherwise an exception will be thrown in the DeInit() call and require the user to unregister events before the camera can be re-initialized again.

See: DeInit()

evtToUnregister: The event to unregister from the camera

thisown
   The membership flag

4.5 PySpin.CameraDefs

4.6 PySpin.CameraList

class PySpin.**CameraList**(**args*)
   Used to hold a list of camera objects.

   C++ includes: CameraList.h

   Append(*self*, *otherList*)

         Parameters otherList (*Spinnaker::CameraList &*) –

      void Spinnaker::CameraList::Append(CameraList &otherList)

      Appends a camera list to the current list.

      otherList: The other list to append to this list

   Clear(*self*)

         Parameters self (*Spinnaker::CameraList **) –

      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()

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

         Parameters index (*int*) –

      CameraPtr Spinnaker::CameraList::GetByIndex(int index) const

      Returns a pointer to a camera object at the "index".

      index: The index at which to retrieve the camera object

      A pointer to an camera object.

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

         Parameters serialNumber (*std::string*) –

CameraPtr Spinnaker::CameraList::GetBySerial(std::string serialNumber) const

Returns a pointer to a camera object with the specified serial number.

serialNumber: The serial number of the camera object to retrieve

A pointer to an camera object.

GetSize(*self*) → int

> **Parameters self** (*Spinnaker::CameraList const **) –

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.

RemoveByIndex(*self*, *index*)

> **Parameters index** (*int*) –

void Spinnaker::CameraList::RemoveByIndex(int index)

Removes a camera at "index" and destroys its corresponding reference counted object.

index: The index at which to remove the Camera object

RemoveBySerial(*self*, *serialNumber*)

> **Parameters serialNumber** (*std::string*) –

void Spinnaker::CameraList::RemoveBySerial(std::string serialNumber)

Removes a camera using its serial number and destroys its corresponding reference counted object.

serialNumber: The serial number of the Camera object to remove

thisown
> The membership flag

4.7 PySpin.CameraPtr

class PySpin.**CameraPtr**(**args*)
> A reference tracked pointer to a camera object.

C++ includes: CameraPtr.h

thisown
> The membership flag

4.8 PySpin.ChunkData

class PySpin.**ChunkData**(**args*)
> The chunk data which contains additional information about an image.

C++ includes: ChunkData.h

GetBlackLevel(*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetCounterValue (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetCounterValue() const

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

GetEncoderValue (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetExposureTime (*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetExposureTime() const

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

GetFrameID (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetGain() const

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

GetHeight (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetHeight() const

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

GetImage (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetLinePitch (*self*) → int64_t

> Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetLinePitch() const

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

GetLineStatusAll (*self*) → int64_t

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

GetOffsetX (*self*) → int64_t

> Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetOffsetX() const

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

GetOffsetY (*self*) → int64_t

> Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetOffsetY() const

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

GetPartSelector (*self*) → int64_t

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

> Parameters **self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

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

GetScan3dInvalidDataValue(*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const

Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetSequencerSetActive() const

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

GetSerialDataLength(*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetStreamChannelID(*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetStreamChannelID() const

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

GetTimerValue (*self*) → float64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const

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

GetTransferBlockID (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const

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

GetWidth (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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 Spinnaker::ChunkData::SetChunks(GenApi::INodeMap &pNodeMap)

thisown
The membership flag

4.9 PySpin.Exception

4.10 PySpin.Image

class PySpin.**Image**(**args*, ***kwargs*)
 The image object class.

 C++ includes: Image.h

 CalculateChannelStatistics(*self*, *channel*) → ChannelStatistics

 Parameters channel (*enum Spinnaker::StatisticsChannel*) –

 Returns a ChannelStatistics instance for the current image on a given channel.

 channel: Channel to generate statistics on.

 CheckCRC(*self*) → bool

 Parameters self (*Spinnaker::Image const **) –

 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.

 Convert(*self*, *format*, *algorithm*) → ImagePtr

 Parameters

 • **format** (*enum Spinnaker::PixelFormatEnums*) –

 • **algorithm** (*enum Spinnaker::ColorProcessingAlgorithm*) –

 • **format) -> ImagePtr** (Convert(*self*,) –

 • **format** –

 ImagePtr Spinnaker::Image::Convert(Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm algorithm=DEFAULT) const

 Converts the current image buffer to the specified output pixel format and stores the result in the specified image. The destination image does not need to be configured in any way before the call is made.

 See: PixelFormatEnums

 format: Output format of the converted image.

 algorithm: processing algorithm for producing the converted image

 The converted image.

 static Create() → ImagePtr
 Create(image) -> ImagePtr

 Parameters

 • **image** (*Spinnaker::ImagePtr const*) –

 • **height, offsetX, offsetY, pixelFormat, pData) -> ImagePtr** (Create(*width*,) –

 • **width** (*size_t*) –

 • **height** (*size_t*) –

- **offsetX** (*size_t*) –

- **offsetY** (*size_t*) –

- **pixelFormat** (*enum Spinnaker::PixelFormatEnums*) –

- **pData** (*void **) –

Creates a new Image object, either using a default constructor, copied from another ImagePtr, or using width, height, offset_x, offset_y, pixel format, and a NumPy array containing 8-bit unsigned ints representing the image data (replaces the void* pData argument).

DeepCopy (*self*, *pSrcImage*)

> **Parameters pSrcImage** (*Spinnaker::ImagePtr const*) –

void Spinnaker::Image::DeepCopy(const ImagePtr pSrcImage)

Performs a deep copy of the Image. After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The Image's current buffer will not be released.

pSrcImage: The Image to copy the data from.

GetBitsPerPixel (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

Returns a pointer to a chunk data interface. No ownership is transfered, 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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

static GetDefaultColorProcessing () → Spinnaker::ColorProcessingAlgorithm

GetFrameID (*self*) → uint64_t

> **Parameters self** (*Spinnaker::Image const **) –

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

uint64_t Spinnaker::Image::GetID() const

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

The 64 bit unique id for this image.

GetImageSize (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus (*self*) → Spinnaker::ImageStatus

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

GetPayloadType (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

GetPixelFormatName (*self*) → gcstring

> **Parameters self** (*Spinnaker::Image const **) –

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 *

> **Parameters self** (*Spinnaker::Image const **) –

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.

GetStride (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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::PayloadTypeInfoIDs

Parameters **self** (*Spinnaker::Image const **) –

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

Parameters **self** (*Spinnaker::Image const **) –

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::PixelFormatNamespaceID

Parameters **self** (*Spinnaker::Image const **) –

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

Parameters **self** (*Spinnaker::Image const **) –

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

Parameters **self** (*Spinnaker::Image const **) –

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

Parameters **self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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.

IsInUse(*self*) → bool

> **Parameters self** (*Spinnaker::Image **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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.

Release(*self*)

> **Parameters self** (*Spinnaker::Image **) –

void Spinnaker::Image::Release()

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

> **Parameters**
>
> - **width** (*size_t*) –
> - **height** (*size_t*) –
> - **offsetX** (*size_t*) –
> - **offsetY** (*size_t*) –
> - **pixelFormat** (*enum Spinnaker::PixelFormatEnums*) –
> - **width, height, offsetX, offsetY, pixelFormat, pData)** (*ResetImage(self,*) –
> - **width** –
> - **height** –
> - **offsetX** –
> - **offsetY** –
> - **pixelFormat** –
> - **pData** (*void **) –

void Spinnaker::Image::ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData)

Sets new dimensions of the image object.

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.

pData: Pointer to the image buffer.

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

> **Parameters**
> - **pFilename** (*char const* *) –
> - **format** (*enum Spinnaker::ImageFileFormat*) –
> - **pFilename)** (Save(self,) –
> - **pFilename** –
> - **pFilename, pOption)** (Save(self,) –
> - **pFilename** –
> - **pOption** (*Spinnaker::BMPOption &*) –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –

void Spinnaker::Image::Save(const char *pFilename, BMPOption &pOption)

Saves the image to the specified file name with the options specified.

pFilename: Filename to save image with.

pOption: Options to use while saving image.

static SetDefaultColorProcessing(*defaultMethod*)

> **Parameters defaultMethod** (*enum Spinnaker::ColorProcessingAlgorithm*) –

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
>
> **`thisown`**
> > The membership flag

4.12 PySpin.ImageStatistics

4.13 PySpin.Interface

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

> An interface object which holds a list of cameras.
>
> C++ includes: Interface.h
>
> **`GetCameras`**(*self*, *updateCameras=True*) → CameraList
>
> > **Parameters**
> >
> > > - **`updateCameras`** (`bool`) –
> > > - **`-> CameraList`** (`GetCameras(self)`) –
> > > - **`self`** (`Spinnaker::Interface const *`) –
> >
> > CameraList Spinnaker::Interface::GetCameras(bool updateCameras=true) const
> >
> > Returns a list of cameras available on this interface. This call returns either usb3 vision or gige vision cameras depending on the underlying transport layer of this interface. The camera list object will reference count the cameras that it holds. It is important that the CameraList is destroyed or is cleared before System::ReleaseInstance() can be called or an InterfaceList that holds this interface can be cleared.
> >
> > See: System::ReleaseInstance()
> >
> > See: InterfaceList::Clear()
> >
> > See: CameraList::Clear()
> >
> > updateCameras: A flag used to issue an updateCameras() call internally before getting the camera list
> >
> > An CameraList object that contains a list of cameras on this interface.
>
> **`GetTLNodeMap`**(*self*) → INodeMap
>
> > **Parameters self** (`Spinnaker::Interface const *`) –
> >
> > GenApi::INodeMap& Spinnaker::Interface::GetTLNodeMap() const
> >
> > Gets a nodeMap that is generated from a GenICam XML file for the GenTL interface Module.
> >
> > A reference to a INodeMap object.
>
> **`IsInUse`**(*self*) → bool
>
> > **Parameters self** (`Spinnaker::Interface const *`) –

bool Spinnaker::Interface::IsInUse() const

Checks if the interface is in use by any camera objects

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

RegisterEvent(*self*, *evtToRegister*)

> Parameters **evtToRegister** (*Spinnaker::Event &*) –

void Spinnaker::Interface::RegisterEvent(Event &evtToRegister)

Registers an event for the interface

evtToRegister: The event to register for the interface

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

> **Parameters**
>
> - **deviceKey** (*unsigned int*) –
> - **groupKey** (*unsigned int*) –
> - **groupMask** (*unsigned int*) –
> - **actionTime** (*unsigned long long*) –
> - **pResultSize** (*unsigned int **) –
> - **results** (*Spinnaker::ActionCommandResult []*) –
> - **deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **pResultSize** –
> - **deviceKey, groupKey, groupMask, actionTime=0)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **deviceKey, groupKey, groupMask)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –

void Spinnaker::Interface::SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL) const

Broadcast an Action Command to all devices on interface

---

deviceKey: The Action Command's device key

groupKey: The Action Command's group key

groupMask: The Action Command's group mask

actionTime: (Optional) Time when to assert a future action. Zero means immediate action.

pResultSize: (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.

results: (Optional) An Array with *****pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

TLInterface
Interface_TLInterface_get(self) -> TransportLayerInterface

>   **Parameters self** (*Spinnaker::Interface **) –

UnregisterEvent (*self*, *evtToUnregister*)

>   **Parameters evtToUnregister** (*Spinnaker::Event &*) –

void Spinnaker::Interface::UnregisterEvent(Event &evtToUnregister)

Unregisters an event for the interface

evtToUnregister: The event to unregister from the interface

UpdateCameras (*self*) → bool

>   **Parameters self** (*Spinnaker::Interface **) –

bool Spinnaker::Interface::UpdateCameras()

Updates the list of cameras on this interface. This function needs to be called before any cameras can be discovered using GetCameras(). System::GetCameras() will automatically call this function for each interface it enumerates. If the list changed after the last time System::GetCameras() or UpdateCameras() was called then the return value will be true, otherwise it is false.

See: System::GetCameras()

See: GetCameras()

true if cameras changed on interface and false otherwise.

thisown
The membership flag

4.14 PySpin.InterfaceList

class PySpin.**InterfaceList**(**args*)
A list of the available interfaces on the system.

C++ includes: InterfaceList.h

Clear (*self*)

>   **Parameters self** (*Spinnaker::InterfaceList **) –

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**(*int*) –

InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index) const

Returns a pointer to an Interface object at the "index".

index: The index at which to retrieve the Interface object

A pointer to an Interface object.

GetSize(*self*) → int

>   **Parameters self**(*Spinnaker::InterfaceList const **) –

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.

thisown
>   The membership flag

4.15 PySpin.InterfacePtr

class PySpin.**InterfacePtr**(**args*)
>   A reference tracked pointer to the interface object.

>   C++ includes: InterfacePtr.h

>   **thisown**
>   >   The membership flag

4.16 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**(*bool*) –
>   >   • **updateCameras**(*bool*) –
>   >   • **updateInterfaces=True) -> CameraList**(GetCameras*(self,)* –
>   >   • **updateInterfaces** –
>   >   • **-> CameraList**(GetCameras*(self))* –
>   >   • **self**(*Spinnaker::System **) –

CameraList Spinnaker::System::GetCameras(bool updateInterfaces=true, bool updateCameras=true)

Returns a list of cameras that are available on the system. This call returns both GigE Vision and Usb3 Vision cameras from all interfaces. The camera list object will reference count the cameras it returns. It is important that the camera list is destroyed or is cleared before calling system-> ReleaseInstance() or else the call to system-> ReleaseInstance() will result in an error message thrown that a reference to the camera is still held.

See: ReleaseInstance()

See: CameraList::Clear()

updateInterfaces: Determines whether or not updateInterfaceList() is called before getting cameras from available interfaces on the system

updateCameras: Determines whether or not UpdateCameras() is called before getting cameras from available interfaces on the system

An CameraList object that contains a list of all cameras.

static GetInstance() → SystemPtr

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

> **Parameters**
>
> > - **updateInterface**(*bool*) –
> > - **-> InterfaceList**(*GetInterfaces(self)*) –
> > - **self**(*Spinnaker::System **) –

InterfaceList Spinnaker::System::GetInterfaces(bool updateInterface=true)

Returns a list of interfaces available on the system. This call returns GigE and Usb2 and Usb3 interfaces.

updateInterface: Determines whether or not UpdateInterfaceList() is called before getting available interfaces

An InterfaceList object that contains a list of all interfaces.

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

> **Parameters self**(*Spinnaker::System **) –

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

IsInUse(*self*) → bool

> **Parameters self**(*Spinnaker::System **) –

---

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.

RegisterInterfaceEvent (*self*, *evtToRegister*, *updateInterface=True*)

> **Parameters**
>
> - **evtToRegister** (*Spinnaker::Event &*) –
> - **updateInterface** (*bool*) –
> - **evtToRegister)** ([RegisterInterfaceEvent](self,)) –
> - **evtToRegister** –

void Spinnaker::System::RegisterInterfaceEvent(Event &evtToRegister, bool updateInterface=true)

Registers events for all available interfaces that are found on the system

evtToRegister: The event to register for the available interfaces

updateInterface: Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

RegisterLoggingEvent (*self*, *handler*)

> **Parameters handler** (*Spinnaker::LoggingEvent &*) –

void Spinnaker::System::RegisterLoggingEvent(LoggingEvent &handler)

Registers a logging event.

handler: The logging event handler to register

ReleaseInstance (*self*)

> **Parameters self** (*Spinnaker::System **) –

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*, *pResultSize=None*, *results=0*)

> **Parameters**
>
> - **deviceKey** (*unsigned int*) –
> - **groupKey** (*unsigned int*) –
> - **groupMask** (*unsigned int*) –
> - **actionTime** (*unsigned long long*) –
> - **pResultSize** (*unsigned int **) –
> - **results** (*Spinnaker::ActionCommandResult []*) –
> - **deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None)** ([SendActionCommand](self,)) –

- **deviceKey** –

- **groupKey** –

- **groupMask** –

- **actionTime** –

- **pResultSize** –

- **deviceKey, groupKey, groupMask, actionTime=0)**
  (SendActionCommand(self,) –

- **deviceKey** –

- **groupKey** –

- **groupMask** –

- **actionTime** –

- **deviceKey, groupKey, groupMask)** (SendActionCommand(self,) –

- **deviceKey** –

- **groupKey** –

- **groupMask** –

void Spinnaker::System::SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL)

Broadcast an Action Command to all devices on system

deviceKey: The Action Command's device key

groupKey: The Action Command's group key

groupMask: The Action Command's group mask

actionTime: (Optional) Time when to assert a future action. Zero means immediate action.

pResultSize: (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.

results: (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

SetLoggingEventPriorityLevel(*self*, *level*)

> **Parameters level** (*enum Spinnaker::SpinnakerLogLevel*) –

void Spinnaker::System::SetLoggingEventPriorityLevel(SpinnakerLogLevel level)

Sets a threshold priority level for logging event. Logging events below such level will not trigger callbacks.

Spinnaker uses five levels of logging: Error - failures that are non- recoverable without user intervention.

Warning - failures that are recoverable without user intervention.

Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.

Info - information about recurring events that are generated regularly such as information on individual images.

Debug - information that can be used to troubleshoot the system.

See: SpinnakerLogLevel

level: The threshold level

UnregisterAllLoggingEvent(*self*)

> **Parameters self** (*Spinnaker::System **) –

void Spinnaker::System::UnregisterAllLoggingEvent()

Unregisters all previously registered logging events.

UnregisterInterfaceEvent(*self*, *evtToUnregister*)

> **Parameters evtToUnregister** (*Spinnaker::Event &*) –

void Spinnaker::System::UnregisterInterfaceEvent(Event &evtToUnregister)

Unregisters events for all available interfaces that are found on the system

evtToUnregister: The event to unregister from the available interfaces

UnregisterLoggingEvent(*self*, *handler*)

> **Parameters handler** (*Spinnaker::LoggingEvent &*) –

void Spinnaker::System::UnregisterLoggingEvent(LoggingEvent &handler)

Unregisters a logging event.

handler: The logging event handler to unregister

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

> **Parameters**
>
> - **updateInterfaces** (*bool*) –
> - **-> bool** (*UpdateCameras(self)*) –
> - **self** (*Spinnaker::System **) –

bool Spinnaker::System::UpdateCameras(bool updateInterfaces=true)

Updates the list of cameras on the system. Note that System::GetCameras() internally calls UpdateCameras() for each interface it enumerates. If the list changed between this call and the last time UpdateCameras was called then the return value will be true, otherwise it is false.

See: GetCameras()

updateInterfaces: Determines whether or not UpdateInterfaceList() is called before updating cameras for available interfaces on the system

True if cameras changed on interface and false otherwise.

thisown
The membership flag

4.17 PySpin.SystemPtr

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

    A reference tracked pointer to a system object.

    C++ includes: SystemPtr.h

    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

DeviceAccessStatus

TransportLayerDevice_DeviceAccessStatus_get(self) -> IEnumerationT_DeviceAccessStatusEnum

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceCurrentSpeed

TransportLayerDevice_DeviceCurrentSpeed_get(self) -> IEnumerationT_DeviceCurrentSpeedEnum

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceDisplayName

TransportLayerDevice_DeviceDisplayName_get(self) -> IString

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceDriverVersion

TransportLayerDevice_DeviceDriverVersion_get(self) -> IString

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceEndianessMechanism

TransportLayerDevice_DeviceEndianessMechanism_get(self) -> IEnumerationT_DeviceEndianessMechanismEnum

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceID

TransportLayerDevice_DeviceID_get(self) -> IString

Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceInstanceId
TransportLayerDevice_DeviceInstanceId_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceLinkSpeed
TransportLayerDevice_DeviceLinkSpeed_get(self) -> IInteger

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceModelName
TransportLayerDevice_DeviceModelName_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceMulticastMonitorMode
TransportLayerDevice_DeviceMulticastMonitorMode_get(self) -> IBoolean

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceSerialNumber
TransportLayerDevice_DeviceSerialNumber_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceType
TransportLayerDevice_DeviceType_get(self) -> IEnumerationT_DeviceTypeEnum

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceUserID
TransportLayerDevice_DeviceUserID_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceVendorName
TransportLayerDevice_DeviceVendorName_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

DeviceVersion
TransportLayerDevice_DeviceVersion_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

GUIXMLLocation
TransportLayerDevice_GUIXMLLocation_get(self) -> IEnumerationT_GUIXMLLocationEnum

    Parameters self (*Spinnaker::TransportLayerDevice **) –

GUIXMLPath
TransportLayerDevice_GUIXMLPath_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

GenICamXMLLocation
TransportLayerDevice_GenICamXMLLocation_get(self) -> IEnumerationT_GenICamXMLLocationEnum

    Parameters self (*Spinnaker::TransportLayerDevice **) –

GenICamXMLPath
TransportLayerDevice_GenICamXMLPath_get(self) -> IString

    Parameters self (*Spinnaker::TransportLayerDevice **) –

GevCCP
>   TransportLayerDevice_GevCCP_get(self) -> IEnumerationT_GevCCPEnum

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceDiscoverMaximumPacketSize
>   TransportLayerDevice_GevDeviceDiscoverMaximumPacketSize_get(self) -> ICommand

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceGateway
>   TransportLayerDevice_GevDeviceGateway_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceIPAddress
>   TransportLayerDevice_GevDeviceIPAddress_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceMACAddress
>   TransportLayerDevice_GevDeviceMACAddress_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceMaximumPacketSize
>   TransportLayerDevice_GevDeviceMaximumPacketSize_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceMaximumRetryCount
>   TransportLayerDevice_GevDeviceMaximumRetryCount_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceModeIsBigEndian
>   TransportLayerDevice_GevDeviceModeIsBigEndian_get(self) -> IBoolean

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDevicePort
>   TransportLayerDevice_GevDevicePort_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceReadAndWriteTimeout
>   TransportLayerDevice_GevDeviceReadAndWriteTimeout_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevDeviceSubnetMask
>   TransportLayerDevice_GevDeviceSubnetMask_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevVersionMajor
>   TransportLayerDevice_GevVersionMajor_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevVersionMinor
>   TransportLayerDevice_GevVersionMinor_get(self) -> IInteger

>>   **Parameters self** (*Spinnaker::TransportLayerDevice **) –

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

ActionCommand

TransportLayerInterface_ActionCommand_get(self) -> ICommand

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

AutoForceIP

TransportLayerInterface_AutoForceIP_get(self) -> ICommand

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceAccessStatus

TransportLayerInterface_DeviceAccessStatus_get(self) -> IEnumerationT_DeviceAccessStatusEnum

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceCount

TransportLayerInterface_DeviceCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceID

TransportLayerInterface_DeviceID_get(self) -> IString

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceModelName

TransportLayerInterface_DeviceModelName_get(self) -> IString

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceSelector

TransportLayerInterface_DeviceSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceUnlock

TransportLayerInterface_DeviceUnlock_get(self) -> IString

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceUpdateList

TransportLayerInterface_DeviceUpdateList_get(self) -> ICommand

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

DeviceVendorName

TransportLayerInterface_DeviceVendorName_get(self) -> IString

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevActionDeviceKey

TransportLayerInterface_GevActionDeviceKey_get(self) -> IInteger

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevActionGroupKey

TransportLayerInterface_GevActionGroupKey_get(self) -> IInteger

> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevActionGroupMask
> TransportLayerInterface_GevActionGroupMask_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevActionTime
> TransportLayerInterface_GevActionTime_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevDeviceIPAddress
> TransportLayerInterface_GevDeviceIPAddress_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevDeviceMACAddress
> TransportLayerInterface_GevDeviceMACAddress_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevDeviceSubnetMask
> TransportLayerInterface_GevDeviceSubnetMask_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevInterfaceGateway
> TransportLayerInterface_GevInterfaceGateway_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevInterfaceIPAddress
> TransportLayerInterface_GevInterfaceIPAddress_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevInterfaceMACAddress
> TransportLayerInterface_GevInterfaceMACAddress_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

GevInterfaceSubnetMask
> TransportLayerInterface_GevInterfaceSubnetMask_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceCount
> TransportLayerInterface_IncompatibleDeviceCount_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceID
> TransportLayerInterface_IncompatibleDeviceID_get(self) -> IString

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceModelName
> TransportLayerInterface_IncompatibleDeviceModelName_get(self) -> IString

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceSelector
> TransportLayerInterface_IncompatibleDeviceSelector_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceVendorName
> TransportLayerInterface_IncompatibleDeviceVendorName_get(self) -> IString

---

> Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceDisplayName
> TransportLayerInterface_InterfaceDisplayName_get(self) -> IString

> > Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceID
> TransportLayerInterface_InterfaceID_get(self) -> IString

> > Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceType
> TransportLayerInterface_InterfaceType_get(self) -> IString

> > Parameters **self** (*Spinnaker::TransportLayerInterface **) –

POEStatus
> TransportLayerInterface_POEStatus_get(self) -> IEnumerationT_POEStatusEnum

> > Parameters **self** (*Spinnaker::TransportLayerInterface **) –

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

GevFailedPacketCount
> TransportLayerStream_GevFailedPacketCount_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevMaximumNumberResendBuffers
> TransportLayerStream_GevMaximumNumberResendBuffers_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevMaximumNumberResendRequests
> TransportLayerStream_GevMaximumNumberResendRequests_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevPacketResendMode
> TransportLayerStream_GevPacketResendMode_get(self) -> IBoolean

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevPacketResendTimeout
> TransportLayerStream_GevPacketResendTimeout_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevResendPacketCount
> TransportLayerStream_GevResendPacketCount_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevResendRequestCount
> TransportLayerStream_GevResendRequestCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

GevTotalPacketCount
TransportLayerStream_GevTotalPacketCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamBlockTransferSize
TransportLayerStream_StreamBlockTransferSize_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamBufferHandlingMode
TransportLayerStream_StreamBufferHandlingMode_get(self)   ->   IEnumera-
tionT_StreamBufferHandlingModeEnum

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamBufferUnderrunCount
TransportLayerStream_StreamBufferUnderrunCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamCRCCheckEnable
TransportLayerStream_StreamCRCCheckEnable_get(self) -> IBoolean

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamDefaultBufferCount
TransportLayerStream_StreamDefaultBufferCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamDefaultBufferCountMax
TransportLayerStream_StreamDefaultBufferCountMax_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamDefaultBufferCountMode
TransportLayerStream_StreamDefaultBufferCountMode_get(self)   ->   IEnumera-
tionT_StreamDefaultBufferCountModeEnum

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamFailedBufferCount
TransportLayerStream_StreamFailedBufferCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamID
TransportLayerStream_StreamID_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamTotalBufferCount
TransportLayerStream_StreamTotalBufferCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamType
TransportLayerStream_StreamType_get(self) -> IEnumerationT_StreamTypeEnum

Parameters **self** (*Spinnaker::TransportLayerStream* *) –

thisown
The membership flag

PYSPIN MODULE

class PySpin.**AVIOption**
    Bases: object

    Options for saving AVI files.

    C++ includes: SpinnakerDefs.h

    frameRate
        AVIOption_frameRate_get(self) -> float

            Parameters **self** (*Spinnaker::AVIOption **) –

    reserved
        AVIOption_reserved_get(self) -> unsigned int [256]

            Parameters **self** (*Spinnaker::AVIOption **) –

    thisown
        The membership flag

class PySpin.**AVIRecorder**
    Bases: object

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

    C++ includes: AVIRecorder.h

    AVIAppend(*self*, *pImage*)

            Parameters **pImage** (*Spinnaker::ImagePtr*) –

        virtual void Spinnaker::AVIRecorder::AVIAppend(ImagePtr pImage)

        Append an image to the AVI/MP4 file.

        pImage: The image to append.

    AVIClose(*self*)

            Parameters **self** (*Spinnaker::AVIRecorder **) –

        virtual void Spinnaker::AVIRecorder::AVIClose()

        Close the AVI/MP4 file.

        See: AVIOpen()

    AVIOpen(*self*, *pFileName*, *pOption*)

            Parameters

                • **pFileName** (*char const **) –

- **pOption** (*Spinnaker::H264Option &*) –

- **pFileName, pOption)** (AVIOpen(*self,*)) –

- **pFileName** –

- **pOption** –

- **pFileName, pOption)** –

- **pFileName** –

- **pOption** –

virtual void Spinnaker::AVIRecorder::AVIOpen(const char *pFileName, H264Option &pOption)

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

pFileName: The filename of the MP4 file.

pOption: H264 options to apply to the MP4 file.

See: AVIClose()

See: H264Option

SetMaximumAVISize(*self*, *size*)

> **Parameters size** (*unsigned int*) –

thisown
> The membership flag

class PySpin.**ActionCommandResult**
> Bases: object

Action Command Result

C++ includes: SpinnakerDefs.h

DeviceAddress
> ActionCommandResult_DeviceAddress_get(self) -> unsigned int

> > **Parameters self** (*Spinnaker::ActionCommandResult **) –

Status
> ActionCommandResult_Status_get(self) -> Spinnaker::ActionCommandStatus

> > **Parameters self** (*Spinnaker::ActionCommandResult **) –

thisown
> The membership flag

class PySpin.**ArrivalEvent**
> Bases: *PySpin.IArrivalEvent*

An event handler for capturing the device arrival event.

C++ includes: ArrivalEvent.h

OnDeviceArrival(*self*, *serialNumber*)

> > **Parameters serialNumber** (*uint64_t*) –

virtual void Spinnaker::ArrivalEvent::OnDeviceArrival(uint64_t serialNumber)=0

Callback to the device arrival event.

thisown
> The membership flag

class PySpin.**BMPOption**
> Bases: object

> Options for saving Bitmap image.

> C++ includes: SpinnakerDefs.h

> **indexedColor_8bit**
> > BMPOption_indexedColor_8bit_get(self) -> bool

> > > Parameters **self** (*Spinnaker::BMPOption **) –

> **reserved**
> > BMPOption_reserved_get(self) -> unsigned int [16]

> > > Parameters **self** (*Spinnaker::BMPOption **) –

> **thisown**
> > The membership flag

class PySpin.**BooleanNode**(**args*, ***kwargs*)
> Bases: *PySpin.IBoolean*, *PySpin.ValueNode*

> Interface for string properties.

> C++ includes: BooleanNode.h

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

> > Parameters

> > > • **Verify** (*bool*) –

> > > • **IgnoreCache** (*bool*) –

> > > • **Verify=False) -> bool** (GetValue(self,)) –

> > > • **Verify** –

> > > • **-> bool** (GetValue(self)) –

> > > • **self** (*Spinnaker::GenApi::BooleanNode const **) –

> > bool Spinnaker::GenApi::BooleanNode::GetValue(bool Verify=false, bool IgnoreCache=false) const

> > Get node value

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

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

> > The value read.

> **SetReference**(*self*, *pBase*)

> > Parameters **pBase** (*Spinnaker::GenApi::INode **) –

> > virtual void Spinnaker::GenApi::BooleanNode::SetReference(INode *pBase)

> > overload SetReference for Value

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

> > Parameters

> > > • **Value** (*bool*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue](self,)) –

- **Value** –

void Spinnaker::GenApi::BooleanNode::SetValue(bool Value, bool Verify=true)

Set node value

Value: The value to set.

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

thisown
> The membership flag

class PySpin.**CBasePtr**(**args*)
> Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBase,IBase > const **) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBase,IBase > const **) –

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

thisown
> The membership flag

class 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*) –

> - **ValueStr)** ([FromString](self,)) –

> - **ValueStr** –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > const **) –

GetAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetCastAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetChildren(*self*, *LinkType*)

> **Parameters**
>
> - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –
>
> - **GetChildren(self)** –
>
> - **self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > const **) –

GetDescription(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetDeviceName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetDisplayName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetDocuURL(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetEventID(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

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

> **Parameters**
>
> - **FullQualified** (*bool*) –
>
> - **-> gcstring** (GetName(self)) –
>
> - **self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const **) –

GetNode(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > **) –

GetNodeMap(*self*) → INodeMap

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

GetParents (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

GetPollingTime (*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

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

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

GetPropertyNames (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

GetSelectedFeatures (*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetSelectingFeatures (*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetToolTip (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> bool** (GetValue(*self*,) –
>
> - **Verify** –
>
> - **-> bool** (GetValue(*self*)) –
>
> - **self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > const *)* –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase >*
> *const *)* –

ImposeAccessMode (*self*, *ImposedAccessMode*)

> > Parameters **ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility (*self*, *ImposedVisibility*)

> > Parameters **ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode (*self*)

> Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > **) –

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

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsCachable (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsDeprecated (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsFeature (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsSelector (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsStreamable (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

IsValid (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

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

IsValueCacheValid (*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IBoolean, IBase > const **) –

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*) –
> - **Value)** ([SetValue(self,)](#) –

- **Value** –

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

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> gcstring** (`ToString(self,`) –
> > - **Verify** –
> > - **-> gcstring** (`ToString(self)`) –
> > - **self** (*Spinnaker::GenApi::CPointer< IBoolean,IBase > **) –

thisown
> The membership flag

class `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*) –
> > > - **ValueStr)** (`FromString(self,`) –
> > > - **ValueStr** –
>
> **GetAccessMode**(*self*) → Spinnaker::GenApi::EAccessMode
>
> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –
>
> **GetAlias**(*self*) → INode
>
> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –
>
> **GetCachingMode**(*self*) → Spinnaker::GenApi::ECachingMode
>
> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –
>
> **GetCastAlias**(*self*) → INode
>
> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –
>
> **GetChildren**(*self*, *LinkType*)
>
> > Parameters
> >
> > > - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –

- **GetChildren(self)** –

- **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetDescription(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetDeviceName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetDisplayName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetDocuURL(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetEventID(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetFeatures(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

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

> **Parameters**
>
> - **FullQualified**(*bool*) –
>
> - **-> gcstring**(GetName*(self)*) –
>
> - **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetNode(*self*) → INode

> **Parameters self**(*Spinnaker::GenApi::CPointer< ICategory,IBase > **) –

GetNodeMap(*self*) → INodeMap

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetParents(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > const **) –

GetPollingTime(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

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

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

GetPropertyNames(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

GetSelectedFeatures(*self*, *arg2*)

> **Parameters arg2**(*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

> **Parameters arg2**(*FeatureList_t &*) –

GetToolTip(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

> **Parameters ImposedAccessMode**(*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

> **Parameters ImposedVisibility**(*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

> **Parameters self**(*Spinnaker::GenApi::CPointer< ICategory,IBase > **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

IsCachable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

IsDeprecated(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >
> const **) –

IsFeature(*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >*
> > *const **) –

IsSelector(*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >*
> > *const **) –

IsStreamable(*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >*
> > *const **) –

IsValid(*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >*
> > *const **) –

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

IsValueCacheValid(*self*) → bool

> > Parameters **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase >*
> > *const **) –

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*) –
> > - **Verify=False) -> gcstring** (*ToString(self,*) –
> > - **Verify** –
> > - **-> gcstring** (*ToString(self)*) –
> > - **self** (*Spinnaker::GenApi::CPointer< ICategory,IBase > **) –

thisown
> The membership flag

class 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*) –

- **Execute(self)** –

- **self** (*Spinnaker::GenApi::CPointer< ICommand,IBase > ⋆*) –

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

> **Parameters**

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

- **Verify** (*bool*) –

- **ValueStr)** (*FromString(self,)* –

- **ValueStr** –

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

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetAlias(*self*) → INode

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

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

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetCastAlias(*self*) → INode

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetChildren(*self*, *LinkType*)

> **Parameters**

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

- **GetChildren(self)** –

- **self** (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetDescription(*self*) → gcstring

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetDeviceName(*self*) → gcstring

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetDisplayName(*self*) → gcstring

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetDocuURL(*self*) → gcstring

> **Parameters self**   (*Spinnaker::GenApi::CPointer< ICommand,IBase > const ⋆*) –

GetEventID(*self*) → gcstring

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

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

> **Parameters**
>
> > • **FullQualified**(*bool*) –
> >
> > • **-> gcstring**(GetName(*self*)) –
> >
> > • **self**(*Spinnaker::GenApi::CPointer< ICommand,IBase > const **) –

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

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

GetNode(*self*) → INode

> > **Parameters self**(*Spinnaker::GenApi::CPointer< ICommand,IBase > **) –

GetNodeMap(*self*) → INodeMap

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

GetParents(*self*)

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

GetPollingTime(*self*) → int64_t

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

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

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

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

> **Parameters**
>
> > • **PropertyName**(*Spinnaker::GenICam::gcstring const &*) –
> >
> > • **ValueStr**(*Spinnaker::GenICam::gcstring &*) –
> >
> > • **AttributeStr**(*Spinnaker::GenICam::gcstring &*) –

GetPropertyNames(*self*)

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

GetSelectedFeatures(*self*, *arg2*)

> > **Parameters arg2**(*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

> > **Parameters arg2**(*FeatureList_t &*) –

GetToolTip(*self*) → gcstring

> > **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> > *const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

> **Parameters ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

> **Parameters ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase > **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsCachable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsDeprecated(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsDone(*self*, *Verify=True*) → bool

> **Parameters**
>
> - **Verify** (*bool*) –
> - **-> bool** (*IsDone(self)*) –
> - **self** (*Spinnaker::GenApi::CPointer< ICommand,IBase > **) –

IsFeature(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsSelector(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsStreamable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

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

IsValueCacheValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< ICommand,IBase >*
> *const **) –

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*) –
> - **Verify=False) -> gcstring** (*ToString(self,*)) –
> - **Verify** –
> - **-> gcstring** (*ToString(self)*) –
> - **self** (*Spinnaker::GenApi::CPointer< ICommand,IBase > **) –

thisown
> The membership flag

class 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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

GetProductGuid(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

GetSchemaVersion(*self*, *Version*)

> **Parameters Version** (*Spinnaker::GenICam::Version_t &*) –

GetStandardNameSpace(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

GetToolTip(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

GetVendorName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

GetVersionGuid(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > **) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IDeviceInfo,INodeMap > const **) –

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

thisown
> The membership flag

class 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*) –
> >
> > - **ValueStr)** ([FromString](self,)) –
> >
> > - **ValueStr** –

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

> > **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

> **GetAlias**(*self*) → INode

> > **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

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

> > **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

> **GetCastAlias**(*self*) → INode

> > **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

> **GetChildren**(*self*, *LinkType*)

> > **Parameters**
> >
> > - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –

- **GetChildren(self)** –

- **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetDescription(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetDeviceName(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetDisplayName(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetDocuURL(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetEventID(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

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

> Parameters
>
> - **FullQualified**(*bool*) –
>
> - **-> gcstring**([GetName(*self*)]) –
>
> - **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetNode(*self*) → INode

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > ***) –

GetNodeMap(*self*) → INodeMap

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetNumericValue(*self*) → double

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > ***) –

GetParents(*self*)

> Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const ***) –

GetPollingTime(*self*) → int64_t

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

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

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

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

Parameters

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

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

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

GetPropertyNames(*self*)

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

GetSelectedFeatures(*self*, *arg2*)

Parameters arg2 (*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

Parameters arg2 (*FeatureList_t &*) –

GetSymbolic(*self*) → gcstring

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

GetToolTip(*self*) → gcstring

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

GetValue(*self*) → int64_t

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > **)
–

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

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

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

ImposeVisibility(*self*, *ImposedVisibility*)

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

InvalidateNode(*self*)

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > **)
–

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

Parameters self (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase >*
const *) –

IsCachable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

IsDeprecated(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

IsFeature(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

IsSelector(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

IsSelfClearing(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > **) –

IsStreamable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

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

IsValueCacheValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > const **) –

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*) –
> - **Verify=False) -> gcstring** (ToString(*self,*) –
> - **Verify** –
> - **-> gcstring** (ToString(*self*)) –
> - **self** (*Spinnaker::GenApi::CPointer< IEnumEntry,IBase > **) –

thisown

> The membership flag

class `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*) –

            • **`ValueStr`)** (`FromString`*(self,)* –

            • **`ValueStr`** –

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

        Parameters `self` (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

    `GetAlias`(*self*) → INode

        Parameters `self` (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

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

        Parameters `self` (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

    `GetCastAlias`(*self*) → INode

        Parameters `self` (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

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

        Parameters

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

            • **`GetChildren(self)`** –

            • **`self`** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

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

        Parameters

            • **`Verify`** (*bool*) –

            • **`IgnoreCache`** (*bool*) –

            • **`Verify=False) -> IEnumEntry`** (`GetCurrentEntry`*(self,)* –

            • **`Verify`** –

            • **`-> IEnumEntry`** (`GetCurrentEntry`*(self)*) –

            • **`self`** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > **) –

GetDescription (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

GetDeviceName (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

GetDisplayName (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

GetDocuURL (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

GetEntries (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > **) –

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

> **Parameters IntValue** (*int64_t const*) –

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

> **Parameters Symbolic** (*Spinnaker::GenICam::gcstring const &*) –

GetEventID (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> int64_t** (GetIntValue(self,) –
> - **Verify** –
> - **-> int64_t** (GetIntValue(self)) –
> - **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > **) –

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

> **Parameters**
>
> - **FullQualified** (*bool*) –
> - **-> gcstring** (GetName(self)) –
> - **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const **) –

GetNode(*self*) → INode

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *) –

GetNodeMap(*self*) → INodeMap

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

GetParents(*self*)

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

GetPollingTime(*self*) → int64_t

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

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

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

GetPropertyNames(*self*)

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase >*
>   *const *) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

>   Parameters **ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

>   Parameters **ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > ⋆*) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsCachable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsDeprecated(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsFeature(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsSelector(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsStreamable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

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

IsValueCacheValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IEnumeration,IBase > const ⋆*) –

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

> **Parameters pCallback** (*Spinnaker::GenApi::CNodeCallback ⋆*) –

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

> **Parameters**
>
> > - **Value** (*int64_t*) –
> > - **Verify** (*bool*) –
> > - **Value)** (*SetIntValue(self,)*) –
> > - **Value** –

SetReference(*self*, *pBase*)

> **Parameters pBase** (*INode ⋆*) –

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

> Parameters
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> gcstring** (*ToString* (*self,*) –
>
> - **Verify** –
>
> - **-> gcstring** (*ToString* (*self*)) –
>
> - **self** (*Spinnaker::GenApi::CPointer< IEnumeration, IBase > **) –

thisown
>   The membership flag

class PySpin.**CFeatureBag**
>   Bases: *PySpin.IPersistScript*

>   Bag holding streamable features of a nodetree.

>   C++ includes: Persistence.h

>   **GetFeatureBagHandle** (*self*) → void *

>   > Parameters **self** (*Spinnaker::GenApi::CFeatureBag **) –

>   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, Verify=True) -> bool** (*LoadFromBag* (*self,*) –
>   >
>   > - **pNodeMap** –
>   >
>   > - **Verify** –
>   >
>   > - **pNodeMap) -> bool** (*LoadFromBag* (*self,*) –
>   >
>   > - **pNodeMap** –

>   bool Spinnaker::GenApi::CFeatureBag::LoadFromBag(INodeMap *pNodeMap, bool Verify=true, GenI-Cam::gcstring_vector *pErrorList=NULL)

>   Loads the features from the bag to the node tree

>   pNodeMap: The node map

>   Verify: If true, all streamable features are read back

>   pErrorList: If an error occurs during loading the error message is stored in the list and the loading continues

>   For Verify=true the list of names in the feature bag is replayed again. If a node is a selector it's value is set to the value from the feature bag If not the value is read from the camera and compared with the value from the feature bag.

>   **PersistFeature** (*self*, *item*)

>   > Parameters **item** (*Spinnaker::GenApi::IValue &*) –

virtual void Spinnaker::GenApi::CFeatureBag::PersistFeature(IValue &item)

Stores a feature

SetInfo(*self*, *Info*)

Parameters **Info** (*Spinnaker::GenICam::gcstring &*) –

virtual void Spinnaker::GenApi::CFeatureBag::SetInfo(GenICam::gcstring &Info)

sets information about the node map

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

Parameters

- **pNodeMap** (*Spinnaker::GenApi::INodeMap **) –

- **MaxNumPersistSkriptEntries** (*int const*) –

- **pNodeMap) -> int64_t** (*StoreToBag(self,*) –

- **pNodeMap** –

int64_t   Spinnaker::GenApi::CFeatureBag::StoreToBag(INodeMap   *****pNodeMap,   const   int
MaxNumPersistSkriptEntries=-1)

Stores the streamable nodes to this feature bag.

pNodeMap: The node map to persist

MaxNumPersistSkriptEntries: The max number of entries in the container; -1 means unlimited

number of entries in the bag

thisown
The membership flag

class PySpin.**CFloatPtr**(**args*)
Bases: PySpin._SWIG_CFltPtr

SmartPointer for IFloat interface pointer

C++ includes: Pointer.h

GetEnumAlias(*self*) → IEnumeration

Parameters **self** (*Spinnaker::GenApi::CFloatPtr **) –

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

gets the interface of an enum alias node.

GetIntAlias(*self*) → IInteger

Parameters **self** (*Spinnaker::GenApi::CFloatPtr **) –

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

gets the interface of an integer alias node.

thisown
The membership flag

class 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*) –
>
>   - **ValueStr)** ([FromString(self,)](#)) –
>
>   - **ValueStr** –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetAlias(*self*) → INode

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetCastAlias(*self*) → INode

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetChildren(*self*, *LinkType*)

>   Parameters
>
>   - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –
>
>   - **GetChildren(self)** –
>
>   - **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetDescription(*self*) → gcstring

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetDeviceName(*self*) → gcstring

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetDisplayName(*self*) → gcstring

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetDocuURL(*self*) → gcstring

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetEventID(*self*) → gcstring

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

GetInc(*self*) → int64_t

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

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

> Parameters
>
> > • **bounded** (*bool*) –
> >
> > • **-> int64_autovector_t** (*GetListOfValidValues(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

GetMax(*self*) → int64_t

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

GetMin(*self*) → int64_t

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

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

> Parameters
>
> > • **FullQualified** (*bool*) –
> >
> > • **-> gcstring** (*GetName(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const *)* –

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

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

GetNode(*self*) → INode

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > *)* –

GetNodeMap(*self*) → INodeMap

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

GetParents(*self*)

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

GetPollingTime(*self*) → int64_t

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

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

> Parameters **self**   (*Spinnaker::GenApi::CPointer< IInteger,IBase >*
> *const *)* –

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

Parameters

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

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

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

GetPropertyNames (*self*)

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > **) –

GetSelectedFeatures (*self*, *arg2*)

> Parameters **arg2** (*FeatureList_t &*) –

GetSelectingFeatures (*self*, *arg2*)

> Parameters **arg2** (*FeatureList_t &*) –

GetToolTip (*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

GetUnit (*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > **) –

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> int64_t** (GetValue *(self,)*) –

- **Verify** –

- **-> int64_t** (GetValue *(self)*) –

- **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > **) –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

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*)

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > **) –

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

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsCachable(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsDeprecated(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsFeature(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsSelector(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsStreamable(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

IsValid(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

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

IsValueCacheValid(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IInteger,IBase > const **) –

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*) –
>   - **Value)** ([SetValue](*self,*)) –
>   - **Value** –

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

>   **Parameters**

- **Verify**(*bool*) –

- **IgnoreCache**(*bool*) –

- **Verify=False) -> gcstring**([ToString](*self,*) –

- **Verify** –

- **-> gcstring**([ToString](*self*)) –

- **self**(*Spinnaker::GenApi::CPointer< IInteger,IBase > **) –

thisown
> The membership flag

class PySpin.**CNodeMapDynPtr**(**args*)
> Bases: object

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

ClearAllNodes(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > **) –

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

> **Parameters**

> - **pPort**(*IPort **) –

> - **PortName**(*Spinnaker::GenICam::gcstring const &*) –

> - **pPort) -> bool**([Connect](*self,*) –

> - **pPort** –

ExtractIndependentSubtree(*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *Extract-edSubtree*)

> **Parameters**

> - **XMLData**(*Spinnaker::GenICam::gcstring const &*) –

> - **InjectXMLData**(*Spinnaker::GenICam::gcstring const &*) –

> - **SubTreeRootNodeName**(*Spinnaker::GenICam::gcstring const &*) –

> - **ExtractedSubtree**(*Spinnaker::GenICam::gcstring &*) –

GetDeviceName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > **) –

GetNode(*self*, *Name*) → INode

> **Parameters Name**(*Spinnaker::GenICam::gcstring const &*) –

GetNodes(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > const **) –

GetNumNodes(*self*) → uint64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > const **) –

GetSupportedSchemaVersions(*self*)

>   **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > **) –

InvalidateNodes(*self*)

>   **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > const **) –

IsValid(*self*) → bool

>   **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMapDyn,INodeMap > const **) –

>   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*)

>   **Parameters**

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **XMLValidation** (*uint32_t const*) –

- **XMLFileName, StyleSheetFileName, OutputFileName)** (*PreprocessXMLFromFile(self,)* –

- **XMLFileName** –

- **StyleSheetFileName** –

- **OutputFileName** –

PreprocessXMLFromZIPFile (*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*, *XML-Validation*)

> Parameters

- **XMLFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **OutputFileName** (*Spinnaker::GenICam::gcstring const &*) –

- **XMLValidation** (*uint32_t const*) –

- **XMLFileName, StyleSheetFileName, OutputFileName)** (*PreprocessXMLFromZIPFile(self,)* –

- **XMLFileName** –

- **StyleSheetFileName** –

- **OutputFileName** –

thisown
> The membership flag

class 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 &*) –

- **pPort) -> bool** (*Connect(self,)* –

- **pPort** –

GetDeviceName (*self*) → gcstring

> Parameters self (*Spinnaker::GenApi::CPointer< INodeMap,INodeMap > **) –

GetNode (*self*, *Name*) → INode

> Parameters Name (*Spinnaker::GenICam::gcstring const &*) –

GetNodes(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMap,INodeMap > const **) –

GetNumNodes(*self*) → uint64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMap,INodeMap > const **) –

InvalidateNodes(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMap,INodeMap > const **) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INodeMap,INodeMap > const **) –

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

Poll(*self*, *ElapsedTime*)

> **Parameters ElapsedTime** (*int64_t*) –

thisown
> The membership flag

class 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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetCastAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetChildren(*self*, *LinkType*)

> **Parameters**
>
> - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –
> - **GetChildren(self)** –

> - **self**(*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetDescription(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetDeviceName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetDisplayName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetDocuURL(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetEventID(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

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

> **Parameters**
>
> > - **FullQualified**(*bool*) –
> > - **-> gcstring**(GetName*(self)*) –
> > - **self**(*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetNodeMap(*self*) → INodeMap

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetParents(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

GetPollingTime(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

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

> **Parameters**
>
> > - **PropertyName**(*Spinnaker::GenICam::gcstring const &*) –

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

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

GetPropertyNames(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

GetSelectedFeatures(*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetToolTip(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

> **Parameters ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

> **Parameters ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > ⋆*) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsCachable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsDeprecated(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsFeature(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsSelector(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsStreamable(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const ⋆*) –

IsValid(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< INode,IBase > const **) –

> 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 **) –

thisown
> The membership flag

class 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*) –
> - **ValueStr)** ([FromString](*self,*) –
> - **ValueStr** –

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

> **Parameters**

> - **pBuffer**(*uint8_t **) –
> - **Verify**(*bool*) –
> - **IgnoreCache**(*bool*) –
> - **pBuffer, Verify=False)** ([Get](*self,*) –
> - **pBuffer** –
> - **Verify** –
> - **pBuffer)** ([Get](*self,*) –
> - **pBuffer** –

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

> 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

> > Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> > const **) –

GetAddress(*self*) → int64_t

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > **) –

GetAlias(*self*) → INode

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetCastAlias(*self*) → INode

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetChildren(*self*, *LinkType*)

> Parameters
>
> - **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –
>
> - **GetChildren(self)** –
>
> - **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **)
>   –

GetDescription(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetDeviceName(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetDisplayName(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetDocuURL(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetEventID(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetLength(*self*) → int64_t

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > **) –

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

> Parameters
>
> - **FullQualified** (*bool*) –

- **-> gcstring** (GetName(*self*)) –

- **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **)
  –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetNode (*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > **) –

GetNodeMap (*self*) → INodeMap

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetParents (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetPollingTime (*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

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

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

GetPropertyNames (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

GetSelectedFeatures (*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetSelectingFeatures (*self*, *arg2*)

> **Parameters arg2** (*FeatureList_t &*) –

GetToolTip (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IRegister,IBase >
> const **) –

ImposeAccessMode (*self*, *ImposedAccessMode*)

> Parameters **ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility (*self*, *ImposedVisibility*)

> Parameters **ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode (*self*)

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > **) –

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

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsCachable (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsDeprecated (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsFeature (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsSelector (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsStreamable (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

IsValid (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

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

IsValueCacheValid (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CPointer< IRegister,IBase > const **) –

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

> Parameters **pCallback** (*Spinnaker::GenApi::CNodeCallback **) –

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

> Parameters
>
> - **pBuffer** (*uint8_t const **) –
> - **Verify** (*bool*) –
> - **pBuffer)** (Set(*self,*) –
> - **pBuffer** –

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

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`) –

> > • **Verify=False) -> gcstring** (`ToString(self,`) –

> > • **Verify** –

> > • **-> gcstring** (`ToString(self)`) –

> > • **self** (`Spinnaker::GenApi::CPointer< IRegister,IBase > *`) –

thisown
> The membership flag

class PySpin.**CSelectorPtr**(**args*)

> Bases: `object`

Encapsulates a GenApi pointer dealing with the dynamic_cast automatically.

C++ includes: Pointer.h

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

> Parameters **self** (`Spinnaker::GenApi::CPointer< ISelector,IBase > const *`) –

GetSelectedFeatures(*self*, *arg2*)

> Parameters **arg2** (`FeatureList_t &`) –

GetSelectingFeatures(*self*, *arg2*)

> Parameters **arg2** (`FeatureList_t &`) –

IsSelector(*self*) → bool

> Parameters **self** (`Spinnaker::GenApi::CPointer< ISelector,IBase > const *`) –

IsValid(*self*) → bool

> Parameters **self** (`Spinnaker::GenApi::CPointer< ISelector,IBase > const *`) –

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

thisown
> The membership flag

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

> Bases: *PySpin.Node*

The set of selectors selecting a given node

C++ includes: SelectorSet.h

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

> Parameters

>> • **Incremental** (*bool*) –

>> • **GetSelectorList(self)** –

>> • **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

IsEmpty(*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

> returns true if no selectors are present

Restore(*self*)

> Parameters **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

SetFirst(*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

SetNext(*self*, *Tick=True*) → bool

> Parameters

>> • **Tick** (*bool*) –

>> • **-> bool** (SetNext(self)) –

>> • **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

ToString(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::CSelectorSet **) –

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

thisown
> The membership flag

class 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*) –

- **ValueStr)** ([FromString](*self,*)) –

- **ValueStr** –

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

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetAlias(*self*) → INode

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

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

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetCastAlias(*self*) → INode

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetChildren(*self*, *LinkType*)

>   **Parameters**
>
>   - **LinkType**(*enum Spinnaker::GenApi::ELinkType*) –
>
>   - **GetChildren(self)** –
>
>   - **self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

GetDescription(*self*) → gcstring

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetDeviceName(*self*) → gcstring

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetDisplayName(*self*) → gcstring

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetDocuURL(*self*) → gcstring

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetEventID(*self*) → gcstring

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const*
>   ***) –

GetMaxLength(*self*) → int64_t

>   **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > **) –

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

Parameters

- **FullQualified**(*bool*) –

- **-> gcstring**([GetName](*self*)) –

- **self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

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

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

GetNode(*self*) → INode

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > **) –

GetNodeMap(*self*) → INodeMap

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

GetParents(*self*)

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

GetPollingTime(*self*) → int64_t

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

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

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

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

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

GetPropertyNames(*self*)

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

GetSelectedFeatures(*self*, *arg2*)

> **Parameters arg2**(*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

> **Parameters arg2**(*FeatureList_t &*) –

GetToolTip(*self*) → gcstring

> **Parameters self**(*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

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

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> gcstring** (GetValue *(self,)* –

- **Verify** –

- **-> gcstring** (GetValue *(self))* –

- **self** (*Spinnaker::GenApi::CPointer< IString,IBase > **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

ImposeAccessMode (*self*, *ImposedAccessMode*)

> **Parameters ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility (*self*, *ImposedVisibility*)

> **Parameters ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode (*self*)

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > **) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsCachable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsDeprecated (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsFeature (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsSelector (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsStreamable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

IsValid (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const **) –

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

IsValueCacheValid (*self*) → bool

> > **Parameters self** (*Spinnaker::GenApi::CPointer< IString,IBase > const* *) –

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*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> gcstring** (*ToString(self,*) –
> - **Verify** –
> - **-> gcstring** (*ToString(self)*) –
> - **self** (*Spinnaker::GenApi::CPointer< IString,IBase > *) –

thisown
> The membership flag

class `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*) –
> - **ValueStr)** (*FromString(self,*) –
> - **ValueStr** –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const* *) –

GetAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetCastAlias(*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetChildren(*self*, *LinkType*)

> **Parameters**
>
> > • **LinkType**(*enum Spinnaker::GenApi::ELinkType*) –
> >
> > • **GetChildren(self)** –
> >
> > • **self**(*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetDescription(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetDeviceName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetDisplayName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetDocuURL(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetEventID(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

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

> **Parameters**
>
> > • **FullQualified**(*bool*) –
> >
> > • **-> gcstring**([GetName](*self*)) –
> >
> > • **self**(*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

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

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ★*) –

GetNode(*self*) → INode

> **Parameters self**(*Spinnaker::GenApi::CPointer< IValue,IBase > ★*) –

GetNodeMap(*self*) → INodeMap

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

GetParents(*self*)

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

GetPollingTime(*self*) → int64_t

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

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

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

GetPropertyNames(*self*)

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

GetSelectedFeatures(*self*, *arg2*)

>   Parameters **arg2** (*FeatureList_t &*) –

GetSelectingFeatures(*self*, *arg2*)

>   Parameters **arg2** (*FeatureList_t &*) –

GetToolTip(*self*) → gcstring

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

>   Parameters **ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

>   Parameters **ImposedVisibility** (*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > ⋆*) –

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

>   Parameters **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsCachable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsDeprecated (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsFeature (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsSelector (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsStreamable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

IsValid (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

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

IsValueCacheValid (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::CPointer< IValue,IBase > const ⋆*) –

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*) –
> - **Verify=False) -> gcstring** (*ToString(self,)*) –
> - **Verify** –
> - **-> gcstring** (*ToString(self))* –
> - **self** (*Spinnaker::GenApi::CPointer< IValue,IBase > ⋆*) –

thisown

> The membership flag

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

> Bases: *PySpin.CameraBase*

The camera object class.

C++ includes: Camera.h

AasRoiEnable
Camera_AasRoiEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

AasRoiHeight
Camera_AasRoiHeight_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AasRoiOffsetX
Camera_AasRoiOffsetX_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AasRoiOffsetY
Camera_AasRoiOffsetY_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AasRoiWidth
Camera_AasRoiWidth_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionAbort
Camera_AcquisitionAbort_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionArm
Camera_AcquisitionArm_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionBurstFrameCount
Camera_AcquisitionBurstFrameCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionFrameCount
Camera_AcquisitionFrameCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionFrameRate
Camera_AcquisitionFrameRate_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionFrameRateEnable
Camera_AcquisitionFrameRateEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionLineRate
Camera_AcquisitionLineRate_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionMode
Camera_AcquisitionMode_get(self) -> IEnumerationT_AcquisitionModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

AcquisitionResultingFrameRate
    Camera_AcquisitionResultingFrameRate_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AcquisitionStart
    Camera_AcquisitionStart_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

AcquisitionStatus
    Camera_AcquisitionStatus_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

AcquisitionStatusSelector
    Camera_AcquisitionStatusSelector_get(self) -> IEnumerationT_AcquisitionStatusSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

AcquisitionStop
    Camera_AcquisitionStop_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

ActionDeviceKey
    Camera_ActionDeviceKey_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ActionGroupKey
    Camera_ActionGroupKey_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ActionGroupMask
    Camera_ActionGroupMask_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ActionQueueSize
    Camera_ActionQueueSize_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ActionSelector
    Camera_ActionSelector_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ActionUnconditionalMode
    Camera_ActionUnconditionalMode_get(self) -> IEnumerationT_ActionUnconditionalModeEnums

        Parameters self (*Spinnaker::Camera **) –

AdcBitDepth
    Camera_AdcBitDepth_get(self) -> IEnumerationT_AdcBitDepthEnums

        Parameters self (*Spinnaker::Camera **) –

AutoAlgorithmSelector
    Camera_AutoAlgorithmSelector_get(self) -> IEnumerationT_AutoAlgorithmSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

AutoExposureControlLoopDamping
    Camera_AutoExposureControlLoopDamping_get(self) -> IFloat

          Parameters self (*Spinnaker::Camera **) –

AutoExposureControlPriority
    Camera_AutoExposureControlPriority_get(self) -> IEnumerationT_AutoExposureControlPriorityEnums

        Parameters self (*Spinnaker::Camera **) –

AutoExposureEVCompensation
    Camera_AutoExposureEVCompensation_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureExposureTimeLowerLimit
    Camera_AutoExposureExposureTimeLowerLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureExposureTimeUpperLimit
    Camera_AutoExposureExposureTimeUpperLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureGainLowerLimit
    Camera_AutoExposureGainLowerLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureGainUpperLimit
    Camera_AutoExposureGainUpperLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureGreyValueLowerLimit
    Camera_AutoExposureGreyValueLowerLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureGreyValueUpperLimit
    Camera_AutoExposureGreyValueUpperLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureLightingMode
    Camera_AutoExposureLightingMode_get(self) -> IEnumerationT_AutoExposureLightingModeEnums

        Parameters self (*Spinnaker::Camera **) –

AutoExposureMeteringMode
    Camera_AutoExposureMeteringMode_get(self) -> IEnumerationT_AutoExposureMeteringModeEnums

        Parameters self (*Spinnaker::Camera **) –

AutoExposureTargetGreyValue
    Camera_AutoExposureTargetGreyValue_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

AutoExposureTargetGreyValueAuto
    Camera_AutoExposureTargetGreyValueAuto_get(self) -> IEnumerationT_AutoExposureTargetGreyValueAutoEnums

        Parameters self (*Spinnaker::Camera **) –

BalanceRatio
    Camera_BalanceRatio_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

BalanceRatioSelector
    Camera_BalanceRatioSelector_get(self) -> IEnumerationT_BalanceRatioSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

BalanceWhiteAuto
    Camera_BalanceWhiteAuto_get(self) -> IEnumerationT_BalanceWhiteAutoEnums

        Parameters self (*Spinnaker::Camera **) –

BalanceWhiteAutoDamping
    Camera_BalanceWhiteAutoDamping_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

BalanceWhiteAutoLowerLimit
    Camera_BalanceWhiteAutoLowerLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

BalanceWhiteAutoProfile
    Camera_BalanceWhiteAutoProfile_get(self) -> IEnumerationT_BalanceWhiteAutoProfileEnums

        Parameters self (*Spinnaker::Camera **) –

BalanceWhiteAutoUpperLimit
    Camera_BalanceWhiteAutoUpperLimit_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

BinningHorizontal
    Camera_BinningHorizontal_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

BinningHorizontalMode
    Camera_BinningHorizontalMode_get(self) -> IEnumerationT_BinningHorizontalModeEnums

        Parameters self (*Spinnaker::Camera **) –

BinningSelector
    Camera_BinningSelector_get(self) -> IEnumerationT_BinningSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

BinningVertical
    Camera_BinningVertical_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

BinningVerticalMode
    Camera_BinningVerticalMode_get(self) -> IEnumerationT_BinningVerticalModeEnums

        Parameters self (*Spinnaker::Camera **) –

BlackLevel
    Camera_BlackLevel_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

BlackLevelAuto
    Camera_BlackLevelAuto_get(self) -> IEnumerationT_BlackLevelAutoEnums

        Parameters self (*Spinnaker::Camera **) –

BlackLevelAutoBalance
    Camera_BlackLevelAutoBalance_get(self) -> IEnumerationT_BlackLevelAutoBalanceEnums

Parameters **self** (*Spinnaker::Camera* *) –

BlackLevelClampingEnable
    Camera_BlackLevelClampingEnable_get(self) -> IBoolean

        Parameters **self** (*Spinnaker::Camera* *) –

BlackLevelRaw
    Camera_BlackLevelRaw_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera* *) –

BlackLevelSelector
    Camera_BlackLevelSelector_get(self) -> IEnumerationT_BlackLevelSelectorEnums

        Parameters **self** (*Spinnaker::Camera* *) –

BsiFlatFieldCorrectionAuto
    Camera_BsiFlatFieldCorrectionAuto_get(self) -> IEnumerationT_BsiFlatFieldCorrectionAutoEnums

        Parameters **self** (*Spinnaker::Camera* *) –

BsiFlatFieldCorrectionAutoDamping
    Camera_BsiFlatFieldCorrectionAutoDamping_get(self) -> IFloat

        Parameters **self** (*Spinnaker::Camera* *) –

BsiFlatFieldCorrectionEnable
    Camera_BsiFlatFieldCorrectionEnable_get(self) -> IBoolean

        Parameters **self** (*Spinnaker::Camera* *) –

BsiFlatFieldCorrectionGain
    Camera_BsiFlatFieldCorrectionGain_get(self) -> IFloat

        Parameters **self** (*Spinnaker::Camera* *) –

BsiFlatFieldCorrectionGainSelector
    Camera_BsiFlatFieldCorrectionGainSelector_get(self) -> IEnumerationT_BsiFlatFieldCorrectionGainSelectorEnums

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkBlackLevel
    Camera_ChunkBlackLevel_get(self) -> IFloat

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkBlackLevelSelector
    Camera_ChunkBlackLevelSelector_get(self) -> IEnumerationT_ChunkBlackLevelSelectorEnums

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkCRC
    Camera_ChunkCRC_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkCounterSelector
    Camera_ChunkCounterSelector_get(self) -> IEnumerationT_ChunkCounterSelectorEnums

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkCounterValue
    Camera_ChunkCounterValue_get(self) -> IInteger

        Parameters **self** (*Spinnaker::Camera* *) –

ChunkEnable
Camera_ChunkEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

ChunkEncoderSelector
Camera_ChunkEncoderSelector_get(self) -> IEnumerationT_ChunkEncoderSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkEncoderStatus
Camera_ChunkEncoderStatus_get(self) -> IEnumerationT_ChunkEncoderStatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkEncoderValue
Camera_ChunkEncoderValue_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkExposureEndLineStatusAll
Camera_ChunkExposureEndLineStatusAll_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkExposureTime
Camera_ChunkExposureTime_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkExposureTimeSelector
Camera_ChunkExposureTimeSelector_get(self) -> IEnumerationT_ChunkExposureTimeSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkFrameID
Camera_ChunkFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkGain
Camera_ChunkGain_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkGainSelector
Camera_ChunkGainSelector_get(self) -> IEnumerationT_ChunkGainSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkHeight
Camera_ChunkHeight_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkImage
Camera_ChunkImage_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkImageComponent
Camera_ChunkImageComponent_get(self) -> IEnumerationT_ChunkImageComponentEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkLinePitch
Camera_ChunkLinePitch_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkLineStatusAll
Camera_ChunkLineStatusAll_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkModeActive
Camera_ChunkModeActive_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

ChunkOffsetX
Camera_ChunkOffsetX_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkOffsetY
Camera_ChunkOffsetY_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkPartSelector
Camera_ChunkPartSelector_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkPixelDynamicRangeMax
Camera_ChunkPixelDynamicRangeMax_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkPixelDynamicRangeMin
Camera_ChunkPixelDynamicRangeMin_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

ChunkPixelFormat
Camera_ChunkPixelFormat_get(self) -> IEnumerationT_ChunkPixelFormatEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkRegionID
Camera_ChunkRegionID_get(self) -> IEnumerationT_ChunkRegionIDEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dAxisMax
Camera_ChunkScan3dAxisMax_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dAxisMin
Camera_ChunkScan3dAxisMin_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateOffset
Camera_ChunkScan3dCoordinateOffset_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateReferenceSelector
Camera_ChunkScan3dCoordinateReferenceSelector_get(self) -> IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

ChunkScan3dCoordinateReferenceValue
Camera_ChunkScan3dCoordinateReferenceValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dCoordinateScale
Camera_ChunkScan3dCoordinateScale_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dCoordinateSelector
Camera_ChunkScan3dCoordinateSelector_get(self) -> IEnumerationT_ChunkScan3dCoordinateSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dCoordinateSystem
Camera_ChunkScan3dCoordinateSystem_get(self) -> IEnumerationT_ChunkScan3dCoordinateSystemEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dCoordinateSystemReference
Camera_ChunkScan3dCoordinateSystemReference_get(self) -> IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dCoordinateTransformSelector
Camera_ChunkScan3dCoordinateTransformSelector_get(self) -> IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dDistanceUnit
Camera_ChunkScan3dDistanceUnit_get(self) -> IEnumerationT_ChunkScan3dDistanceUnitEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dInvalidDataFlag
Camera_ChunkScan3dInvalidDataFlag_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dInvalidDataValue
Camera_ChunkScan3dInvalidDataValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dOutputMode
Camera_ChunkScan3dOutputMode_get(self) -> IEnumerationT_ChunkScan3dOutputModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScan3dTransformValue
Camera_ChunkScan3dTransformValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

ChunkScanLineSelector
Camera_ChunkScanLineSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

ChunkSelector
Camera_ChunkSelector_get(self) -> IEnumerationT_ChunkSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

ChunkSequencerSetActive
Camera_ChunkSequencerSetActive_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkSerialData
Camera_ChunkSerialData_get(self) -> IString

   Parameters self (*Spinnaker::Camera **) –

ChunkSerialDataLength
Camera_ChunkSerialDataLength_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkSerialReceiveOverflow
Camera_ChunkSerialReceiveOverflow_get(self) -> IBoolean

   Parameters self (*Spinnaker::Camera **) –

ChunkSourceID
Camera_ChunkSourceID_get(self) -> IEnumerationT_ChunkSourceIDEnums

   Parameters self (*Spinnaker::Camera **) –

ChunkStreamChannelID
Camera_ChunkStreamChannelID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkTimerSelector
Camera_ChunkTimerSelector_get(self) -> IEnumerationT_ChunkTimerSelectorEnums

   Parameters self (*Spinnaker::Camera **) –

ChunkTimerValue
Camera_ChunkTimerValue_get(self) -> IFloat

   Parameters self (*Spinnaker::Camera **) –

ChunkTimestamp
Camera_ChunkTimestamp_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkTimestampLatchValue
Camera_ChunkTimestampLatchValue_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkTransferBlockID
Camera_ChunkTransferBlockID_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkTransferQueueCurrentBlockCount
Camera_ChunkTransferQueueCurrentBlockCount_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

ChunkTransferStreamID
Camera_ChunkTransferStreamID_get(self) -> IEnumerationT_ChunkTransferStreamIDEnums

   Parameters self (*Spinnaker::Camera **) –

ChunkWidth
Camera_ChunkWidth_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

ClConfiguration
> Camera_ClConfiguration_get(self) -> IEnumerationT_ClConfigurationEnums

>> Parameters **self** (*Spinnaker::Camera **) –

ClTimeSlotsCount
> Camera_ClTimeSlotsCount_get(self) -> IEnumerationT_ClTimeSlotsCountEnums

>> Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationEnable
> Camera_ColorTransformationEnable_get(self) -> IBoolean

>> Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationSelector
> Camera_ColorTransformationSelector_get(self) -> IEnumerationT_ColorTransformationSelectorEnums

>> Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationValue
> Camera_ColorTransformationValue_get(self) -> IFloat

>> Parameters **self** (*Spinnaker::Camera **) –

ColorTransformationValueSelector
> Camera_ColorTransformationValueSelector_get(self) -> IEnumerationT_ColorTransformationValueSelectorEnums

>> Parameters **self** (*Spinnaker::Camera **) –

CounterDelay
> Camera_CounterDelay_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

CounterDuration
> Camera_CounterDuration_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

CounterEventActivation
> Camera_CounterEventActivation_get(self) -> IEnumerationT_CounterEventActivationEnums

>> Parameters **self** (*Spinnaker::Camera **) –

CounterEventSource
> Camera_CounterEventSource_get(self) -> IEnumerationT_CounterEventSourceEnums

>> Parameters **self** (*Spinnaker::Camera **) –

CounterReset
> Camera_CounterReset_get(self) -> ICommand

>> Parameters **self** (*Spinnaker::Camera **) –

CounterResetActivation
> Camera_CounterResetActivation_get(self) -> IEnumerationT_CounterResetActivationEnums

>> Parameters **self** (*Spinnaker::Camera **) –

CounterResetSource
> Camera_CounterResetSource_get(self) -> IEnumerationT_CounterResetSourceEnums

>> Parameters **self** (*Spinnaker::Camera **) –

CounterSelector
Camera_CounterSelector_get(self) -> IEnumerationT_CounterSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

CounterStatus
Camera_CounterStatus_get(self) -> IEnumerationT_CounterStatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

CounterTriggerActivation
Camera_CounterTriggerActivation_get(self) -> IEnumerationT_CounterTriggerActivationEnums

> **Parameters self** (*Spinnaker::Camera **) –

CounterTriggerSource
Camera_CounterTriggerSource_get(self) -> IEnumerationT_CounterTriggerSourceEnums

> **Parameters self** (*Spinnaker::Camera **) –

CounterValue
Camera_CounterValue_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

CounterValueAtReset
Camera_CounterValueAtReset_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

CxpConnectionSelector
Camera_CxpConnectionSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

CxpConnectionTestErrorCount
Camera_CxpConnectionTestErrorCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

CxpConnectionTestMode
Camera_CxpConnectionTestMode_get(self) -> IEnumerationT_CxpConnectionTestModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

CxpConnectionTestPacketCount
Camera_CxpConnectionTestPacketCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

CxpLinkConfiguration
Camera_CxpLinkConfiguration_get(self) -> IEnumerationT_CxpLinkConfigurationEnums

> **Parameters self** (*Spinnaker::Camera **) –

CxpLinkConfigurationPreferred
Camera_CxpLinkConfigurationPreferred_get(self) -> IEnumerationT_CxpLinkConfigurationPreferredEnums

> **Parameters self** (*Spinnaker::Camera **) –

CxpLinkConfigurationStatus
Camera_CxpLinkConfigurationStatus_get(self) -> IEnumerationT_CxpLinkConfigurationStatusEnums

> **Parameters self** (*Spinnaker::Camera **) –

CxpPoCxpAuto
Camera_CxpPoCxpAuto_get(self) -> ICommand

Parameters self (*Spinnaker::Camera **) –

CxpPoCxpStatus
    Camera_CxpPoCxpStatus_get(self) -> IEnumerationT_CxpPoCxpStatusEnums

    Parameters self (*Spinnaker::Camera **) –

CxpPoCxpTripReset
    Camera_CxpPoCxpTripReset_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

CxpPoCxpTurnOff
    Camera_CxpPoCxpTurnOff_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

DecimationHorizontal
    Camera_DecimationHorizontal_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DecimationHorizontalMode
    Camera_DecimationHorizontalMode_get(self) -> IEnumerationT_DecimationHorizontalModeEnums

    Parameters self (*Spinnaker::Camera **) –

DecimationSelector
    Camera_DecimationSelector_get(self) -> IEnumerationT_DecimationSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

DecimationVertical
    Camera_DecimationVertical_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DecimationVerticalMode
    Camera_DecimationVerticalMode_get(self) -> IEnumerationT_DecimationVerticalModeEnums

    Parameters self (*Spinnaker::Camera **) –

DefectTableApply
    Camera_DefectTableApply_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

DefectTableCoordinateX
    Camera_DefectTableCoordinateX_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DefectTableCoordinateY
    Camera_DefectTableCoordinateY_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DefectTableFactoryRestore
    Camera_DefectTableFactoryRestore_get(self) -> ICommand

    Parameters self (*Spinnaker::Camera **) –

DefectTableIndex
    Camera_DefectTableIndex_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DefectTablePixelCount
    Camera_DefectTablePixelCount_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DefectTableSave
    Camera_DefectTableSave_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

Deinterlacing
    Camera_Deinterlacing_get(self) -> IEnumerationT_DeinterlacingEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceCharacterSet
    Camera_DeviceCharacterSet_get(self) -> IEnumerationT_DeviceCharacterSetEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceClockFrequency
    Camera_DeviceClockFrequency_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

DeviceClockSelector
    Camera_DeviceClockSelector_get(self) -> IEnumerationT_DeviceClockSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionSelector
    Camera_DeviceConnectionSelector_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionSpeed
    Camera_DeviceConnectionSpeed_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DeviceConnectionStatus
    Camera_DeviceConnectionStatus_get(self) -> IEnumerationT_DeviceConnectionStatusEnums

        Parameters self (*Spinnaker::Camera **) –

DeviceEventChannelCount
    Camera_DeviceEventChannelCount_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

DeviceFamilyName
    Camera_DeviceFamilyName_get(self) -> IString

        Parameters self (*Spinnaker::Camera **) –

DeviceFeaturePersistenceEnd
    Camera_DeviceFeaturePersistenceEnd_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

DeviceFeaturePersistenceStart
    Camera_DeviceFeaturePersistenceStart_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

DeviceFirmwareVersion
    Camera_DeviceFirmwareVersion_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera **) –

DeviceGenCPVersionMajor
> Camera_DeviceGenCPVersionMajor_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceGenCPVersionMinor
> Camera_DeviceGenCPVersionMinor_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceID
> Camera_DeviceID_get(self) -> IString

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceIndicatorMode
> Camera_DeviceIndicatorMode_get(self) -> IEnumerationT_DeviceIndicatorModeEnums

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkBandwidthReserve
> Camera_DeviceLinkBandwidthReserve_get(self) -> IFloat

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkCommandTimeout
> Camera_DeviceLinkCommandTimeout_get(self) -> IFloat

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkConnectionCount
> Camera_DeviceLinkConnectionCount_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkCurrentThroughput
> Camera_DeviceLinkCurrentThroughput_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkHeartbeatMode
> Camera_DeviceLinkHeartbeatMode_get(self) -> IEnumerationT_DeviceLinkHeartbeatModeEnums

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkHeartbeatTimeout
> Camera_DeviceLinkHeartbeatTimeout_get(self) -> IFloat

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkSelector
> Camera_DeviceLinkSelector_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkSpeed
> Camera_DeviceLinkSpeed_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkThroughputLimit
> Camera_DeviceLinkThroughputLimit_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

DeviceLinkThroughputLimitMode
    Camera_DeviceLinkThroughputLimitMode_get(self) -> IEnumerationT_DeviceLinkThroughputLimitModeEnums

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestEntrySelector
    Camera_DeviceManifestEntrySelector_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestPrimaryURL
    Camera_DeviceManifestPrimaryURL_get(self) -> IString

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestSchemaMajorVersion
    Camera_DeviceManifestSchemaMajorVersion_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestSchemaMinorVersion
    Camera_DeviceManifestSchemaMinorVersion_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestSecondaryURL
    Camera_DeviceManifestSecondaryURL_get(self) -> IString

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestXMLMajorVersion
    Camera_DeviceManifestXMLMajorVersion_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestXMLMinorVersion
    Camera_DeviceManifestXMLMinorVersion_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManifestXMLSubMinorVersion
    Camera_DeviceManifestXMLSubMinorVersion_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceManufacturerInfo
    Camera_DeviceManufacturerInfo_get(self) -> IString

    Parameters self (*Spinnaker::Camera **) –

DeviceMaxThroughput
    Camera_DeviceMaxThroughput_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

DeviceModelName
    Camera_DeviceModelName_get(self) -> IString

    Parameters self (*Spinnaker::Camera **) –

DevicePowerSupplySelector
    Camera_DevicePowerSupplySelector_get(self) -> IEnumerationT_DevicePowerSupplySelectorEnums

    Parameters self (*Spinnaker::Camera **) –

DeviceRegistersCheck
    Camera_DeviceRegistersCheck_get(self) -> ICommand

Parameters self (*Spinnaker::Camera* *) –

DeviceRegistersEndianness
Camera_DeviceRegistersEndianness_get(self) -> IEnumerationT_DeviceRegistersEndiannessEnums

Parameters self (*Spinnaker::Camera* *) –

DeviceRegistersStreamingEnd
Camera_DeviceRegistersStreamingEnd_get(self) -> ICommand

Parameters self (*Spinnaker::Camera* *) –

DeviceRegistersStreamingStart
Camera_DeviceRegistersStreamingStart_get(self) -> ICommand

Parameters self (*Spinnaker::Camera* *) –

DeviceRegistersValid
Camera_DeviceRegistersValid_get(self) -> IBoolean

Parameters self (*Spinnaker::Camera* *) –

DeviceReset
Camera_DeviceReset_get(self) -> ICommand

Parameters self (*Spinnaker::Camera* *) –

DeviceSFNCVersionMajor
Camera_DeviceSFNCVersionMajor_get(self) -> IInteger

Parameters self (*Spinnaker::Camera* *) –

DeviceSFNCVersionMinor
Camera_DeviceSFNCVersionMinor_get(self) -> IInteger

Parameters self (*Spinnaker::Camera* *) –

DeviceSFNCVersionSubMinor
Camera_DeviceSFNCVersionSubMinor_get(self) -> IInteger

Parameters self (*Spinnaker::Camera* *) –

DeviceScanType
Camera_DeviceScanType_get(self) -> IEnumerationT_DeviceScanTypeEnums

Parameters self (*Spinnaker::Camera* *) –

DeviceSerialNumber
Camera_DeviceSerialNumber_get(self) -> IString

Parameters self (*Spinnaker::Camera* *) –

DeviceSerialPortBaudRate
Camera_DeviceSerialPortBaudRate_get(self) -> IEnumerationT_DeviceSerialPortBaudRateEnums

Parameters self (*Spinnaker::Camera* *) –

DeviceSerialPortSelector
Camera_DeviceSerialPortSelector_get(self) -> IEnumerationT_DeviceSerialPortSelectorEnums

Parameters self (*Spinnaker::Camera* *) –

DeviceStreamChannelCount
Camera_DeviceStreamChannelCount_get(self) -> IInteger

Parameters self (*Spinnaker::Camera* *) –

DeviceStreamChannelEndianness
Camera_DeviceStreamChannelEndianness_get(self) -> IEnumerationT_DeviceStreamChannelEndiannessEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelLink
Camera_DeviceStreamChannelLink_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelPacketSize
Camera_DeviceStreamChannelPacketSize_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelSelector
Camera_DeviceStreamChannelSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceStreamChannelType
Camera_DeviceStreamChannelType_get(self) -> IEnumerationT_DeviceStreamChannelTypeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLType
Camera_DeviceTLType_get(self) -> IEnumerationT_DeviceTLTypeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionMajor
Camera_DeviceTLVersionMajor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionMinor
Camera_DeviceTLVersionMinor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTLVersionSubMinor
Camera_DeviceTLVersionSubMinor_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTapGeometry
Camera_DeviceTapGeometry_get(self) -> IEnumerationT_DeviceTapGeometryEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTemperature
Camera_DeviceTemperature_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

DeviceTemperatureSelector
Camera_DeviceTemperatureSelector_get(self) -> IEnumerationT_DeviceTemperatureSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceType
Camera_DeviceType_get(self) -> IEnumerationT_DeviceTypeEnums

> **Parameters self** (*Spinnaker::Camera **) –

DeviceUptime
Camera_DeviceUptime_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

DeviceUserID
    Camera_DeviceUserID_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera* *) –

DeviceVendorName
    Camera_DeviceVendorName_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera* *) –

DeviceVersion
    Camera_DeviceVersion_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderDivider
    Camera_EncoderDivider_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderMode
    Camera_EncoderMode_get(self) -> IEnumerationT_EncoderModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderOutputMode
    Camera_EncoderOutputMode_get(self) -> IEnumerationT_EncoderOutputModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderReset
    Camera_EncoderReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderResetActivation
    Camera_EncoderResetActivation_get(self) -> IEnumerationT_EncoderResetActivationEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderResetSource
    Camera_EncoderResetSource_get(self) -> IEnumerationT_EncoderResetSourceEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderSelector
    Camera_EncoderSelector_get(self) -> IEnumerationT_EncoderSelectorEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderSourceA
    Camera_EncoderSourceA_get(self) -> IEnumerationT_EncoderSourceAEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderSourceB
    Camera_EncoderSourceB_get(self) -> IEnumerationT_EncoderSourceBEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderStatus
    Camera_EncoderStatus_get(self) -> IEnumerationT_EncoderStatusEnums

> **Parameters self** (*Spinnaker::Camera* *) –

EncoderTimeout
Camera_EncoderTimeout_get(self) -> IFloat

    Parameters self (*Spinnaker::Camera **) –

EncoderValue
Camera_EncoderValue_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EncoderValueAtReset
Camera_EncoderValueAtReset_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EnumerationCount
Camera_EnumerationCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionEnd
Camera_EventAcquisitionEnd_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionEndFrameID
Camera_EventAcquisitionEndFrameID_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionEndTimestamp
Camera_EventAcquisitionEndTimestamp_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionError
Camera_EventAcquisitionError_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionErrorFrameID
Camera_EventAcquisitionErrorFrameID_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionErrorTimestamp
Camera_EventAcquisitionErrorTimestamp_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionStart
Camera_EventAcquisitionStart_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionStartFrameID
Camera_EventAcquisitionStartFrameID_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionStartTimestamp
Camera_EventAcquisitionStartTimestamp_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

EventAcquisitionTransferEnd
Camera_EventAcquisitionTransferEnd_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTransferEndFrameID
    Camera_EventAcquisitionTransferEndFrameID_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTransferEndTimestamp
    Camera_EventAcquisitionTransferEndTimestamp_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTransferStart
    Camera_EventAcquisitionTransferStart_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTransferStartFrameID
    Camera_EventAcquisitionTransferStartFrameID_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTransferStartTimestamp
    Camera_EventAcquisitionTransferStartTimestamp_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTrigger
    Camera_EventAcquisitionTrigger_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTriggerFrameID
    Camera_EventAcquisitionTriggerFrameID_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventAcquisitionTriggerTimestamp
    Camera_EventAcquisitionTriggerTimestamp_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventActionLate
    Camera_EventActionLate_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventActionLateFrameID
    Camera_EventActionLateFrameID_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventActionLateTimestamp
    Camera_EventActionLateTimestamp_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventCounter0End
    Camera_EventCounter0End_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventCounter0EndFrameID
    Camera_EventCounter0EndFrameID_get(self) -> IInteger

           Parameters **self** (*Spinnaker::Camera* *) –

EventCounter0EndTimestamp
 Camera_EventCounter0EndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter0Start
 Camera_EventCounter0Start_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter0StartFrameID
 Camera_EventCounter0StartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter0StartTimestamp
 Camera_EventCounter0StartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1End
 Camera_EventCounter1End_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1EndFrameID
 Camera_EventCounter1EndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1EndTimestamp
 Camera_EventCounter1EndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1Start
 Camera_EventCounter1Start_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1StartFrameID
 Camera_EventCounter1StartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventCounter1StartTimestamp
 Camera_EventCounter1StartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventEncoder0Restarted
 Camera_EventEncoder0Restarted_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventEncoder0RestartedFrameID
 Camera_EventEncoder0RestartedFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventEncoder0RestartedTimestamp
 Camera_EventEncoder0RestartedTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventEncoder0Stopped
 Camera_EventEncoder0Stopped_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0StoppedFrameID
: Camera_EventEncoder0StoppedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder0StoppedTimestamp
: Camera_EventEncoder0StoppedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1Restarted
: Camera_EventEncoder1Restarted_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1RestartedFrameID
: Camera_EventEncoder1RestartedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1RestartedTimestamp
: Camera_EventEncoder1RestartedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1Stopped
: Camera_EventEncoder1Stopped_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1StoppedFrameID
: Camera_EventEncoder1StoppedFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventEncoder1StoppedTimestamp
: Camera_EventEncoder1StoppedTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventError
: Camera_EventError_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventErrorCode
: Camera_EventErrorCode_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventErrorFrameID
: Camera_EventErrorFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventErrorTimestamp
: Camera_EventErrorTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventExposureEnd
: Camera_EventExposureEnd_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventExposureEndFrameID
Camera_EventExposureEndFrameID_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventExposureEndTimestamp
Camera_EventExposureEndTimestamp_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventExposureStart
Camera_EventExposureStart_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventExposureStartFrameID
Camera_EventExposureStartFrameID_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventExposureStartTimestamp
Camera_EventExposureStartTimestamp_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstEnd
Camera_EventFrameBurstEnd_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstEndFrameID
Camera_EventFrameBurstEndFrameID_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstEndTimestamp
Camera_EventFrameBurstEndTimestamp_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstStart
Camera_EventFrameBurstStart_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstStartFrameID
Camera_EventFrameBurstStartFrameID_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameBurstStartTimestamp
Camera_EventFrameBurstStartTimestamp_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameEnd
Camera_EventFrameEnd_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameEndFrameID
Camera_EventFrameEndFrameID_get(self) -> IInteger

  Parameters self (*Spinnaker::Camera **) –

EventFrameEndTimestamp
Camera_EventFrameEndTimestamp_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameStart
> Camera_EventFrameStart_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameStartFrameID
> Camera_EventFrameStartFrameID_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameStartTimestamp
> Camera_EventFrameStartTimestamp_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferEnd
> Camera_EventFrameTransferEnd_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferEndFrameID
> Camera_EventFrameTransferEndFrameID_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferEndTimestamp
> Camera_EventFrameTransferEndTimestamp_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferStart
> Camera_EventFrameTransferStart_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferStartFrameID
> Camera_EventFrameTransferStartFrameID_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTransferStartTimestamp
> Camera_EventFrameTransferStartTimestamp_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTrigger
> Camera_EventFrameTrigger_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTriggerFrameID
> Camera_EventFrameTriggerFrameID_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventFrameTriggerTimestamp
> Camera_EventFrameTriggerTimestamp_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventLine0AnyEdge
> Camera_EventLine0AnyEdge_get(self) -> IInteger

>> Parameters **self** (*Spinnaker::Camera **) –

EventLine0AnyEdgeFrameID
    Camera_EventLine0AnyEdgeFrameID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0AnyEdgeTimestamp
    Camera_EventLine0AnyEdgeTimestamp_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdge
    Camera_EventLine0FallingEdge_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdgeFrameID
    Camera_EventLine0FallingEdgeFrameID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0FallingEdgeTimestamp
    Camera_EventLine0FallingEdgeTimestamp_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdge
    Camera_EventLine0RisingEdge_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdgeFrameID
    Camera_EventLine0RisingEdgeFrameID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine0RisingEdgeTimestamp
    Camera_EventLine0RisingEdgeTimestamp_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1AnyEdge
    Camera_EventLine1AnyEdge_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1AnyEdgeFrameID
    Camera_EventLine1AnyEdgeFrameID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1AnyEdgeTimestamp
    Camera_EventLine1AnyEdgeTimestamp_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1FallingEdge
    Camera_EventLine1FallingEdge_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1FallingEdgeFrameID
    Camera_EventLine1FallingEdgeFrameID_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

EventLine1FallingEdgeTimestamp
    Camera_EventLine1FallingEdgeTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdge
Camera_EventLine1RisingEdge_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdgeFrameID
Camera_EventLine1RisingEdgeFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLine1RisingEdgeTimestamp
Camera_EventLine1RisingEdgeTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChange
Camera_EventLinkSpeedChange_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChangeFrameID
Camera_EventLinkSpeedChangeFrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkSpeedChangeTimestamp
Camera_EventLinkSpeedChangeTimestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0
Camera_EventLinkTrigger0_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0FrameID
Camera_EventLinkTrigger0FrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger0Timestamp
Camera_EventLinkTrigger0Timestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger1
Camera_EventLinkTrigger1_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger1FrameID
Camera_EventLinkTrigger1FrameID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventLinkTrigger1Timestamp
Camera_EventLinkTrigger1Timestamp_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

EventNotification
Camera_EventNotification_get(self) -> IEnumerationT_EventNotificationEnums

Parameters **self** (*Spinnaker::Camera **) –

EventSelector
> Camera_EventSelector_get(self) -> IEnumerationT_EventSelectorEnums

> > **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChange
> Camera_EventSequencerSetChange_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChangeFrameID
> Camera_EventSequencerSetChangeFrameID_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSequencerSetChangeTimestamp
> Camera_EventSequencerSetChangeTimestamp_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSerialData
> Camera_EventSerialData_get(self) -> IString

> > **Parameters self** (*Spinnaker::Camera **) –

EventSerialDataLength
> Camera_EventSerialDataLength_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSerialPortReceive
> Camera_EventSerialPortReceive_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSerialPortReceiveTimestamp
> Camera_EventSerialPortReceiveTimestamp_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventSerialReceiveOverflow
> Camera_EventSerialReceiveOverflow_get(self) -> IBoolean

> > **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEnd
> Camera_EventStream0TransferBlockEnd_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEndFrameID
> Camera_EventStream0TransferBlockEndFrameID_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockEndTimestamp
> Camera_EventStream0TransferBlockEndTimestamp_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStart
> Camera_EventStream0TransferBlockStart_get(self) -> IInteger

> > **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStartFrameID
> Camera_EventStream0TransferBlockStartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockStartTimestamp
  Camera_EventStream0TransferBlockStartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTrigger
  Camera_EventStream0TransferBlockTrigger_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTriggerFrameID
  Camera_EventStream0TransferBlockTriggerFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBlockTriggerTimestamp
  Camera_EventStream0TransferBlockTriggerTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEnd
  Camera_EventStream0TransferBurstEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEndFrameID
  Camera_EventStream0TransferBurstEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstEndTimestamp
  Camera_EventStream0TransferBurstEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStart
  Camera_EventStream0TransferBurstStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStartFrameID
  Camera_EventStream0TransferBurstStartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferBurstStartTimestamp
  Camera_EventStream0TransferBurstStartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEnd
  Camera_EventStream0TransferEnd_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEndFrameID
  Camera_EventStream0TransferEndFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferEndTimestamp
  Camera_EventStream0TransferEndTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflow
Camera_EventStream0TransferOverflow_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflowFrameID
Camera_EventStream0TransferOverflowFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferOverflowTimestamp
Camera_EventStream0TransferOverflowTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPause
Camera_EventStream0TransferPause_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPauseFrameID
Camera_EventStream0TransferPauseFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferPauseTimestamp
Camera_EventStream0TransferPauseTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResume
Camera_EventStream0TransferResume_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResumeFrameID
Camera_EventStream0TransferResumeFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferResumeTimestamp
Camera_EventStream0TransferResumeTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferStart
Camera_EventStream0TransferStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferStartFrameID
Camera_EventStream0TransferStartFrameID_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventStream0TransferStartTimestamp
Camera_EventStream0TransferStartTimestamp_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventTest
Camera_EventTest_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

EventTestTimestamp
Camera_EventTestTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0End
Camera_EventTimer0End_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0EndFrameID
Camera_EventTimer0EndFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0EndTimestamp
Camera_EventTimer0EndTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0Start
Camera_EventTimer0Start_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0StartFrameID
Camera_EventTimer0StartFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer0StartTimestamp
Camera_EventTimer0StartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1End
Camera_EventTimer1End_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1EndFrameID
Camera_EventTimer1EndFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1EndTimestamp
Camera_EventTimer1EndTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1Start
Camera_EventTimer1Start_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1StartFrameID
Camera_EventTimer1StartFrameID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

EventTimer1StartTimestamp
Camera_EventTimer1StartTimestamp_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

ExposureActiveMode
Camera_ExposureActiveMode_get(self) -> IEnumerationT_ExposureActiveModeEnums

> Parameters **self** (*Spinnaker::Camera **) –

ExposureAuto
Camera_ExposureAuto_get(self) -> IEnumerationT_ExposureAutoEnums

> **Parameters self** (*Spinnaker::Camera* *) –

ExposureMode
Camera_ExposureMode_get(self) -> IEnumerationT_ExposureModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

ExposureTime
Camera_ExposureTime_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera* *) –

ExposureTimeMode
Camera_ExposureTimeMode_get(self) -> IEnumerationT_ExposureTimeModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

ExposureTimeSelector
Camera_ExposureTimeSelector_get(self) -> IEnumerationT_ExposureTimeSelectorEnums

> **Parameters self** (*Spinnaker::Camera* *) –

FactoryReset
Camera_FactoryReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserGain
Camera_FfcUserGain_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserGainAll
Camera_FfcUserGainAll_get(self) -> IRegister

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserOffset
Camera_FfcUserOffset_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserOffsetAll
Camera_FfcUserOffsetAll_get(self) -> IRegister

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserTableReset
Camera_FfcUserTableReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserTableSave
Camera_FfcUserTableSave_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera* *) –

FfcUserTableXCoordinate
Camera_FfcUserTableXCoordinate_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

FileAccessBuffer
Camera_FileAccessBuffer_get(self) -> IRegister

> Parameters **self** (*Spinnaker::Camera* *) –

FileAccessLength
Camera_FileAccessLength_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

FileAccessOffset
Camera_FileAccessOffset_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

FileOpenMode
Camera_FileOpenMode_get(self) -> IEnumerationT_FileOpenModeEnums

> Parameters **self** (*Spinnaker::Camera* *) –

FileOperationExecute
Camera_FileOperationExecute_get(self) -> ICommand

> Parameters **self** (*Spinnaker::Camera* *) –

FileOperationResult
Camera_FileOperationResult_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

FileOperationSelector
Camera_FileOperationSelector_get(self) -> IEnumerationT_FileOperationSelectorEnums

> Parameters **self** (*Spinnaker::Camera* *) –

FileOperationStatus
Camera_FileOperationStatus_get(self) -> IEnumerationT_FileOperationStatusEnums

> Parameters **self** (*Spinnaker::Camera* *) –

FileSelector
Camera_FileSelector_get(self) -> IEnumerationT_FileSelectorEnums

> Parameters **self** (*Spinnaker::Camera* *) –

FileSize
Camera_FileSize_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera* *) –

Gain
Camera_Gain_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera* *) –

GainAuto
Camera_GainAuto_get(self) -> IEnumerationT_GainAutoEnums

> Parameters **self** (*Spinnaker::Camera* *) –

GainAutoBalance
Camera_GainAutoBalance_get(self) -> IEnumerationT_GainAutoBalanceEnums

> Parameters **self** (*Spinnaker::Camera* *) –

GainSelector
Camera_GainSelector_get(self) -> IEnumerationT_GainSelectorEnums

> Parameters **self** (*Spinnaker::Camera* *) –

Gamma
   Camera_Gamma_get(self) -> IFloat

   > **Parameters self** (*Spinnaker::Camera* *) –

GammaEnable
   Camera_GammaEnable_get(self) -> IBoolean

   > **Parameters self** (*Spinnaker::Camera* *) –

GevActiveLinkCount
   Camera_GevActiveLinkCount_get(self) -> IInteger

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCCP
   Camera_GevCCP_get(self) -> IEnumerationT_GevCCPEnums

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentDefaultGateway
   Camera_GevCurrentDefaultGateway_get(self) -> IInteger

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentIPAddress
   Camera_GevCurrentIPAddress_get(self) -> IInteger

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentIPConfigurationDHCP
   Camera_GevCurrentIPConfigurationDHCP_get(self) -> IBoolean

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentIPConfigurationLLA
   Camera_GevCurrentIPConfigurationLLA_get(self) -> IBoolean

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentIPConfigurationPersistentIP
   Camera_GevCurrentIPConfigurationPersistentIP_get(self) -> IBoolean

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentPhysicalLinkConfiguration
   Camera_GevCurrentPhysicalLinkConfiguration_get(self) -> IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums

   > **Parameters self** (*Spinnaker::Camera* *) –

GevCurrentSubnetMask
   Camera_GevCurrentSubnetMask_get(self) -> IInteger

   > **Parameters self** (*Spinnaker::Camera* *) –

GevDiscoveryAckDelay
   Camera_GevDiscoveryAckDelay_get(self) -> IInteger

   > **Parameters self** (*Spinnaker::Camera* *) –

GevFirstURL
   Camera_GevFirstURL_get(self) -> IString

   > **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPExtendedStatusCodes
Camera_GevGVCPExtendedStatusCodes_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPExtendedStatusCodesSelector
Camera_GevGVCPExtendedStatusCodesSelector_get(self) -> IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPHeartbeatDisable
Camera_GevGVCPHeartbeatDisable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPPendingAck
Camera_GevGVCPPendingAck_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

GevGVCPPendingTimeout
Camera_GevGVCPPendingTimeout_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

GevGVSPExtendedIDMode
Camera_GevGVSPExtendedIDMode_get(self) -> IEnumerationT_GevGVSPExtendedIDModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevHeartbeatTimeout
Camera_GevHeartbeatTimeout_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588
Camera_GevIEEE1588_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588ClockAccuracy
Camera_GevIEEE1588ClockAccuracy_get(self) -> IEnumerationT_GevIEEE1588ClockAccuracyEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588Mode
Camera_GevIEEE1588Mode_get(self) -> IEnumerationT_GevIEEE1588ModeEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevIEEE1588Status
Camera_GevIEEE1588Status_get(self) -> IEnumerationT_GevIEEE1588StatusEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevIPConfigurationStatus
Camera_GevIPConfigurationStatus_get(self) -> IEnumerationT_GevIPConfigurationStatusEnums

> **Parameters self** (*Spinnaker::Camera* *) –

GevInterfaceSelector
Camera_GevInterfaceSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera* *) –

GevMACAddress
Camera_GevMACAddress_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevMCDA
Camera_GevMCDA_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevMCPHostPort
Camera_GevMCPHostPort_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevMCRC
Camera_GevMCRC_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevMCSP
Camera_GevMCSP_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevMCTT
Camera_GevMCTT_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevNumberOfInterfaces
Camera_GevNumberOfInterfaces_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPAUSEFrameReception
Camera_GevPAUSEFrameReception_get(self) -> IBoolean

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPAUSEFrameTransmission
Camera_GevPAUSEFrameTransmission_get(self) -> IBoolean

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPersistentDefaultGateway
Camera_GevPersistentDefaultGateway_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPersistentIPAddress
Camera_GevPersistentIPAddress_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPersistentSubnetMask
Camera_GevPersistentSubnetMask_get(self) -> IInteger

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPhysicalLinkConfiguration
Camera_GevPhysicalLinkConfiguration_get(self) -> IEnumerationT_GevPhysicalLinkConfigurationEnums

>>> **Parameters self** (*Spinnaker::Camera **) –

GevPrimaryApplicationIPAddress
Camera_GevPrimaryApplicationIPAddress_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevPrimaryApplicationSocket
    Camera_GevPrimaryApplicationSocket_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevPrimaryApplicationSwitchoverKey
    Camera_GevPrimaryApplicationSwitchoverKey_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCCFGAllInTransmission
    Camera_GevSCCFGAllInTransmission_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCCFGExtendedChunkData
    Camera_GevSCCFGExtendedChunkData_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCCFGPacketResendDestination
    Camera_GevSCCFGPacketResendDestination_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCCFGUnconditionalStreaming
    Camera_GevSCCFGUnconditionalStreaming_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCDA
    Camera_GevSCDA_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPD
    Camera_GevSCPD_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPDirection
    Camera_GevSCPDirection_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPHostPort
    Camera_GevSCPHostPort_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPInterfaceIndex
    Camera_GevSCPInterfaceIndex_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPSBigEndian
    Camera_GevSCPSBigEndian_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPSDoNotFragment
    Camera_GevSCPSDoNotFragment_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPSFireTestPacket
Camera_GevSCPSFireTestPacket_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCPSPacketSize
Camera_GevSCPSPacketSize_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCSP
Camera_GevSCSP_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCZoneConfigurationLock
Camera_GevSCZoneConfigurationLock_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSCZoneCount
Camera_GevSCZoneCount_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSCZoneDirectionAll
Camera_GevSCZoneDirectionAll_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSecondURL
Camera_GevSecondURL_get(self) -> IString

> Parameters **self** (*Spinnaker::Camera **) –

GevStreamChannelSelector
Camera_GevStreamChannelSelector_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GevSupportedOption
Camera_GevSupportedOption_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

GevSupportedOptionSelector
Camera_GevSupportedOptionSelector_get(self) -> IEnumerationT_GevSupportedOptionSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

GevTimestampTickFrequency
Camera_GevTimestampTickFrequency_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

GuiXmlManifestAddress
Camera_GuiXmlManifestAddress_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

Height
Camera_Height_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

HeightMax
Camera_HeightMax_get(self) -> IInteger

      Parameters self (*Spinnaker::Camera **) –

ImageComponentEnable
    Camera_ImageComponentEnable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

ImageComponentSelector
    Camera_ImageComponentSelector_get(self) -> IEnumerationT_ImageComponentSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

ImageCompressionBitrate
    Camera_ImageCompressionBitrate_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

ImageCompressionJPEGFormatOption
    Camera_ImageCompressionJPEGFormatOption_get(self) -> IEnumerationT_ImageCompressionJPEGFormatOptionEnums

        Parameters self (*Spinnaker::Camera **) –

ImageCompressionMode
    Camera_ImageCompressionMode_get(self) -> IEnumerationT_ImageCompressionModeEnums

        Parameters self (*Spinnaker::Camera **) –

ImageCompressionQuality
    Camera_ImageCompressionQuality_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

ImageCompressionRateOption
    Camera_ImageCompressionRateOption_get(self) -> IEnumerationT_ImageCompressionRateOptionEnums

        Parameters self (*Spinnaker::Camera **) –

Init (*self*)

        Parameters self (*Spinnaker::Camera **) –

    void Spinnaker::Camera::Init()

IspEnable
    Camera_IspEnable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

LUTEnable
    Camera_LUTEnable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

LUTIndex
    Camera_LUTIndex_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

LUTSelector
    Camera_LUTSelector_get(self) -> IEnumerationT_LUTSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

LUTValue
    Camera_LUTValue_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

LUTValueAll
Camera_LUTValueAll_get(self) -> IRegister

Parameters **self** (*Spinnaker::Camera **) –

LineFilterWidth
Camera_LineFilterWidth_get(self) -> IFloat

Parameters **self** (*Spinnaker::Camera **) –

LineFormat
Camera_LineFormat_get(self) -> IEnumerationT_LineFormatEnums

Parameters **self** (*Spinnaker::Camera **) –

LineInputFilterSelector
Camera_LineInputFilterSelector_get(self) -> IEnumerationT_LineInputFilterSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

LineInverter
Camera_LineInverter_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera **) –

LineMode
Camera_LineMode_get(self) -> IEnumerationT_LineModeEnums

Parameters **self** (*Spinnaker::Camera **) –

LinePitch
Camera_LinePitch_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

LineSelector
Camera_LineSelector_get(self) -> IEnumerationT_LineSelectorEnums

Parameters **self** (*Spinnaker::Camera **) –

LineSource
Camera_LineSource_get(self) -> IEnumerationT_LineSourceEnums

Parameters **self** (*Spinnaker::Camera **) –

LineStatus
Camera_LineStatus_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera **) –

LineStatusAll
Camera_LineStatusAll_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

LinkErrorCount
Camera_LinkErrorCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

LinkRecoveryCount
Camera_LinkRecoveryCount_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera **) –

LinkUptime
  Camera_LinkUptime_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTInputActivation
  Camera_LogicBlockLUTInputActivation_get(self) -> IEnumerationT_LogicBlockLUTInputActivationEnums

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTInputSelector
  Camera_LogicBlockLUTInputSelector_get(self) -> IEnumerationT_LogicBlockLUTInputSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTInputSource
  Camera_LogicBlockLUTInputSource_get(self) -> IEnumerationT_LogicBlockLUTInputSourceEnums

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTOutputValue
  Camera_LogicBlockLUTOutputValue_get(self) -> IBoolean

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTOutputValueAll
  Camera_LogicBlockLUTOutputValueAll_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTRowIndex
  Camera_LogicBlockLUTRowIndex_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

LogicBlockLUTSelector
  Camera_LogicBlockLUTSelector_get(self) -> IEnumerationT_LogicBlockLUTSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

LogicBlockSelector
  Camera_LogicBlockSelector_get(self) -> IEnumerationT_LogicBlockSelectorEnums

    Parameters self (*Spinnaker::Camera **) –

MaxDeviceResetTime
  Camera_MaxDeviceResetTime_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

OffsetX
  Camera_OffsetX_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

OffsetY
  Camera_OffsetY_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

PacketResendRequestCount
  Camera_PacketResendRequestCount_get(self) -> IInteger

    Parameters self (*Spinnaker::Camera **) –

PayloadSize
  Camera_PayloadSize_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelColorFilter
Camera_PixelColorFilter_get(self) -> IEnumerationT_PixelColorFilterEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelDynamicRangeMax
Camera_PixelDynamicRangeMax_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelDynamicRangeMin
Camera_PixelDynamicRangeMin_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormat
Camera_PixelFormat_get(self) -> IEnumerationT_PixelFormatEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormatInfoID
Camera_PixelFormatInfoID_get(self) -> IInteger

> Parameters **self** (*Spinnaker::Camera **) –

PixelFormatInfoSelector
Camera_PixelFormatInfoSelector_get(self) -> IEnumerationT_PixelFormatInfoSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

PixelSize
Camera_PixelSize_get(self) -> IEnumerationT_PixelSizeEnums

> Parameters **self** (*Spinnaker::Camera **) –

PowerSupplyCurrent
Camera_PowerSupplyCurrent_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

PowerSupplyVoltage
Camera_PowerSupplyVoltage_get(self) -> IFloat

> Parameters **self** (*Spinnaker::Camera **) –

RegionDestination
Camera_RegionDestination_get(self) -> IEnumerationT_RegionDestinationEnums

> Parameters **self** (*Spinnaker::Camera **) –

RegionMode
Camera_RegionMode_get(self) -> IEnumerationT_RegionModeEnums

> Parameters **self** (*Spinnaker::Camera **) –

RegionSelector
Camera_RegionSelector_get(self) -> IEnumerationT_RegionSelectorEnums

> Parameters **self** (*Spinnaker::Camera **) –

ReverseX
Camera_ReverseX_get(self) -> IBoolean

> Parameters **self** (*Spinnaker::Camera **) –

ReverseY
Camera_ReverseY_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

RgbTransformLightSource
Camera_RgbTransformLightSource_get(self) -> IEnumerationT_RgbTransformLightSourceEnums

> **Parameters self** (*Spinnaker::Camera **) –

Saturation
Camera_Saturation_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

SaturationEnable
Camera_SaturationEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dAxisMax
Camera_Scan3dAxisMax_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dAxisMin
Camera_Scan3dAxisMin_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateOffset
Camera_Scan3dCoordinateOffset_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateReferenceSelector
Camera_Scan3dCoordinateReferenceSelector_get(self) -> IEnumerationT_Scan3dCoordinateReferenceSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateReferenceValue
Camera_Scan3dCoordinateReferenceValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateScale
Camera_Scan3dCoordinateScale_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateSelector
Camera_Scan3dCoordinateSelector_get(self) -> IEnumerationT_Scan3dCoordinateSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateSystem
Camera_Scan3dCoordinateSystem_get(self) -> IEnumerationT_Scan3dCoordinateSystemEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateSystemReference
Camera_Scan3dCoordinateSystemReference_get(self) -> IEnumerationT_Scan3dCoordinateSystemReferenceEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dCoordinateTransformSelector
Camera_Scan3dCoordinateTransformSelector_get(self) -> IEnumerationT_Scan3dCoordinateTransformSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dDistanceUnit
Camera_Scan3dDistanceUnit_get(self) -> IEnumerationT_Scan3dDistanceUnitEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dInvalidDataFlag
Camera_Scan3dInvalidDataFlag_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dInvalidDataValue
Camera_Scan3dInvalidDataValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dOutputMode
Camera_Scan3dOutputMode_get(self) -> IEnumerationT_Scan3dOutputModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

Scan3dTransformValue
Camera_Scan3dTransformValue_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

SensorDescription
Camera_SensorDescription_get(self) -> IString

> **Parameters self** (*Spinnaker::Camera **) –

SensorDigitizationTaps
Camera_SensorDigitizationTaps_get(self) -> IEnumerationT_SensorDigitizationTapsEnums

> **Parameters self** (*Spinnaker::Camera **) –

SensorHeight
Camera_SensorHeight_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SensorShutterMode
Camera_SensorShutterMode_get(self) -> IEnumerationT_SensorShutterModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

SensorTaps
Camera_SensorTaps_get(self) -> IEnumerationT_SensorTapsEnums

> **Parameters self** (*Spinnaker::Camera **) –

SensorWidth
Camera_SensorWidth_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationMode
Camera_SequencerConfigurationMode_get(self) -> IEnumerationT_SequencerConfigurationModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationReset
Camera_SequencerConfigurationReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerConfigurationValid
Camera_SequencerConfigurationValid_get(self) -> IEnumerationT_SequencerConfigurationValidEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerFeatureEnable
Camera_SequencerFeatureEnable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

SequencerFeatureSelector
Camera_SequencerFeatureSelector_get(self) -> IEnumerationT_SequencerFeatureSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerMode
Camera_SequencerMode_get(self) -> IEnumerationT_SequencerModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerPathSelector
Camera_SequencerPathSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetActive
Camera_SequencerSetActive_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetLoad
Camera_SequencerSetLoad_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetNext
Camera_SequencerSetNext_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetSave
Camera_SequencerSetSave_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetSelector
Camera_SequencerSetSelector_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetStart
Camera_SequencerSetStart_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

SequencerSetValid
Camera_SequencerSetValid_get(self) -> IEnumerationT_SequencerSetValidEnums

> **Parameters self** (*Spinnaker::Camera **) –

SequencerTriggerActivation
Camera_SequencerTriggerActivation_get(self) -> IEnumerationT_SequencerTriggerActivationEnums

Parameters **self** (*Spinnaker::Camera **) –

SequencerTriggerSource
    Camera_SequencerTriggerSource_get(self) -> IEnumerationT_SequencerTriggerSourceEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortBaudRate
    Camera_SerialPortBaudRate_get(self) -> IEnumerationT_SerialPortBaudRateEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortDataBits
    Camera_SerialPortDataBits_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortParity
    Camera_SerialPortParity_get(self) -> IEnumerationT_SerialPortParityEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortSelector
    Camera_SerialPortSelector_get(self) -> IEnumerationT_SerialPortSelectorEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortSource
    Camera_SerialPortSource_get(self) -> IEnumerationT_SerialPortSourceEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialPortStopBits
    Camera_SerialPortStopBits_get(self) -> IEnumerationT_SerialPortStopBitsEnums

    Parameters **self** (*Spinnaker::Camera **) –

SerialReceiveFramingErrorCount
    Camera_SerialReceiveFramingErrorCount_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialReceiveParityErrorCount
    Camera_SerialReceiveParityErrorCount_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialReceiveQueueClear
    Camera_SerialReceiveQueueClear_get(self) -> ICommand

    Parameters **self** (*Spinnaker::Camera **) –

SerialReceiveQueueCurrentCharacterCount
    Camera_SerialReceiveQueueCurrentCharacterCount_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialReceiveQueueMaxCharacterCount
    Camera_SerialReceiveQueueMaxCharacterCount_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialTransmitQueueCurrentCharacterCount
    Camera_SerialTransmitQueueCurrentCharacterCount_get(self) -> IInteger

    Parameters **self** (*Spinnaker::Camera **) –

SerialTransmitQueueMaxCharacterCount
    Camera_SerialTransmitQueueMaxCharacterCount_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

Sharpening
    Camera_Sharpening_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

SharpeningAuto
    Camera_SharpeningAuto_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

SharpeningEnable
    Camera_SharpeningEnable_get(self) -> IBoolean

        Parameters self (*Spinnaker::Camera **) –

SharpeningThreshold
    Camera_SharpeningThreshold_get(self) -> IFloat

        Parameters self (*Spinnaker::Camera **) –

SoftwareSignalPulse
    Camera_SoftwareSignalPulse_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

SoftwareSignalSelector
    Camera_SoftwareSignalSelector_get(self) -> IEnumerationT_SoftwareSignalSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

SourceCount
    Camera_SourceCount_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

SourceSelector
    Camera_SourceSelector_get(self) -> IEnumerationT_SourceSelectorEnums

        Parameters self (*Spinnaker::Camera **) –

TLParamsLocked
    Camera_TLParamsLocked_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

Test0001
    Camera_Test0001_get(self) -> IInteger

        Parameters self (*Spinnaker::Camera **) –

TestEventGenerate
    Camera_TestEventGenerate_get(self) -> ICommand

        Parameters self (*Spinnaker::Camera **) –

TestPattern
    Camera_TestPattern_get(self) -> IEnumerationT_TestPatternEnums

        Parameters self (*Spinnaker::Camera **) –

TestPatternGeneratorSelector
    Camera_TestPatternGeneratorSelector_get(self) -> IEnumerationT_TestPatternGeneratorSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

TestPendingAck
>   Camera_TestPendingAck_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::Camera **) –

TimerDelay
>   Camera_TimerDelay_get(self) -> IFloat

>> **Parameters self** (*Spinnaker::Camera **) –

TimerDuration
>   Camera_TimerDuration_get(self) -> IFloat

>> **Parameters self** (*Spinnaker::Camera **) –

TimerReset
>   Camera_TimerReset_get(self) -> ICommand

>> **Parameters self** (*Spinnaker::Camera **) –

TimerSelector
>   Camera_TimerSelector_get(self) -> IEnumerationT_TimerSelectorEnums

>> **Parameters self** (*Spinnaker::Camera **) –

TimerStatus
>   Camera_TimerStatus_get(self) -> IEnumerationT_TimerStatusEnums

>> **Parameters self** (*Spinnaker::Camera **) –

TimerTriggerActivation
>   Camera_TimerTriggerActivation_get(self) -> IEnumerationT_TimerTriggerActivationEnums

>> **Parameters self** (*Spinnaker::Camera **) –

TimerTriggerSource
>   Camera_TimerTriggerSource_get(self) -> IEnumerationT_TimerTriggerSourceEnums

>> **Parameters self** (*Spinnaker::Camera **) –

TimerValue
>   Camera_TimerValue_get(self) -> IFloat

>> **Parameters self** (*Spinnaker::Camera **) –

Timestamp
>   Camera_Timestamp_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::Camera **) –

TimestampIncrement
>   Camera_TimestampIncrement_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::Camera **) –

TimestampLatch
>   Camera_TimestampLatch_get(self) -> ICommand

>> **Parameters self** (*Spinnaker::Camera **) –

TimestampLatchValue
>   Camera_TimestampLatchValue_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::Camera **) –

TimestampReset
Camera_TimestampReset_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferAbort
Camera_TransferAbort_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferBlockCount
Camera_TransferBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferBurstCount
Camera_TransferBurstCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferComponentSelector
Camera_TransferComponentSelector_get(self) -> IEnumerationT_TransferComponentSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferControlMode
Camera_TransferControlMode_get(self) -> IEnumerationT_TransferControlModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferOperationMode
Camera_TransferOperationMode_get(self) -> IEnumerationT_TransferOperationModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferPause
Camera_TransferPause_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueCurrentBlockCount
Camera_TransferQueueCurrentBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueMaxBlockCount
Camera_TransferQueueMaxBlockCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueMode
Camera_TransferQueueMode_get(self) -> IEnumerationT_TransferQueueModeEnums

> **Parameters self** (*Spinnaker::Camera **) –

TransferQueueOverflowCount
Camera_TransferQueueOverflowCount_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

TransferResume
Camera_TransferResume_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

TransferSelector
Camera_TransferSelector_get(self) -> IEnumerationT_TransferSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TransferStart
> Camera_TransferStart_get(self) -> ICommand

> > Parameters **self** (*Spinnaker::Camera **) –

TransferStatus
> Camera_TransferStatus_get(self) -> IBoolean

> > Parameters **self** (*Spinnaker::Camera **) –

TransferStatusSelector
> Camera_TransferStatusSelector_get(self) -> IEnumerationT_TransferStatusSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TransferStop
> Camera_TransferStop_get(self) -> ICommand

> > Parameters **self** (*Spinnaker::Camera **) –

TransferStreamChannel
> Camera_TransferStreamChannel_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerActivation
> Camera_TransferTriggerActivation_get(self) -> IEnumerationT_TransferTriggerActivationEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerMode
> Camera_TransferTriggerMode_get(self) -> IEnumerationT_TransferTriggerModeEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerSelector
> Camera_TransferTriggerSelector_get(self) -> IEnumerationT_TransferTriggerSelectorEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TransferTriggerSource
> Camera_TransferTriggerSource_get(self) -> IEnumerationT_TransferTriggerSourceEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TriggerActivation
> Camera_TriggerActivation_get(self) -> IEnumerationT_TriggerActivationEnums

> > Parameters **self** (*Spinnaker::Camera **) –

TriggerDelay
> Camera_TriggerDelay_get(self) -> IFloat

> > Parameters **self** (*Spinnaker::Camera **) –

TriggerDivider
> Camera_TriggerDivider_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::Camera **) –

TriggerEventTest
> Camera_TriggerEventTest_get(self) -> ICommand

> > Parameters **self** (*Spinnaker::Camera **) –

TriggerMode
   Camera_TriggerMode_get(self) -> IEnumerationT_TriggerModeEnums

   Parameters self (*Spinnaker::Camera **) –

TriggerMultiplier
   Camera_TriggerMultiplier_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

TriggerOverlap
   Camera_TriggerOverlap_get(self) -> IEnumerationT_TriggerOverlapEnums

   Parameters self (*Spinnaker::Camera **) –

TriggerSelector
   Camera_TriggerSelector_get(self) -> IEnumerationT_TriggerSelectorEnums

   Parameters self (*Spinnaker::Camera **) –

TriggerSoftware
   Camera_TriggerSoftware_get(self) -> ICommand

   Parameters self (*Spinnaker::Camera **) –

TriggerSource
   Camera_TriggerSource_get(self) -> IEnumerationT_TriggerSourceEnums

   Parameters self (*Spinnaker::Camera **) –

U3VAccessPrivilege
   Camera_U3VAccessPrivilege_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

U3VCPCapability
   Camera_U3VCPCapability_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

U3VCPEIRMAvailable
   Camera_U3VCPEIRMAvailable_get(self) -> IBoolean

   Parameters self (*Spinnaker::Camera **) –

U3VCPIIDC2Available
   Camera_U3VCPIIDC2Available_get(self) -> IBoolean

   Parameters self (*Spinnaker::Camera **) –

U3VCPSIRMAvailable
   Camera_U3VCPSIRMAvailable_get(self) -> IBoolean

   Parameters self (*Spinnaker::Camera **) –

U3VCurrentSpeed
   Camera_U3VCurrentSpeed_get(self) -> IEnumerationT_U3VCurrentSpeedEnums

   Parameters self (*Spinnaker::Camera **) –

U3VMaxAcknowledgeTransferLength
   Camera_U3VMaxAcknowledgeTransferLength_get(self) -> IInteger

   Parameters self (*Spinnaker::Camera **) –

U3VMaxCommandTransferLength
   Camera_U3VMaxCommandTransferLength_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

U3VMaxDeviceResponseTime
Camera_U3VMaxDeviceResponseTime_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

U3VMessageChannelID
Camera_U3VMessageChannelID_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

U3VNumberOfStreamChannels
Camera_U3VNumberOfStreamChannels_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

U3VVersionMajor
Camera_U3VVersionMajor_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

U3VVersionMinor
Camera_U3VVersionMinor_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

UserOutputSelector
Camera_UserOutputSelector_get(self) -> IEnumerationT_UserOutputSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

UserOutputValue
Camera_UserOutputValue_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

UserOutputValueAll
Camera_UserOutputValueAll_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

UserOutputValueAllMask
Camera_UserOutputValueAllMask_get(self) -> IInteger

Parameters **self** (*Spinnaker::Camera* *) –

UserSetDefault
Camera_UserSetDefault_get(self) -> IEnumerationT_UserSetDefaultEnums

Parameters **self** (*Spinnaker::Camera* *) –

UserSetFeatureEnable
Camera_UserSetFeatureEnable_get(self) -> IBoolean

Parameters **self** (*Spinnaker::Camera* *) –

UserSetFeatureSelector
Camera_UserSetFeatureSelector_get(self) -> IEnumerationT_UserSetFeatureSelectorEnums

Parameters **self** (*Spinnaker::Camera* *) –

UserSetLoad
Camera_UserSetLoad_get(self) -> ICommand

Parameters **self** (*Spinnaker::Camera* *) –

UserSetSave
Camera_UserSetSave_get(self) -> ICommand

> **Parameters self** (*Spinnaker::Camera **) –

UserSetSelector
Camera_UserSetSelector_get(self) -> IEnumerationT_UserSetSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

V3_3Enable
Camera_V3_3Enable_get(self) -> IBoolean

> **Parameters self** (*Spinnaker::Camera **) –

WhiteClip
Camera_WhiteClip_get(self) -> IFloat

> **Parameters self** (*Spinnaker::Camera **) –

WhiteClipSelector
Camera_WhiteClipSelector_get(self) -> IEnumerationT_WhiteClipSelectorEnums

> **Parameters self** (*Spinnaker::Camera **) –

Width
Camera_Width_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

WidthMax
Camera_WidthMax_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

aPAUSEMACCtrlFramesReceived
Camera_aPAUSEMACCtrlFramesReceived_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

aPAUSEMACCtrlFramesTransmitted
Camera_aPAUSEMACCtrlFramesTransmitted_get(self) -> IInteger

> **Parameters self** (*Spinnaker::Camera **) –

thisown
The membership flag

class PySpin.**CameraBase**(**args*, ***kwargs*)
Bases: object

The base class for the camera object.

C++ includes: CameraBase.h

BeginAcquisition(*self*)

> **Parameters self** (*Spinnaker::CameraBase **) –

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*)

> Parameters **self** (*Spinnaker::CameraBase **) –

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

> Parameters **self** (*Spinnaker::CameraBase **) –

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*)

> Parameters **self** (*Spinnaker::CameraBase **) –

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()

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

> Parameters **self** (*Spinnaker::CameraBase const **) –

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

GetGuiXml(*self*) → gcstring

> Parameters **self** (*Spinnaker::CameraBase const **) –

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*, *streamID=0*) → ImagePtr

> **Parameters**

- **grabTimeout** (*uint64_t*) –
- **streamID** (*uint64_t*) –
- **grabTimeout) -> ImagePtr** (*GetNextImage (self,*) –
- **grabTimeout** –
- **-> ImagePtr** (*GetNextImage (self)*) –
- **self** (*Spinnaker::CameraBase **) –

ImagePtr   Spinnaker::CameraBase::GetNextImage(uint64_t   grabTimeout=EVENT_TIMEOUT_INFINITE, uint64_t streamID=0)

Gets the next image that was received by the transport layer. This function will block indefinitely until an image arrives. Most cameras support one stream so the default streamID is 0 but if a camera supports multiple streams the user can input the streamID to select from which stream to grab images

See: Init()

See: BeginAcquisition()

See: EndAcquisition()

grabTimeout: a 64bit value that represents a timeout in milliseconds

streamID: The stream to grab the image.

pointer to an Image object

GetNodeMap (*self*) → INodeMap

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

unsigned int Spinnaker::CameraBase::GetNumDataStreams()

Returns the number of streams that a device supports.

The number of data streams

GetNumImagesInUse (*self*) → unsigned int

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase const **) –

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*) → INodeMap

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

Gets a reference to the node map that is generated from a GenICam XML file for the GenTL Stream module. The camera does not need to be initialized before acquiring this node map.

A reference to the INodeMap.

GetUniqueID (*self*) → gcstring

> **Parameters self** (*Spinnaker::CameraBase **) –

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)

Init (*self*)

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

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

> **Parameters self** (*Spinnaker::CameraBase const **) –

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

> **Parameters self** (*Spinnaker::CameraBase **) –

bool Spinnaker::CameraBase::IsValid()

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

If camera is valid or not

RegisterEvent(*self*, *evtToRegister*)

> **Parameters**
>
> > • **evtToRegister** (*Spinnaker::Event &*) –
> >
> > • **evtToRegister, eventName)** ([RegisterEvent](*self,*)) –
> >
> > • **evtToRegister** –
> >
> > • **eventName** (*Spinnaker::GenICam::gcstring const &*) –

void Spinnaker::CameraBase::RegisterEvent(Event &evtToRegister, const GenICam::gcstring &event-Name)

Registers a specific event for the camera

See: Init()

evtToRegister: The event to register for the camera

eventName: The event name to register

TLDevice

> CameraBase_TLDevice_get(self) -> TransportLayerDevice
>
> > **Parameters self** (*Spinnaker::CameraBase **) –

TLStream

> CameraBase_TLStream_get(self) -> TransportLayerStream
>
> > **Parameters self** (*Spinnaker::CameraBase **) –

UnregisterEvent(*self*, *evtToUnregister*)

> **Parameters evtToUnregister** (*Spinnaker::Event &*) –

void Spinnaker::CameraBase::UnregisterEvent(Event &evtToUnregister)

Unregisters an event for the camera Events should be unregistered first before calling camera DeInit(). Otherwise an exception will be thrown in the DeInit() call and require the user to unregister events before the camera can be re-initialized again.

See: DeInit()

evtToUnregister: The event to unregister from the camera

thisown

> The membership flag

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

Bases: object

Used to hold a list of camera objects.

C++ includes: CameraList.h

Append(*self*, *otherList*)

> **Parameters otherList** (`Spinnaker::CameraList &`) –

void Spinnaker::CameraList::Append(CameraList &otherList)

Appends a camera list to the current list.

otherList: The other list to append to this list

Clear (*self*)

> **Parameters self** (`Spinnaker::CameraList *`) –

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()

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

> **Parameters index** (`int`) –

CameraPtr Spinnaker::CameraList::GetByIndex(int index) const

Returns a pointer to a camera object at the "index".

index: The index at which to retrieve the camera object

A pointer to an camera object.

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

> **Parameters serialNumber** (`std::string`) –

CameraPtr Spinnaker::CameraList::GetBySerial(std::string serialNumber) const

Returns a pointer to a camera object with the specified serial number.

serialNumber: The serial number of the camera object to retrieve

A pointer to an camera object.

GetSize (*self*) → int

> **Parameters self** (`Spinnaker::CameraList const *`) –

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.

RemoveByIndex (*self*, *index*)

> **Parameters index** (`int`) –

void Spinnaker::CameraList::RemoveByIndex(int index)

Removes a camera at "index" and destroys its corresponding reference counted object.

index: The index at which to remove the Camera object

RemoveBySerial (*self*, *serialNumber*)

> **Parameters serialNumber** (`std::string`) –

void Spinnaker::CameraList::RemoveBySerial(std::string serialNumber)

Removes a camera using its serial number and destroys its corresponding reference counted object.

serialNumber: The serial number of the Camera object to remove

thisown
The membership flag

class PySpin.**CameraPtr**(**args*)
Bases: PySpin._SWIG_CamPtr

A reference tracked pointer to a camera object.

C++ includes: CameraPtr.h

thisown
The membership flag

class PySpin.**CategoryNode**(**args*, ***kwargs*)
Bases: *PySpin.ICategory*, *PySpin.ValueNode*

Interface for string properties.

C++ includes: CategoryNode.h

GetFeatures(*self*)

> **Parameters self** (*Spinnaker::GenApi::CategoryNode const **) –

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 **) –

virtual void Spinnaker::GenApi::CategoryNode::SetReference(INode *pBase)

overload SetReference for Value

thisown
The membership flag

class 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.

channel
ChannelStatistics_channel_get(self) -> int

>   **Parameters self** (*ChannelStatistics **) –

histogram
>   ChannelStatistics_histogram_get(self) -> PyObject *

>   **Parameters self** (*ChannelStatistics **) –

num_pixel_values
>   ChannelStatistics_num_pixel_values_get(self) -> unsigned int

>   **Parameters self** (*ChannelStatistics **) –

pixel_value_max
>   ChannelStatistics_pixel_value_max_get(self) -> unsigned int

>   **Parameters self** (*ChannelStatistics **) –

pixel_value_mean
>   ChannelStatistics_pixel_value_mean_get(self) -> float

>   **Parameters self** (*ChannelStatistics **) –

pixel_value_min
>   ChannelStatistics_pixel_value_min_get(self) -> unsigned int

>   **Parameters self** (*ChannelStatistics **) –

range_max
>   ChannelStatistics_range_max_get(self) -> unsigned int

>   **Parameters self** (*ChannelStatistics **) –

range_min
>   ChannelStatistics_range_min_get(self) -> unsigned int

>   **Parameters self** (*ChannelStatistics **) –

thisown
>   The membership flag

class PySpin.**ChunkData**(**args*)
>   Bases: *PySpin.IChunkData*

>   The chunk data which contains additional information about an image.

>   C++ includes: ChunkData.h

>   **GetBlackLevel**(*self*) → float64_t

>   >   **Parameters self** (*Spinnaker::ChunkData const **) –

>   >   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

>   >   **Parameters self** (*Spinnaker::ChunkData const **) –

>   **GetCounterValue**(*self*) → int64_t

>   >   **Parameters self** (*Spinnaker::ChunkData const **) –

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

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

GetEncoderValue (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetExposureTime (*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetExposureTime() const

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

GetFrameID (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetGain() const

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

GetHeight (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetHeight() const

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

GetImage (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetLinePitch (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetLinePitch() const

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

GetLineStatusAll (*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

GetOffsetX (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetOffsetX() const

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

GetOffsetY (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetOffsetY() const

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

GetPartSelector (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetScan3dCoordinateScale() const

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

GetScan3dInvalidDataValue(*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

float64_t Spinnaker::ChunkData::GetScan3dTransformValue() const

Description: Returns the transform value. Visibility: Expert

GetScanLineSelector(*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetSequencerSetActive() const

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

GetSerialDataLength(*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

GetStreamChannelID(*self*) → int64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetStreamChannelID() const

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

GetTimerValue(*self*) → float64_t

> **Parameters self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetTimestampLatchValue() const

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

GetTransferBlockID (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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

Parameters **self** (*Spinnaker::ChunkData const **) –

int64_t Spinnaker::ChunkData::GetTransferQueueCurrentBlockCount() const

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

GetWidth (*self*) → int64_t

Parameters **self** (*Spinnaker::ChunkData const **) –

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 Spinnaker::ChunkData::SetChunks(GenApi::INodeMap &pNodeMap)

thisown
The membership flag

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

Parameters

- **Peter** (*enum Spinnaker::GenApi::EAccessMode*) –

- **Paul** (*enum Spinnaker::GenApi::EAccessMode*) –

:param Combine(Peter, Paul) -> Spinnaker::GenApi::EVisibility:

Parameters

- **Peter** (*enum Spinnaker::GenApi::EVisibility*) –

- **Paul** (*enum Spinnaker::GenApi::EVisibility*) –

:param Combine(Peter, Paul) -> Spinnaker::GenApi::ECachingMode:

Parameters

- **Peter** (*enum Spinnaker::GenApi::ECachingMode*) –

- **Paul** (*enum Spinnaker::GenApi::ECachingMode*) –

ECachingMode Spinnaker::GenApi::Combine(ECachingMode Peter, ECachingMode Paul)

Computes which CachingMode results from a combination

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

    Bases: *PySpin.ICommand*, *PySpin.ValueNode*

    Interface for string properties.

    C++ includes: CommandNode.h

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

            Parameters

- **Verify** (*bool*) –
- **Execute(self)** –
- **self** (*Spinnaker::GenApi::CommandNode **) –

    virtual void Spinnaker::GenApi::CommandNode::Execute(bool Verify=true)

    Execute the command

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

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

            Parameters

- **Verify** (*bool*) –
- **-> bool** (*IsDone(self)*) –
- **self** (*Spinnaker::GenApi::CommandNode **) –

    virtual bool Spinnaker::GenApi::CommandNode::IsDone(bool Verify=true)

    Query whether the command is executed

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

    True if the Execute command has finished; false otherwise

    SetReference(*self*, *pBase*)

            Parameters pBase (*Spinnaker::GenApi::INode **) –

    virtual void Spinnaker::GenApi::CommandNode::SetReference(INode *pBase)

    overload SetReference for Value

    thisown
        The membership flag

PySpin.**DeregisterNodeCallback**(*f*)

        Parameters f (*NodeCallback &*) –

class PySpin.**DeviceEvent**

    Bases: *PySpin.IDeviceEvent*

    A handler to device events.

    C++ includes: DeviceEvent.h

    GetDeviceEventId(*self*) → uint64_t

> Parameters **self** (*Spinnaker::DeviceEvent const **) –

uint64_t Spinnaker::DeviceEvent::GetDeviceEventId() const

Get the ID of the device event.

The device event ID

GetDeviceEventName (*self*) → gcstring

> Parameters **self** (*Spinnaker::DeviceEvent const **) –

GenICam::gcstring Spinnaker::DeviceEvent::GetDeviceEventName() const

Get the name of the device event.

The device event name

OnDeviceEvent (*self*, *eventName*)

> Parameters **eventName** (*Spinnaker::GenICam::gcstring*) –

virtual void Spinnaker::DeviceEvent::OnDeviceEvent(Spinnaker::GenICam::gcstring eventName)=0

Device event callback.

eventName: The name of the event

thisown
> The membership flag

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

> Parameters **VariableName** (*Spinnaker::GenICam::gcstring const &*) –

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

Returns true if an environment variable exists

class 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*) –

thisown
> The membership flag

PySpin.**EAccessModeClass_FromString**(*ValueStr*, *pValue*) → bool

> **Parameters**
>
> > • **ValueStr**(*Spinnaker::GenICam::gcstring const &*) –
> >
> > • **pValue**(*Spinnaker::GenApi::EAccessMode ***) –

PySpin.**EAccessModeClass_ToString**(**args*)
> ToString(ValueStr, pValue)

> **Parameters**
>
> > • **ValueStr**(*Spinnaker::GenICam::gcstring &*) –
> >
> > • **pValue**(*Spinnaker::GenApi::EAccessMode ***) –
> >
> > • **-> gcstring**([EAccessModeClass_ToString(*Value*))) –
> >
> > • **Value**(*enum Spinnaker::GenApi::EAccessMode*) –

class 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*) –

> **thisown**
> > The membership flag

PySpin.**ECachingModeClass_FromString**(*ValueStr*, *pValue*) → bool

> **Parameters**
>
> > • **ValueStr**(*Spinnaker::GenICam::gcstring const &*) –
> >
> > • **pValue**(*Spinnaker::GenApi::ECachingMode ***) –

PySpin.**ECachingModeClass_ToString**(**args*)
> ToString(ValueStr, pValue)

> **Parameters**
>
> > • **ValueStr**(*Spinnaker::GenICam::gcstring &*) –
> >
> > • **pValue**(*Spinnaker::GenApi::ECachingMode ***) –
> >
> > • **-> gcstring**([ECachingModeClass_ToString(*Value*))) –

- **Value**(*enum Spinnaker::GenApi::ECachingMode*) –

class 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*) –

thisown

> The membership flag

PySpin.**EDisplayNotationClass_FromString**(*ValueStr*, *pValue*) → bool

> Parameters

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

PySpin.**EDisplayNotationClass_ToString**(**args*)

ToString(ValueStr, pValue)

> Parameters

> - **ValueStr**(*Spinnaker::GenICam::gcstring &*) –
> - **pValue**(*Spinnaker::GenApi::EDisplayNotation **) –
> - **-> gcstring**([EDisplayNotationClass_ToString*(Value)*](#)) –
> - **Value**(*enum Spinnaker::GenApi::EDisplayNotation*) –

class 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*) –

> **thisown**
> > The membership flag

PySpin.**EEndianessClass_FromString**(*ValueStr*, *pValue*) → bool

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

PySpin.**EEndianessClass_ToString**(**args*)
> ToString(ValueStr, pValue)

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

class PySpin.**EGenApiSchemaVersionClass**
> Bases: object

> helper class converting EGenApiSchemaVersion from and to string

> C++ includes: EnumClasses.h

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

> > Parameters
> >
> > > - **ValueStr** (*Spinnaker::GenICam::gcstring const &*) –
> > >
> > > - **pValue** (*Spinnaker::GenApi::EGenApiSchemaVersion **) –

> **static ToString**(*ValueStr*, *pValue*)

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

> **thisown**
> > The membership flag

PySpin.**EGenApiSchemaVersionClass_FromString**(*ValueStr*, *pValue*) → bool

> Parameters
>
> > - **ValueStr** (*Spinnaker::GenICam::gcstring const &*) –

> • **pValue** (*Spinnaker::GenApi::EGenApiSchemaVersion **) –

PySpin.**EGenApiSchemaVersionClass_ToString**(**args*)
　　ToString(ValueStr, pValue)

　　　　Parameters

> • **ValueStr** (*Spinnaker::GenICam::gcstring &*) –
>
> • **pValue** (*Spinnaker::GenApi::EGenApiSchemaVersion **) –
>
> • **-> gcstring** ([EGenApiSchemaVersionClass_ToString](*Value*)) –
>
> • **Value** (*enum Spinnaker::GenApi::EGenApiSchemaVersion*) –

class 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*) –

　　thisown
　　　　The membership flag

PySpin.**EInputDirectionClass_FromString**(*ValueStr*, *pValue*) → bool

　　　　Parameters

> • **ValueStr** (*Spinnaker::GenICam::gcstring const &*) –
>
> • **pValue** (*Spinnaker::GenApi::EInputDirection **) –

PySpin.**EInputDirectionClass_ToString**(**args*)
　　ToString(ValueStr, pValue)

　　　　Parameters

> • **ValueStr** (*Spinnaker::GenICam::gcstring &*) –
>
> • **pValue** (*Spinnaker::GenApi::EInputDirection **) –
>
> • **-> gcstring** ([EInputDirectionClass_ToString](*Value*)) –
>
> • **Value** (*enum Spinnaker::GenApi::EInputDirection*) –

class 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*)–

thisown
> The membership flag

PySpin.**ENameSpaceClass_FromString**(*ValueStr*, *pValue*) → bool

> Parameters

>> • **ValueStr**(*Spinnaker::GenICam::gcstring const &*)–

>> • **pValue**(*Spinnaker::GenApi::ENameSpace ★*)–

PySpin.**ENameSpaceClass_ToString**(*★args*)
> ToString(ValueStr, pValue)

> Parameters

>> • **ValueStr**(*Spinnaker::GenICam::gcstring &*)–

>> • **pValue**(*Spinnaker::GenApi::ENameSpace ★*)–

>> • **-> gcstring**(ENameSpaceClass_ToString*(Value)*)–

>> • **Value**(*enum Spinnaker::GenApi::ENameSpace*)–

class 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*) –

thisown
> The membership flag

PySpin.**ERepresentationClass_FromString** (*ValueStr*, *pValue*) → bool

> **Parameters**

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

- **pValue** (*Spinnaker::GenApi::ERepresentation **) –

PySpin.**ERepresentationClass_ToString** (**args*)
> ToString(ValueStr, pValue)

> **Parameters**

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

- **pValue** (*Spinnaker::GenApi::ERepresentation **) –

- **-> gcstring** ([ERepresentationClass_ToString](*Value*)) –

- **Value** (*enum Spinnaker::GenApi::ERepresentation*) –

class 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*) –

> **thisown**
> > The membership flag

PySpin.**ESignClass_FromString** (*ValueStr*, *pValue*) → bool

> **Parameters**

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

- **pValue** (*Spinnaker::GenApi::ESign **) –

PySpin.**ESignClass_ToString** (**args*)
> ToString(ValueStr, pValue)

> **Parameters**

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

- **pValue** (*Spinnaker::GenApi::ESign **) –

- **-> gcstring** ([ESignClass_ToString](Value)) –

- **Value** (*enum Spinnaker::GenApi::ESign*) –

class 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*) –

    thisown

        The membership flag

PySpin.**ESlopeClass_FromString** (*ValueStr*, *pValue*) → bool

    Parameters

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

- **pValue** (*Spinnaker::GenApi::ESlope **) –

PySpin.**ESlopeClass_ToString** (**args*)

    ToString(ValueStr, pValue)

    Parameters

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

- **pValue** (*Spinnaker::GenApi::ESlope **) –

- **-> gcstring** ([ESlopeClass_ToString](Value)) –

- **Value** (*enum Spinnaker::GenApi::ESlope*) –

class 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*) –

thisown
   The membership flag

PySpin.**EStandardNameSpaceClass_FromString**(*ValueStr*, *pValue*) → bool

   Parameters

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

   - **pValue**(*Spinnaker::GenApi::EStandardNameSpace **) –

PySpin.**EStandardNameSpaceClass_ToString**(**args*)
   ToString(ValueStr, pValue)

   Parameters

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

   - **pValue**(*Spinnaker::GenApi::EStandardNameSpace **) –

   - **-> gcstring**([EStandardNameSpaceClass_ToString(Value)]) –

   - **Value**(*enum Spinnaker::GenApi::EStandardNameSpace*) –

class 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*) –

thisown
   The membership flag

PySpin.**EVisibilityClass_FromString**(*ValueStr*, *pValue*) → bool

Parameters

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

- **pValue** (*Spinnaker::GenApi::EVisibility *) –

PySpin.**EVisibilityClass_ToString**(**args*)
     ToString(ValueStr, pValue)

Parameters

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

- **pValue** (*Spinnaker::GenApi::EVisibility *) –

- **-> gcstring** (EVisibilityClass_ToString*(Value)*) –

- **Value** (*enum Spinnaker::GenApi::EVisibility*) –

class 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*) –

thisown
     The membership flag

PySpin.**EYesNoClass_FromString**(*ValueStr*, *pValue*) → bool

Parameters

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

- **pValue** (*Spinnaker::GenApi::EYesNo *) –

PySpin.**EYesNoClass_ToString**(**args*)
     ToString(ValueStr, pValue)

Parameters

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

- **pValue** (*Spinnaker::GenApi::EYesNo *) –

- **-> gcstring** (EYesNoClass_ToString*(Value)*) –

- **Value** (*enum Spinnaker::GenApi::EYesNo*) –

PySpin.**EatComments**(*arg1*) → std::istream &

>   **Parameters is** (`std::istream &`) –

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

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

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

> Bases: *PySpin.IEnumEntry*, *PySpin.ValueNode*

Interface for string properties.

C++ includes: EnumEntryNode.h

GetNumericValue(*self*) → double

>   **Parameters self** (`Spinnaker::GenApi::EnumEntryNode *`) –

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

Get double number associated with the entry

GetSymbolic(*self*) → gcstring

>   **Parameters self** (`Spinnaker::GenApi::EnumEntryNode const *`) –

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

Get symbolic enum value

GetValue(*self*) → int64_t

>   **Parameters self** (`Spinnaker::GenApi::EnumEntryNode *`) –

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

Get numeric enum value

IsSelfClearing(*self*) → bool

>   **Parameters self** (`Spinnaker::GenApi::EnumEntryNode *`) –

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

Indicates if the corresponding EnumEntry is self clearing

SetReference(*self*, *pBase*)

>   **Parameters pBase** (`Spinnaker::GenApi::INode *`) –

virtual void Spinnaker::GenApi::EnumEntryNode::SetReference(INode *pBase)

overload SetReference for EnumEntry

thisown

> The membership flag

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

> Bases: *PySpin.IEnumeration*, *PySpin.ValueNode*

Interface for string properties.

C++ includes: EnumNode.h

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

>   **Parameters**
>
> - **Verify** (`bool`) –

- **IgnoreCache**(*bool*) –
- **Verify=False) -> IEnumEntry**(GetCurrentEntry(*self,*) –
- **Verify** –
- **-> IEnumEntry**(GetCurrentEntry(*self*)) –
- **self**(*Spinnaker::GenApi::EnumNode **) –

virtual IEnumEntry* Spinnaker::GenApi::EnumNode::GetCurrentEntry(bool Verify=false, bool Ignore-Cache=false)

Get the current entry

GetEntries(*self*)

> Parameters **self**(*Spinnaker::GenApi::EnumNode **) –

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

Get list of entry nodes

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

> Parameters **IntValue**(*int64_t const*) –

virtual IEnumEntry* Spinnaker::GenApi::EnumNode::GetEntry(const int64_t IntValue)

Get an entry node by its IntValue

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

> Parameters **Symbolic**(*Spinnaker::GenICam::gcstring const &*) –

virtual IEnumEntry* Spinnaker::GenApi::EnumNode::GetEntryByName(const GenICam::gcstring &Symbolic)

Get an entry node by name

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

> Parameters
>
> - **Verify**(*bool*) –
> - **IgnoreCache**(*bool*) –
> - **Verify=False) -> int64_t**(GetIntValue(*self,*) –
> - **Verify** –
> - **-> int64_t**(GetIntValue(*self*)) –
> - **self**(*Spinnaker::GenApi::EnumNode **) –

virtual int64_t Spinnaker::GenApi::EnumNode::GetIntValue(bool Verify=false, bool IgnoreCache=false)

Get integer node value

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

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

The value read

GetSymbolics(*self*, *Symbolics*)

> Parameters **Symbolics**(*Spinnaker::GenApi::StringList_t &*) –

virtual void Spinnaker::GenApi::EnumNode::GetSymbolics(StringList_t &Symbolics)

Get list of symbolic Values

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

> **Parameters**
>
> > - **Value** (*int64_t*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetIntValue(self,*) –
> >
> > - **Value** –

virtual void Spinnaker::GenApi::EnumNode::SetIntValue(int64_t Value, bool Verify=true)

Set integer node value

Value: The value to set

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

SetReference(*self*, *pBase*)

> **Parameters pBase** (*Spinnaker::GenApi::INode **) –

virtual void Spinnaker::GenApi::EnumNode::SetReference(INode *****pBase)

overload SetReference for Enumeration

thisown
: The membership flag

class PySpin.**Event**(**args*, ***kwargs*)
: Bases: object

The base class for all event types.

C++ includes: Event.h

GetEventPayloadData(*self*) → PyObject *

> **Parameters self** (*Spinnaker::Event **) –

const uint8_t* Spinnaker::Event::GetEventPayloadData()

Gets the event payload data

The event payload data

GetEventPayloadDataSize(*self*) → size_t const

> **Parameters self** (*Spinnaker::Event **) –

const size_t Spinnaker::Event::GetEventPayloadDataSize()

Gets the event payload data size

The event payload data size

GetEventType(*self*) → Spinnaker::EventType

> **Parameters self** (*Spinnaker::Event **) –

EventType Spinnaker::Event::GetEventType()

Gets the event type

The event type

SetEventType(*self*, *eventType*)

> **Parameters eventType** (`enum Spinnaker::EventType`) –

> void Spinnaker::Event::SetEventType(EventType eventType)

> Sets the event type

> eventType: The event type

thisown
> The membership flag

class `PySpin.`**FloatNode**(**args*, ***kwargs*)
> Bases: *PySpin.IFloat*, *PySpin.ValueNode*

> Interface for string properties.

> C++ includes: FloatNode.h

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

>> **Parameters self** (`Spinnaker::GenApi::FloatNode const *`) –

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

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

> **GetDisplayPrecision**(*self*) → int64_t

>> **Parameters self** (`Spinnaker::GenApi::FloatNode const *`) –

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

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

> **GetEnumAlias**(*self*) → IEnumeration

>> **Parameters self** (`Spinnaker::GenApi::FloatNode *`) –

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

>> gets the interface of an alias node.

> **GetInc**(*self*) → double

>> **Parameters self** (`Spinnaker::GenApi::FloatNode *`) –

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

>> Get the constant increment if there is any

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

>> **Parameters self** (`Spinnaker::GenApi::FloatNode *`) –

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

>> Get increment mode

> **GetIntAlias**(*self*) → IInteger

>> **Parameters self** (`Spinnaker::GenApi::FloatNode *`) –

>> IInteger* Spinnaker::GenApi::FloatNode::GetIntAlias()

>> gets the interface of an alias node.

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

>> **Parameters**

- **bounded** (*bool*) –
- **-> double_autovector_t** ([GetListOfValidValues](self)) –
- **self** (*Spinnaker::GenApi::FloatNode **) –

virtual double_autovector_t Spinnaker::GenApi::FloatNode::GetListOfValidValues(bool bounded=true)

Get list of valid value

GetMax (*self*) → double

>>> **Parameters self** (*Spinnaker::GenApi::FloatNode **) –

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

Get maximum value allowed

GetMin (*self*) → double

>>> **Parameters self** (*Spinnaker::GenApi::FloatNode **) –

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

Get minimum value allowed

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

>>> **Parameters self** (*Spinnaker::GenApi::FloatNode **) –

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

Get recommended representation

GetUnit (*self*) → gcstring

>>> **Parameters self** (*Spinnaker::GenApi::FloatNode const **) –

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

Get the physical unit name

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

>>> **Parameters**
>>>
>>> - **Verify** (*bool*) –
>>> - **IgnoreCache** (*bool*) –
>>> - **Verify=False) -> double** ([GetValue](self,)) –
>>> - **Verify** –
>>> - **-> double** ([GetValue](self)) –
>>> - **self** (*Spinnaker::GenApi::FloatNode **) –

virtual double Spinnaker::GenApi::FloatNode::GetValue(bool Verify=false, bool IgnoreCache=false)

Get node value

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

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

The value read

HasInc (*self*) → bool

>>> **Parameters self** (*Spinnaker::GenApi::FloatNode **) –

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

True if the float has a constant increment

ImposeMax(*self*, *Value*)

> **Parameters Value** (`double`) –

virtual void Spinnaker::GenApi::FloatNode::ImposeMax(double Value)

Restrict maximum value

ImposeMin(*self*, *Value*)

> **Parameters Value** (`double`) –

virtual void Spinnaker::GenApi::FloatNode::ImposeMin(double Value)

Restrict minimum value

SetReference(*self*, *pBase*)

> **Parameters pBase** (`Spinnaker::GenApi::INode *`) –

virtual void Spinnaker::GenApi::FloatNode::SetReference(INode *pBase)

overload SetReference for Float

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

> **Parameters**
>
> - **Value** (`double`) –
> - **Verify** (`bool`) –
> - **Value)** (`SetValue(self,`) –
> - **Value** –

virtual void Spinnaker::GenApi::FloatNode::SetValue(double Value, bool Verify=true)

Set node value

Value: The value to set

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

thisown
The membership flag

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

Bases: *PySpin.FloatNode*, *PySpin.RegisterNode*

Interface for string properties.

C++ includes: FloatRegNode.h

SetReference(*self*, *pBase*)

> **Parameters pBase** (`Spinnaker::GenApi::INode *`) –

virtual void Spinnaker::GenApi::FloatRegNode::SetReference(INode *pBase)

overload SetReference for Value

thisown
The membership flag

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

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

> Parameters

>> • **FileTemplate** (*Spinnaker::GenICam::gcstring const &*) –

>> • **DirectoriesOnly** (*bool const*) –

>> • **GetFiles(FileTemplate)** –

>> • **FileTemplate** –

SPINNAKER_API void Spinnaker::GenICam::GetFiles(const gcstring &FileTemplate, gcstring_vector &File-Names, const bool DirectoriesOnly=false)

Gets a list of files or directories matching a given FileTemplate

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 SetGenI-CamCLProtocolFolder(). 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.**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 SetGenI-CamCacheFolder(). 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.**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 GetGenICam-LogConfig() is called before SetGenICamLogConfig(), it will return the value of environment variable GENI-CAM_LOG_CONFIG_Vx_y. If this environment variable does not exist, an exception will be thrown.

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

> **Parameters pBase** (*Spinnaker::GenApi::IBase **) –

GenICam::gcstring Spinnaker::GenApi::GetInterfaceName(IBase *pBase)

Returns the name of the main interface as string DEPRICATED, use IBase::GetPrincipalInterfaceType() instead

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

> **Parameters pFunction** (*void **) –

SPINNAKER_API gcstring Spinnaker::GenICam::GetModulePathFromFunction(void *pFunction)

true = only subdirectories (ex . and ..) are retrieved; false = only files are retrieved

Gets the full path to the module (DLL/SO) containing the given pFunction; empty string if not found.

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

> Parameters

>> • **VariableName** (*Spinnaker::GenICam::gcstring const &*) –

- **VariableContent) -> bool**(GetValueOfEnvironmentVariable(*VariableName,* *)* –

- **VariableName** –

- **VariableContent**(*Spinnaker::GenICam::gcstring &*) –

SPINNAKER_API bool Spinnaker::GenICam::GetValueOfEnvironmentVariable(const gcstring &Variable-Name, gcstring &VariableContent)

Retrieve the value of an environment variable true if environment variable was found, otherwise false

class PySpin.**H264Option**
    Bases: object

    Options for saving H264 files.

    C++ includes: SpinnakerDefs.h

    bitrate
        H264Option_bitrate_get(self) -> unsigned int

            Parameters self (*Spinnaker::H264Option **) –

    frameRate
        H264Option_frameRate_get(self) -> float

            Parameters self (*Spinnaker::H264Option **) –

    height
        H264Option_height_get(self) -> unsigned int

            Parameters self (*Spinnaker::H264Option **) –

    reserved
        H264Option_reserved_get(self) -> unsigned int [256]

            Parameters self (*Spinnaker::H264Option **) –

    thisown
        The membership flag

    width
        H264Option_width_get(self) -> unsigned int

            Parameters self (*Spinnaker::H264Option **) –

class PySpin.**IArrivalEvent**(**args*, ***kwargs*)
    Bases: *PySpin.Event*

    Proxy of C++ Spinnaker::IArrivalEvent class.

    OnDeviceArrival(*self*, *serialNumber*)

            Parameters serialNumber (*uint64_t*) –

    thisown
        The membership flag

class PySpin.**IBase**(**args*, ***kwargs*)
    Bases: object

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

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

            Parameters self (*Spinnaker::GenApi::IBase const **) –

thisown
> The membership flag

class PySpin.**IBoolean**(*args*, ***kwargs*)
> Bases: *PySpin.IValue*

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

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

>> **Parameters**

>>> • **Verify** (*bool*) –

>>> • **IgnoreCache** (*bool*) –

>>> • **Verify=False) -> bool** (GetValue(self,) –

>>> • **Verify** –

>>> • **-> bool** (GetValue(self)) –

>>> • **self** (*Spinnaker::GenApi::IBoolean const **) –

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

>> **Parameters**

>>> • **Value** (*bool*) –

>>> • **Verify** (*bool*) –

>>> • **Value)** (SetValue(self,) –

>>> • **Value** –

> **thisown**
>> The membership flag

class PySpin.**ICategory**(*args*, ***kwargs*)
> Bases: *PySpin.IValue*

> Proxy of C++ Spinnaker::GenApi::ICategory class.

> **GetFeatures**(*self*)

>> **Parameters self** (*Spinnaker::GenApi::ICategory const **) –

> **thisown**
>> The membership flag

class PySpin.**IChunkData**(*args*, ***kwargs*)
> Bases: object

> Proxy of C++ Spinnaker::IChunkData class.

> **GetBlackLevel**(*self*) → float64_t

>> **Parameters self** (*Spinnaker::IChunkData const **) –

> **GetCRC**(*self*) → int64_t

>> **Parameters self** (*Spinnaker::IChunkData const **) –

> **GetCounterValue**(*self*) → int64_t

>> **Parameters self** (*Spinnaker::IChunkData const **) –

> **GetEncoderValue**(*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetExposureEndLineStatusAll (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetExposureTime (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetFrameID (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetGain (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetHeight (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetImage (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetLinePitch (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetLineStatusAll (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetOffsetX (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetOffsetY (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetPartSelector (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetPixelDynamicRangeMax (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetPixelDynamicRangeMin (*self*) → int64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetScan3dAxisMax (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetScan3dAxisMin (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetScan3dCoordinateOffset (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetScan3dCoordinateReferenceValue (*self*) → float64_t

> **Parameters self** (*Spinnaker::IChunkData const **) –

GetScan3dCoordinateScale (*self*) → float64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetScan3dInvalidDataValue (*self*) → float64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetScan3dTransformValue (*self*) → float64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetScanLineSelector (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetSequencerSetActive (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetSerialDataLength (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetStreamChannelID (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetTimerValue (*self*) → float64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetTimestamp (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetTimestampLatchValue (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetTransferBlockID (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetTransferQueueCurrentBlockCount (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

GetWidth (*self*) → int64_t

> Parameters **self** (*Spinnaker::IChunkData const* *) –

SetChunks (*self*, *pNodeMap*)

> Parameters **pNodeMap** (*Spinnaker::GenApi::INodeMap &*) –

thisown
> The membership flag

class PySpin.**ICommand** (**args*, ***kwargs*)
> Bases: *PySpin.IValue*

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

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

>> Parameters

>> - **Verify** (*bool*) –
>> - **Execute(self)** –
>> - **self** (*Spinnaker::GenApi::ICommand* *) –

IsDone(*self*, *Verify=True*) → bool

> Parameters
>
> > • **Verify**(*bool*) –
> >
> > • **-> bool**(`IsDone(self)`) –
> >
> > • **self**(*Spinnaker::GenApi::ICommand **) –

thisown
> The membership flag

class `PySpin.`**IDestroy**(**args*, ***kwargs*)
> Bases: `object`
>
> Proxy of C++ Spinnaker::GenApi::IDestroy class.
>
> **Destroy**(*self*)
>
> > Parameters **self**(*Spinnaker::GenApi::IDestroy **) –
>
> **thisown**
> > The membership flag

class `PySpin.`**IDeviceEvent**(**args*, ***kwargs*)
> Bases: `PySpin.Event`
>
> Proxy of C++ Spinnaker::IDeviceEvent class.
>
> **GetDeviceEventId**(*self*) → uint64_t
>
> > Parameters **self**(*Spinnaker::IDeviceEvent const **) –
>
> **GetDeviceEventName**(*self*) → gcstring
>
> > Parameters **self**(*Spinnaker::IDeviceEvent const **) –
>
> **OnDeviceEvent**(*self*, *eventName*)
>
> > Parameters **eventName**(*Spinnaker::GenICam::gcstring*) –
>
> **thisown**
> > The membership flag

class `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
>
> > Parameters **self**(*Spinnaker::GenApi::IDeviceInfo **) –
>
> **GetProductGuid**(*self*) → gcstring
>
> > Parameters **self**(*Spinnaker::GenApi::IDeviceInfo **) –

GetSchemaVersion(*self*, *Version*)

> **Parameters Version** (*Spinnaker::GenICam::Version_t &*) –

GetStandardNameSpace(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IDeviceInfo **) –

GetToolTip(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IDeviceInfo **) –

GetVendorName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IDeviceInfo **) –

GetVersionGuid(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IDeviceInfo **) –

thisown
> The membership flag

class PySpin.**IEnumEntry**(**args*, ***kwargs*)
> Bases: *PySpin.IValue*

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

GetNumericValue(*self*) → double

> **Parameters self** (*Spinnaker::GenApi::IEnumEntry **) –

GetSymbolic(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IEnumEntry const **) –

GetValue(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IEnumEntry **) –

IsSelfClearing(*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::IEnumEntry **) –

thisown
> The membership flag

class 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*) –

thisown
> The membership flag

class PySpin.**IEnumeration**(**args*, ***kwargs*)
    Bases: *PySpin.IValue*

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

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

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

                    • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –

                    • **Verify** –

                    • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

                    • **self** (*Spinnaker::GenApi::IEnumeration **) –

    GetEntries(*self*)

            Parameters **self** (*Spinnaker::GenApi::IEnumeration **) –

    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*) –

                    • **Verify=False) -> int64_t** (*GetIntValue(self,)* –

                    • **Verify** –

                    • **-> int64_t** (*GetIntValue(self)*) –

                    • **self** (*Spinnaker::GenApi::IEnumeration **) –

    GetSymbolics(*self*, *Symbolics*)

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

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

            Parameters

                    • **Value** (*int64_t*) –

                    • **Verify** (*bool*) –

                    • **Value)** (*SetIntValue(self,)* –

                    • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_AcquisitionModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AcquisitionModeEnums)> class.

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

        Parameters

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
- **Verify** –
- **-> IEnumEntry** (`GetCurrentEntry(self)`) –
- **self** (*Spinnaker::GenApi::IEnumerationT< AcquisitionModeEnums > **) –

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

        Parameters Value (*enum Spinnaker::AcquisitionModeEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AcquisitionModeEnums

        Parameters

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

    :param GetValue(self, Verify=False) -> Spinnaker::AcquisitionModeEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::AcquisitionModeEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< AcquisitionModeEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::AcquisitionModeEnums*) –
- **Verify** (*bool*) –
- **Value)** (`SetValue(self,`) –
- **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_AcquisitionStatusSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AcquisitionStatusSelectorEnums)> class.

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

        Parameters

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

- **Verify=False) -> IEnumEntry**(`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry**(`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  AcquisitionStatusSelectorEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::AcquisitionStatusSelectorEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AcquisitionStatusSelectorEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::AcquisitionStatusSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::AcquisitionStatusSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *AcquisitionStatusSelectorEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::AcquisitionStatusSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,)`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ActionUnconditionalModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ActionUnconditionalModeEnums)> class.

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

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry**(`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry**(`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  ActionUnconditionalModeEnums > *) –

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

> > > Parameters **Value** (*enum Spinnaker::ActionUnconditionalModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ActionUnconditionalModeEnums

> > Parameters

> > > - **Verify** (*bool*) –
> > > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ActionUnconditionalModeEnums:

> > Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ActionUnconditionalModeEnums:

> > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< ActionUnconditionalModeEnums > **) –

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

> > Parameters

> > > - **Value** (*enum Spinnaker::ActionUnconditionalModeEnums*) –
> > > - **Verify** (*bool*) –
> > > - **Value)** (*SetValue(self,*) –
> > > - **Value** –

> **thisown**
> > The membership flag

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

> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AdcBitDepthEnums)> class.

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

> > Parameters

> > > - **Verify** (*bool*) –
> > > - **IgnoreCache** (*bool*) –
> > > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > > - **Verify** –
> > > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > > - **self** (*Spinnaker::GenApi::IEnumerationT< AdcBitDepthEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > > Parameters **Value** (*enum Spinnaker::AdcBitDepthEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AdcBitDepthEnums

> > Parameters

> > > - **Verify** (*bool*) –
> > > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::AdcBitDepthEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::AdcBitDepthEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< AdcBitDepthEnums > **) –

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

> **Parameters**

>> • **Value** (*enum Spinnaker::AdcBitDepthEnums*) –

>> • **Verify** (*bool*) –

>> • **Value)** (*SetValue(self,*) –

>> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_AutoAlgorithmSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AutoAlgorithmSelectorEnums)> class.

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

> **Parameters**

>> • **Verify** (*bool*) –

>> • **IgnoreCache** (*bool*) –

>> • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

>> • **Verify** –

>> • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>> • **self** (*Spinnaker::GenApi::IEnumerationT< AutoAlgorithmSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoAlgorithmSelectorEnums

> **Parameters**

>> • **Verify** (*bool*) –

>> • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::AutoAlgorithmSelectorEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::AutoAlgorithmSelectorEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< AutoAlgorithmSelectorEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::AutoAlgorithmSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,)`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_AutoExposureControlPriorityEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AutoExposureControlPriorityEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureControlPriorityEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::AutoExposureControlPriorityEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureControlPriorityEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::AutoExposureControlPriorityEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::AutoExposureControlPriorityEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureControlPriorityEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::AutoExposureControlPriorityEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AutoExposureLightingModeEnums)> class.

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

        Parameters

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
- **Verify** –
- **-> IEnumEntry** (`GetCurrentEntry(self)`) –
- **self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureLightingModeEnums > **) –

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

        Parameters Value (*enum Spinnaker::AutoExposureLightingModeEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureLightingModeEnums

        Parameters

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

    :param GetValue(self, Verify=False) -> Spinnaker::AutoExposureLightingModeEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::AutoExposureLightingModeEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< AutoExposureLightingModeEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::AutoExposureLightingModeEnums*) –
- **Verify** (*bool*) –
- **Value)** (`SetValue(self,`) –
- **Value** –

    thisown

        The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AutoExposureMeteringModeEnums)> class.

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

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureMeteringModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::AutoExposureMeteringModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureMeteringModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::AutoExposureMeteringModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::AutoExposureMeteringModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureMeteringModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::AutoExposureMeteringModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

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

Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(AutoExposureTargetGreyValueAutoEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureTargetGreyValueAutoEnums > **) –

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

>> **Parameters Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::AutoExposureTargetGreyValueAutoEnums

> **Parameters**

>> • **Verify** (*bool*) –

>> • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::AutoExposureTargetGreyValueAutoEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::AutoExposureTargetGreyValueAutoEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< AutoExposureTargetGreyValueAutoEnums > **) –

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

> **Parameters**

>> • **Value** (*enum Spinnaker::AutoExposureTargetGreyValueAutoEnums*) –

>> • **Verify** (*bool*) –

>> • **Value)** (*SetValue(self,)*) –

>> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BalanceRatioSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BalanceRatioSelectorEnums)> class.

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

> **Parameters**

>> • **Verify** (*bool*) –

>> • **IgnoreCache** (*bool*) –

>> • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

>> • **Verify** –

>> • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>> • **self** (*Spinnaker::GenApi::IEnumerationT< BalanceRatioSelectorEnums > **) –

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

>> **Parameters Value** (*enum Spinnaker::BalanceRatioSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BalanceRatioSelectorEnums

> **Parameters**

>> • **Verify** (*bool*) –

>> • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::BalanceRatioSelectorEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::BalanceRatioSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *BalanceRatioSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::BalanceRatioSelectorEnums`) –
> - **Verify** (`bool`) –
> - **Value)** (`SetValue(self,`) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BalanceWhiteAutoEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BalanceWhiteAutoEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> - **Verify** –
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
> *BalanceWhiteAutoEnums > **) –

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

> **Parameters Value** (`enum Spinnaker::BalanceWhiteAutoEnums const`) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BalanceWhiteAutoEnums

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::BalanceWhiteAutoEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::BalanceWhiteAutoEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *BalanceWhiteAutoEnums > **) –

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

> **Parameters**

> - **Value** (*enum Spinnaker::BalanceWhiteAutoEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BalanceWhiteAutoProfileEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BalanceWhiteAutoProfileEnums)> class.

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

>> **Parameters**

>>> - **Verify** (*bool*) –
>>>
>>> - **IgnoreCache** (*bool*) –
>>>
>>> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>>>
>>> - **Verify** –
>>>
>>> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>>>
>>> - **self** (*Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoProfileEnums > **) –

> **GetEntry** (*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums const*) –

> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BalanceWhiteAutoProfileEnums

>> **Parameters**

>>> - **Verify** (*bool*) –
>>>
>>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::BalanceWhiteAutoProfileEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::BalanceWhiteAutoProfileEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< BalanceWhiteAutoProfileEnums > **) –

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

>> **Parameters**

>>> - **Value** (*enum Spinnaker::BalanceWhiteAutoProfileEnums*) –
>>>
>>> - **Verify** (*bool*) –
>>>
>>> - **Value)** (*SetValue(self,*) –
>>>
>>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BinningHorizontalModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BinningHorizontalModeEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > - **Verify** –
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< BinningHorizontalModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::BinningHorizontalModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BinningHorizontalModeEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::BinningHorizontalModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::BinningHorizontalModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< BinningHorizontalModeEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::BinningHorizontalModeEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** (*SetValue(self,*) –
> > - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_BinningSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BinningSelectorEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< BinningSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::BinningSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BinningSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::BinningSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::BinningSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< BinningSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::BinningSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BinningVerticalModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BinningVerticalModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< BinningVerticalModeEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::BinningVerticalModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BinningVerticalModeEnums

> Parameters
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::BinningVerticalModeEnums:
>
> Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::BinningVerticalModeEnums:
>
> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< BinningVerticalModeEnums > ∗*) –

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

> Parameters
>
> > • **Value** (*enum Spinnaker::BinningVerticalModeEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** ([*SetValue(self,*](#)) –
> >
> > • **Value** –

thisown

> The membership flag

class PySpin.**IEnumerationT_BlackLevelAutoBalanceEnums** (*∗args*, *∗∗kwargs*)

> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BlackLevelAutoBalanceEnums)> class.
>
> **GetCurrentEntry** (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** ([*GetCurrentEntry(self,*](#)) –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** ([*GetCurrentEntry(self)*](#)) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelAutoBalanceEnums > ∗*) –
>
> **GetEntry** (*self*, *Value*) → IEnumEntry
>
> > Parameters **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums const*) –
>
> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BlackLevelAutoBalanceEnums
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> >
> > :param GetValue(self, Verify=False) -> Spinnaker::BlackLevelAutoBalanceEnums:

> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::BlackLevelAutoBalanceEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelAutoBalanceEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::BlackLevelAutoBalanceEnums*) –
> - **Verify** (*bool*) –
> - **Value)** ([SetValue(self,)](#)) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BlackLevelAutoEnums** (**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BlackLevelAutoEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –
> - **Verify** –
> - **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelAutoEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::BlackLevelAutoEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BlackLevelAutoEnums

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::BlackLevelAutoEnums:

> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::BlackLevelAutoEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelAutoEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::BlackLevelAutoEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue(self,) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_BlackLevelSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BlackLevelSelectorEnums)> class.

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

> **Parameters**

>> - **Verify** (*bool*) –

>> - **IgnoreCache** (*bool*) –

>> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

>> - **Verify** –

>> - **-> IEnumEntry** (GetCurrentEntry(self)) –

>> - **self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::BlackLevelSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BlackLevelSelectorEnums

> **Parameters**

>> - **Verify** (*bool*) –

>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::BlackLevelSelectorEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::BlackLevelSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< BlackLevelSelectorEnums > **) –

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

> **Parameters**

>> - **Value** (*enum Spinnaker::BlackLevelSelectorEnums*) –

>> - **Verify** (*bool*) –

>> - **Value)** (SetValue(self,) –

>> - **Value** –

thisown
> The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BsiFlatFieldCorrectionAutoEnums)> class.

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

            Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionAutoEnums > **) –

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

            Parameters Value (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BsiFlatFieldCorrectionAutoEnums

            Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::BsiFlatFieldCorrectionAutoEnums:

            Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::BsiFlatFieldCorrectionAutoEnums:

            Parameters self (*Spinnaker::GenApi::IEnumerationT< BsiFlatFieldCorrectionAutoEnums > **) –

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

            Parameters

- **Value** (*enum Spinnaker::BsiFlatFieldCorrectionAutoEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,)`) –

- **Value** –

    thisown

        The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(BsiFlatFieldCorrectionGainSelectorEnums)> class.

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

            Parameters

- **Verify** (*bool*) –

- **IgnoreCache**(*bool*) –

- **Verify=False) -> IEnumEntry**(GetCurrentEntry(*self,*) –

- **Verify** –

- **-> IEnumEntry**(GetCurrentEntry(*self*)) –

- **self** (*Spinnaker::GenApi::IEnumerationT<
  BsiFlatFieldCorrectionGainSelectorEnums > **) –

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

> **Parameters Value**(*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums
> const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<
> BsiFlatFieldCorrectionGainSelectorEnums > **) –

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

> **Parameters**
>
> - **Value**(*enum Spinnaker::BsiFlatFieldCorrectionGainSelectorEnums*)
>   –
>
> - **Verify**(*bool*) –
>
> - **Value)**(SetValue(*self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkBlackLevelSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkBlackLevelSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –
>
> - **Verify=False) -> IEnumEntry**(GetCurrentEntry(*self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry**(GetCurrentEntry(*self*)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkBlackLevelSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkBlackLevelSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkBlackLevelSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkBlackLevelSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkBlackLevelSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkCounterSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkCounterSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkCounterSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkCounterSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkCounterSelectorEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkCounterSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkCounterSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<* *ChunkCounterSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkCounterSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkEncoderSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkEncoderSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<* *ChunkEncoderSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkEncoderSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkEncoderSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkEncoderSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkEncoderSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<* *ChunkEncoderSelectorEnums > **) –

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

> **Parameters**
>
> > - **Value** (`enum Spinnaker::ChunkEncoderSelectorEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** (`SetValue(self,`) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkEncoderStatusEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkEncoderStatusEnums)> class.

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

> **Parameters**
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –
> >
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> >
> > - **self** (`Spinnaker::GenApi::IEnumerationT< ChunkEncoderStatusEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::ChunkEncoderStatusEnums const`) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkEncoderStatusEnums

> **Parameters**
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::ChunkEncoderStatusEnums:
>
> > **Parameters Verify** (`bool`) –
>
> :param GetValue(self) -> Spinnaker::ChunkEncoderStatusEnums:
>
> > **Parameters self** (`Spinnaker::GenApi::IEnumerationT< ChunkEncoderStatusEnums > *`) –

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

> **Parameters**
>
> > - **Value** (`enum Spinnaker::ChunkEncoderStatusEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** (`SetValue(self,`) –
> >
> > - **Value** –

> **thisown**
>> The membership flag

class PySpin.**IEnumerationT_ChunkExposureTimeSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkExposureTimeSelectorEnums)> class.

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

>> **Parameters**

>>> • **Verify**(*bool*) –

>>> • **IgnoreCache**(*bool*) –

>>> • **Verify=False) -> IEnumEntry**(GetCurrentEntry*(self,)* –

>>> • **Verify** –

>>> • **-> IEnumEntry**(GetCurrentEntry*(self)*) –

>>> • **self**   (*Spinnaker::GenApi::IEnumerationT<*
>>>   *ChunkExposureTimeSelectorEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value**   (*enum Spinnaker::ChunkExposureTimeSelectorEnums*
>>   *const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkExposureTimeSelectorEnums

>> **Parameters**

>>> • **Verify**(*bool*) –

>>> • **IgnoreCache**(*bool*) –

>> :param GetValue(self, Verify=False) -> Spinnaker::ChunkExposureTimeSelectorEnums:

>> **Parameters Verify**(*bool*) –

>> :param GetValue(self) -> Spinnaker::ChunkExposureTimeSelectorEnums:

>> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT<*
>>   *ChunkExposureTimeSelectorEnums > **) –

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

>> **Parameters**

>>> • **Value**(*enum Spinnaker::ChunkExposureTimeSelectorEnums*) –

>>> • **Verify**(*bool*) –

>>> • **Value)**(SetValue*(self,)* –

>>> • **Value** –

> **thisown**
>> The membership flag

class PySpin.**IEnumerationT_ChunkGainSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkGainSelectorEnums)> class.

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

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
> > - **Verify** –
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkGainSelectorEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::ChunkGainSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkGainSelectorEnums

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::ChunkGainSelectorEnums:
>
> Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::ChunkGainSelectorEnums:
>
> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< ChunkGainSelectorEnums > **) –

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

> Parameters
>
> > - **Value** (*enum Spinnaker::ChunkGainSelectorEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** (`SetValue(self,)`) –
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkImageComponentEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkImageComponentEnums)> class.

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

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
> > - **Verify** –
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<ChunkImageComponentEnums > ***) –

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

> **Parameters Value** (*enum Spinnaker::ChunkImageComponentEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkImageComponentEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkImageComponentEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkImageComponentEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<ChunkImageComponentEnums > ***) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkImageComponentEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkPixelFormatEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkPixelFormatEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<ChunkPixelFormatEnums > ***) –

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

> **Parameters Value** (*enum Spinnaker::ChunkPixelFormatEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkPixelFormatEnums

> **Parameters**
>
> - **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkPixelFormatEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkPixelFormatEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkPixelFormatEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkPixelFormatEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (*SetValue(self,)*) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkRegionIDEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkRegionIDEnums)> class.

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

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
> > - **Verify** –
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkRegionIDEnums > **) –

> **GetEntry** (*self*, *Value*) → IEnumEntry

> > **Parameters Value** (*enum Spinnaker::ChunkRegionIDEnums const*) –

> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkRegionIDEnums

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkRegionIDEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkRegionIDEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkRegionIDEnums > **) –

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

> Parameters
>
> > - **Value** (*enum Spinnaker::ChunkRegionIDEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** ([*SetValue(self,*](#)) –
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateReferenceSelectorEnums**(**args,*
**kwargs)

Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dCoordinateReferenceSelectorEnums)> class.

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

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** ([*GetCurrentEntry(self,*](#)) –
> > - **Verify** –
> > - **-> IEnumEntry** ([*GetCurrentEntry(self)*](#)) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT<*
> >   *ChunkScan3dCoordinateReferenceSelectorEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums

> Parameters
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums:

> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
> *ChunkScan3dCoordinateReferenceSelectorEnums > **) –

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

> Parameters
>
> > - **Value** (*enum Spinnaker::ChunkScan3dCoordinateReferenceSelectorEnums*)
> >   –
> > - **Verify** (*bool*) –
> > - **Value)** ([*SetValue(self,*](#)) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateSelectorEnums**(*args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dCoordinateSelectorEnums)> class.

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

> > **Parameters**

> > - **Verify** (*bool*) –

> > - **IgnoreCache** (*bool*) –

> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

> > - **Verify** –

> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

> > - **self** (*Spinnaker::GenApi::IEnumerationT<*
> >   *ChunkScan3dCoordinateSelectorEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums*
> > *const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateSelectorEnums

> > **Parameters**

> > - **Verify** (*bool*) –

> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dCoordinateSelectorEnums:

> > **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ChunkScan3dCoordinateSelectorEnums:

> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> > *ChunkScan3dCoordinateSelectorEnums > **) –

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

> > **Parameters**

> > - **Value** (*enum Spinnaker::ChunkScan3dCoordinateSelectorEnums*) –

> > - **Verify** (*bool*) –

> > - **Value)** (*SetValue(self,*) –

> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateSystemEnums**(*args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dCoordinateSystemEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
> > - **Verify** –
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateSystemEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dCoordinateSystemEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::ChunkScan3dCoordinateSystemEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** (`SetValue(self,)`) –
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateSystemReferenceEnums** (**args*, ***kwargs*)

> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dCoordinateSystemReferenceEnums)> class.
>
> **GetCurrentEntry** (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > **Parameters**
> >
> > > - **Verify** (*bool*) –
> > > - **IgnoreCache** (*bool*) –
> > > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateSystemReferenceEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkScan3dCoordinateSystemReferenceEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,)* –
>
> - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_ChunkScan3dCoordinateTransformSelectorEnums** (**args*, ***kwargs*)

Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dCoordinateTransformSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dCoordinateTransformSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::ChunkScan3dCoordinateTransformSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,)* –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dDistanceUnitEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dDistanceUnitEnums)> class.

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

> > **Parameters**
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< ChunkScan3dDistanceUnitEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkScan3dDistanceUnitEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dDistanceUnitEnums

> **Parameters**

- **Verify**(*bool*) –

- **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dDistanceUnitEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::ChunkScan3dDistanceUnitEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *ChunkScan3dDistanceUnitEnums > **) –

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

> **Parameters**
>
> - **Value**(*enum Spinnaker::ChunkScan3dDistanceUnitEnums*) –
>
> - **Verify**(*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkScan3dOutputModeEnums**(**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkScan3dOutputModeEnums)> class.

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

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –
>
> - **Verify=False) -> IEnumEntry**(`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry**(`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
>   *ChunkScan3dOutputModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkScan3dOutputModeEnums const*)
> –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkScan3dOutputModeEnums

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkScan3dOutputModeEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::ChunkScan3dOutputModeEnums:

> > Parameters **self**   (*Spinnaker::GenApi::IEnumerationT<*
> >   *ChunkScan3dOutputModeEnums > ⋆*) –

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

> > **Parameters**

> > > • **Value**(*enum Spinnaker::ChunkScan3dOutputModeEnums*) –

> > > • **Verify**(*bool*) –

> > > • **Value)** (*SetValue(self,*) –

> > > • **Value** –

> **thisown**
>   The membership flag

class PySpin.**IEnumerationT_ChunkSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkSelectorEnums)> class.

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

> > **Parameters**

> > > • **Verify**(*bool*) –

> > > • **IgnoreCache**(*bool*) –

> > > • **Verify=False) -> IEnumEntry**(*GetCurrentEntry(self,*) –

> > > • **Verify** –

> > > • **-> IEnumEntry**(*GetCurrentEntry(self)*) –

> > > • **self** (*Spinnaker::GenApi::IEnumerationT< ChunkSelectorEnums >*
> > >   *⋆*) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters** **Value**(*enum Spinnaker::ChunkSelectorEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkSelectorEnums

> > **Parameters**

> > > • **Verify**(*bool*) –

> > > • **IgnoreCache**(*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ChunkSelectorEnums:

> > **Parameters** **Verify**(*bool*) –

> :param GetValue(self) -> Spinnaker::ChunkSelectorEnums:

> > Parameters **self**   (*Spinnaker::GenApi::IEnumerationT<*
> >   *ChunkSelectorEnums > ⋆*) –

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

> > **Parameters**

> > > • **Value**(*enum Spinnaker::ChunkSelectorEnums*) –

> > > • **Verify**(*bool*) –

> > > • **Value)** (*SetValue(self,*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkSourceIDEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkSourceIDEnums)> class.

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

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

>>> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

>>> - **Verify** –

>>> - **-> IEnumEntry** (GetCurrentEntry(self)) –

>>> - **self** (*Spinnaker::GenApi::IEnumerationT< ChunkSourceIDEnums >* ***) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::ChunkSourceIDEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkSourceIDEnums

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ChunkSourceIDEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ChunkSourceIDEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkSourceIDEnums >* ***) –

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

>> **Parameters**

>>> - **Value** (*enum Spinnaker::ChunkSourceIDEnums*) –

>>> - **Verify** (*bool*) –

>>> - **Value)** (SetValue(self,) –

>>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkTimerSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkTimerSelectorEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< ChunkTimerSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkTimerSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkTimerSelectorEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ChunkTimerSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ChunkTimerSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ChunkTimerSelectorEnums > **) –

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

Parameters

- **Value** (*enum Spinnaker::ChunkTimerSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ChunkTransferStreamIDEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ChunkTransferStreamIDEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

> - **self** (*Spinnaker::GenApi::IEnumerationT<*
>   *ChunkTransferStreamIDEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ChunkTransferStreamIDEnums const*)
> –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ChunkTransferStreamIDEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ChunkTransferStreamIDEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ChunkTransferStreamIDEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *ChunkTransferStreamIDEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ChunkTransferStreamIDEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ClConfigurationEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ClConfigurationEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< ClConfigurationEnums*
>   *> **) –

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

> **Parameters Value** (*enum Spinnaker::ClConfigurationEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ClConfigurationEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ClConfigurationEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ClConfigurationEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *ClConfigurationEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::ClConfigurationEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ClTimeSlotsCountEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ClTimeSlotsCountEnums)> class.

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

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
> *ClTimeSlotsCountEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ClTimeSlotsCountEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ClTimeSlotsCountEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ClTimeSlotsCountEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ClTimeSlotsCountEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *ClTimeSlotsCountEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::ClTimeSlotsCountEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,)*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ColorTransformationSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ColorTransformationSelectorEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< ColorTransformationSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ColorTransformationSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ColorTransformationSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ColorTransformationSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ColorTransformationSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ColorTransformationSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::ColorTransformationSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,)*) –
> >
> > • **Value** –

thisown
    The membership flag

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

    Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ColorTransformationValueSelectorEnums)> class.

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

        Parameters

            • **Verify** (*bool*) –

            • **IgnoreCache** (*bool*) –

            • **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](#)) –

            • **Verify** –

            • **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –

            • **self**   (*Spinnaker::GenApi::IEnumerationT<
              ColorTransformationValueSelectorEnums > **) –

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

        Parameters **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums
            const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ColorTransformationValueSelectorEnums

        Parameters

            • **Verify** (*bool*) –

            • **IgnoreCache** (*bool*) –

        :param GetValue(self, Verify=False) -> Spinnaker::ColorTransformationValueSelectorEnums:

        Parameters **Verify** (*bool*) –

        :param GetValue(self) -> Spinnaker::ColorTransformationValueSelectorEnums:

        Parameters **self**   (*Spinnaker::GenApi::IEnumerationT<
            ColorTransformationValueSelectorEnums > **) –

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

        Parameters

            • **Value** (*enum Spinnaker::ColorTransformationValueSelectorEnums*)
              –

            • **Verify** (*bool*) –

            • **Value)** ([SetValue(self,](#)) –

            • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_CounterEventActivationEnums**(**args*, ***kwargs*)
    Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterEventActivationEnums)> class.

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

> **Parameters**
>
> > * **Verify** (*bool*) –
> >
> > * **IgnoreCache** (*bool*) –
> >
> > * **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > * **Verify** –
> >
> > * **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > * **self** (*Spinnaker::GenApi::IEnumerationT< CounterEventActivationEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::CounterEventActivationEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterEventActivationEnums

> **Parameters**
>
> > * **Verify** (*bool*) –
> >
> > * **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CounterEventActivationEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::CounterEventActivationEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< CounterEventActivationEnums > **) –

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

> **Parameters**
>
> > * **Value** (*enum Spinnaker::CounterEventActivationEnums*) –
> >
> > * **Verify** (*bool*) –
> >
> > * **Value)** (*SetValue(self,*) –
> >
> > * **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CounterEventSourceEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterEventSourceEnums)> class.

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

> **Parameters**
>
> > * **Verify** (*bool*) –
> >
> > * **IgnoreCache** (*bool*) –
> >
> > * **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > * **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  CounterEventSourceEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::CounterEventSourceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterEventSourceEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CounterEventSourceEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::CounterEventSourceEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *CounterEventSourceEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::CounterEventSourceEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CounterResetActivationEnums** (**args*, ***kwargs*)
> Bases: [`PySpin.IEnumeration`](), [`PySpin.IEnumReference`]()

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterResetActivationEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
>   *CounterResetActivationEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::CounterResetActivationEnums*
> *const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterResetActivationEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CounterResetActivationEnums:

Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::CounterResetActivationEnums:

Parameters self (*Spinnaker::GenApi::IEnumerationT< CounterResetActivationEnums > **) –

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

Parameters

- **Value** (*enum Spinnaker::CounterResetActivationEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,)* –

- **Value** –

thisown
  The membership flag

class PySpin.**IEnumerationT_CounterResetSourceEnums** (**args*, ***kwargs*)
  Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterResetSourceEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< CounterResetSourceEnums > **) –

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

Parameters Value (*enum Spinnaker::CounterResetSourceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterResetSourceEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CounterResetSourceEnums:

Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::CounterResetSourceEnums:

> > > > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< CounterResetSourceEnums > **) –

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

> > Parameters

> > > - **Value** (*enum Spinnaker::CounterResetSourceEnums*) –

> > > - **Verify** (*bool*) –

> > > - **Value)** ([SetValue(self,](#)) –

> > > - **Value** –

> **thisown**

> > The membership flag

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

> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterSelectorEnums)> class.

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

> > Parameters

> > > - **Verify** (*bool*) –

> > > - **IgnoreCache** (*bool*) –

> > > - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](#)) –

> > > - **Verify** –

> > > - **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –

> > > - **self** (*Spinnaker::GenApi::IEnumerationT< CounterSelectorEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > Parameters **Value** (*enum Spinnaker::CounterSelectorEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterSelectorEnums

> > Parameters

> > > - **Verify** (*bool*) –

> > > - **IgnoreCache** (*bool*) –

> > :param GetValue(self, Verify=False) -> Spinnaker::CounterSelectorEnums:

> > > Parameters **Verify** (*bool*) –

> > :param GetValue(self) -> Spinnaker::CounterSelectorEnums:

> > > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< CounterSelectorEnums > **) –

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

> > Parameters

> > > - **Value** (*enum Spinnaker::CounterSelectorEnums*) –

> > > - **Verify** (*bool*) –

> > > - **Value)** ([SetValue(self,](#)) –

> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CounterStatusEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterStatusEnums)> class.

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

> > **Parameters**

> > > • **Verify** (*bool*) –

> > > • **IgnoreCache** (*bool*) –

> > > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

> > > • **Verify** –

> > > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

> > > • **self** (*Spinnaker::GenApi::IEnumerationT< CounterStatusEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters Value** (*enum Spinnaker::CounterStatusEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterStatusEnums

> > **Parameters**

> > > • **Verify** (*bool*) –

> > > • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::CounterStatusEnums:

> > **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::CounterStatusEnums:

> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< CounterStatusEnums > **) –

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

> > **Parameters**

> > > • **Value** (*enum Spinnaker::CounterStatusEnums*) –

> > > • **Verify** (*bool*) –

> > > • **Value)** (*SetValue(self,)*) –

> > > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CounterTriggerActivationEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterTriggerActivationEnums)> class.

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

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< CounterTriggerActivationEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::CounterTriggerActivationEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterTriggerActivationEnums

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::CounterTriggerActivationEnums:

> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::CounterTriggerActivationEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< CounterTriggerActivationEnums > **) –

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

> Parameters
>
> > - **Value** (*enum Spinnaker::CounterTriggerActivationEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CounterTriggerSourceEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CounterTriggerSourceEnums)> class.

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

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT<
  CounterTriggerSourceEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::CounterTriggerSourceEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CounterTriggerSourceEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CounterTriggerSourceEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::CounterTriggerSourceEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<
> CounterTriggerSourceEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::CounterTriggerSourceEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CxpConnectionTestModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CxpConnectionTestModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<
>   CxpConnectionTestModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::CxpConnectionTestModeEnums const*)
> –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CxpConnectionTestModeEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CxpConnectionTestModeEnums:

    Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::CxpConnectionTestModeEnums:

    Parameters self   (*Spinnaker::GenApi::IEnumerationT< CxpConnectionTestModeEnums > **) –

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

    Parameters

- **Value** (*enum Spinnaker::CxpConnectionTestModeEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_CxpLinkConfigurationEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CxpLinkConfigurationEnums)> class.

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

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self**   (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationEnums > **) –

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

    Parameters Value (*enum Spinnaker::CxpLinkConfigurationEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CxpLinkConfigurationEnums

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CxpLinkConfigurationEnums:

    Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::CxpLinkConfigurationEnums:

    Parameters self   (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::CxpLinkConfigurationEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_CxpLinkConfigurationPreferredEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CxpLinkConfigurationPreferredEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationPreferredEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CxpLinkConfigurationPreferredEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::CxpLinkConfigurationPreferredEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::CxpLinkConfigurationPreferredEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationPreferredEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::CxpLinkConfigurationPreferredEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

> **thisown**
>> The membership flag

class PySpin.**IEnumerationT_CxpLinkConfigurationStatusEnums**(**args*, ***kwargs*)
>> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

>> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CxpLinkConfigurationStatusEnums)> class.

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

>>> Parameters

>>>> - **Verify** (*bool*) –

>>>> - **IgnoreCache** (*bool*) –

>>>> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

>>>> - **Verify** –

>>>> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –

>>>> - **self** (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationStatusEnums > **) –

>> **GetEntry**(*self*, *Value*) → IEnumEntry

>>> Parameters **Value** (*enum Spinnaker::CxpLinkConfigurationStatusEnums const*) –

>> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CxpLinkConfigurationStatusEnums

>>> Parameters

>>>> - **Verify** (*bool*) –

>>>> - **IgnoreCache** (*bool*) –

>> :param GetValue(self, Verify=False) -> Spinnaker::CxpLinkConfigurationStatusEnums:

>>> Parameters **Verify** (*bool*) –

>> :param GetValue(self) -> Spinnaker::CxpLinkConfigurationStatusEnums:

>>> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< CxpLinkConfigurationStatusEnums > **) –

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

>>> Parameters

>>>> - **Value** (*enum Spinnaker::CxpLinkConfigurationStatusEnums*) –

>>>> - **Verify** (*bool*) –

>>>> - **Value)** (`SetValue(self,)`) –

>>>> - **Value** –

> **thisown**
>> The membership flag

class PySpin.**IEnumerationT_CxpPoCxpStatusEnums**(**args*, ***kwargs*)
>> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

>> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(CxpPoCxpStatusEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry*(self,)*) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry*(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< CxpPoCxpStatusEnums > **) –

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

Parameters **Value** (*enum Spinnaker::CxpPoCxpStatusEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::CxpPoCxpStatusEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::CxpPoCxpStatusEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::CxpPoCxpStatusEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT< CxpPoCxpStatusEnums > **) –

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

Parameters

- **Value** (*enum Spinnaker::CxpPoCxpStatusEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue*(self,)*) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_DecimationHorizontalModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DecimationHorizontalModeEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry*(self,)*) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry*(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
DecimationHorizontalModeEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::DecimationHorizontalModeEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DecimationHorizontalModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DecimationHorizontalModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DecimationHorizontalModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *DecimationHorizontalModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DecimationHorizontalModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DecimationSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DecimationSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
> *DecimationSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DecimationSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DecimationSelectorEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DecimationSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DecimationSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DecimationSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DecimationSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DecimationVerticalModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DecimationVerticalModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< DecimationVerticalModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DecimationVerticalModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DecimationVerticalModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DecimationVerticalModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DecimationVerticalModeEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
> *DecimationVerticalModeEnums > *)* –

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

> Parameters
>
> - **Value** (*enum Spinnaker::DecimationVerticalModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** ([SetValue(self,)](#)) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeinterlacingEnums**(**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeinterlacingEnums)> class.

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

> Parameters
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –
>
> - **Verify** –
>
> - **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< DeinterlacingEnums >*
>   **)* –

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

> Parameters **Value** (*enum Spinnaker::DeinterlacingEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeinterlacingEnums

> Parameters
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeinterlacingEnums:

> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeinterlacingEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
> *DeinterlacingEnums > *)* –

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

> Parameters
>
> - **Value** (*enum Spinnaker::DeinterlacingEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** ([SetValue(self,)](#)) –

> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceAccessStatusEnum**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceAccessStatusEnum)> class.

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

>> **Parameters**

>>> • **Verify** (*bool*) –

>>> • **IgnoreCache** (*bool*) –

>>> • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

>>> • **Verify** –

>>> • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>>> • **self** (*Spinnaker::GenApi::IEnumerationT< DeviceAccessStatusEnum > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::DeviceAccessStatusEnum const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceAccessStatusEnum

>> **Parameters**

>>> • **Verify** (*bool*) –

>>> • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceAccessStatusEnum:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceAccessStatusEnum:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceAccessStatusEnum > **) –

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

>> **Parameters**

>>> • **Value** (*enum Spinnaker::DeviceAccessStatusEnum*) –

>>> • **Verify** (*bool*) –

>>> • **Value)** (*SetValue(self,)*) –

>>> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceCharacterSetEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceCharacterSetEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry(self)) –

- **self** (*Spinnaker::GenApi::IEnumerationT<
  DeviceCharacterSetEnums > **) –

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

Parameters **Value** (*enum Spinnaker::DeviceCharacterSetEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceCharacterSetEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceCharacterSetEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceCharacterSetEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT<
  DeviceCharacterSetEnums > **) –

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

Parameters

- **Value** (*enum Spinnaker::DeviceCharacterSetEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue(self,) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_DeviceClockSelectorEnums** (**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceClockSelectorEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry(self)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< DeviceClockSelectorEnums > **) –

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

    Parameters **Value** (*enum Spinnaker::DeviceClockSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceClockSelectorEnums

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceClockSelectorEnums:

    Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceClockSelectorEnums:

    Parameters **self** (*Spinnaker::GenApi::IEnumerationT< DeviceClockSelectorEnums > **) –

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

    Parameters

- **Value** (*enum Spinnaker::DeviceClockSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,)*) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_DeviceConnectionStatusEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceConnectionStatusEnums)> class.

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

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< DeviceConnectionStatusEnums > **) –

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

    Parameters **Value** (*enum Spinnaker::DeviceConnectionStatusEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceConnectionStatusEnums

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceConnectionStatusEnums:

    Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceConnectionStatusEnums:

    Parameters self (*Spinnaker::GenApi::IEnumerationT< DeviceConnectionStatusEnums > **) –

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

    Parameters

- **Value** (*enum Spinnaker::DeviceConnectionStatusEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_DeviceCurrentSpeedEnum** (**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceCurrentSpeedEnum)> class.

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

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< DeviceCurrentSpeedEnum > **) –

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

    Parameters Value (*enum Spinnaker::DeviceCurrentSpeedEnum const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceCurrentSpeedEnum

    Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceCurrentSpeedEnum:

    Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceCurrentSpeedEnum:

    Parameters self (*Spinnaker::GenApi::IEnumerationT< DeviceCurrentSpeedEnum > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::DeviceCurrentSpeedEnum*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceEndianessMechanismEnum**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceEndianessMechanismEnum)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceEndianessMechanismEnum > **) –

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

> > **Parameters Value** (*enum Spinnaker::DeviceEndianessMechanismEnum const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceEndianessMechanismEnum

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceEndianessMechanismEnum:

> > **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceEndianessMechanismEnum:

> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceEndianessMechanismEnum > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::DeviceEndianessMechanismEnum*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

> **thisown**
> The membership flag

class PySpin.**IEnumerationT_DeviceIndicatorModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceIndicatorModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> - **Verify** –
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceIndicatorModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DeviceIndicatorModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceIndicatorModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceIndicatorModeEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceIndicatorModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceIndicatorModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DeviceIndicatorModeEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

> **thisown**
> The membership flag

class PySpin.**IEnumerationT_DeviceLinkHeartbeatModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceLinkHeartbeatModeEnums)> class.

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

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<
  DeviceLinkHeartbeatModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums
> const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceLinkHeartbeatModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceLinkHeartbeatModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceLinkHeartbeatModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<
> DeviceLinkHeartbeatModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DeviceLinkHeartbeatModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceLinkThroughputLimitModeEnums** (**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceLinkThroughputLimitModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  DeviceLinkThroughputLimitModeEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceLinkThroughputLimitModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceLinkThroughputLimitModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceLinkThroughputLimitModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *DeviceLinkThroughputLimitModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DeviceLinkThroughputLimitModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DevicePowerSupplySelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DevicePowerSupplySelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
>   *DevicePowerSupplySelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DevicePowerSupplySelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DevicePowerSupplySelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DevicePowerSupplySelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DevicePowerSupplySelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::DevicePowerSupplySelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceRegistersEndiannessEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceRegistersEndiannessEnums)> class.
>
> **GetCurrentEntry** (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > **Parameters**
> >
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceRegistersEndiannessEnums > **) –
>
> **GetEntry** (*self*, *Value*) → IEnumEntry
>
> > **Parameters Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums const*) –
>
> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceRegistersEndiannessEnums
>
> > **Parameters**
> >
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::DeviceRegistersEndiannessEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::DeviceRegistersEndiannessEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
DeviceRegistersEndiannessEnums > ⋆) –

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

Parameters

- **Value** (*enum Spinnaker::DeviceRegistersEndiannessEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue(self,)](#)) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_DeviceScanTypeEnums**(**args*, ***kwargs*)
Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceScanTypeEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –

- **Verify** –

- **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< DeviceScanTypeEnums*
> ⋆) –

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

Parameters **Value** (*enum Spinnaker::DeviceScanTypeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceScanTypeEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceScanTypeEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceScanTypeEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
DeviceScanTypeEnums > ⋆) –

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

Parameters

- **Value** (*enum Spinnaker::DeviceScanTypeEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue(self,)](#)) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceSerialPortBaudRateEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceSerialPortBaudRateEnums)> class.

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

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

>>> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

>>> - **Verify** –

>>> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>>> - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceSerialPortBaudRateEnums

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceSerialPortBaudRateEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceSerialPortBaudRateEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceSerialPortBaudRateEnums > **) –

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

>> **Parameters**

>>> - **Value** (*enum Spinnaker::DeviceSerialPortBaudRateEnums*) –

>>> - **Verify** (*bool*) –

>>> - **Value)** (*SetValue(self,)*) –

>>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceSerialPortSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceSerialPortSelectorEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](#)) –
> > - **Verify** –
> > - **-> IEnumEntry** ([GetCurrentEntry(self))](#) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceSerialPortSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceSerialPortSelectorEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceSerialPortSelectorEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceSerialPortSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceSerialPortSelectorEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::DeviceSerialPortSelectorEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** ([SetValue(self,](#)) –
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceStreamChannelEndiannessEnums**(**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceStreamChannelEndiannessEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](#)) –
> > - **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (`Spinnaker::GenApi::IEnumerationT<`
  `DeviceStreamChannelEndiannessEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::DeviceStreamChannelEndiannessEnums`
> `const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceStreamChannelEndiannessEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceStreamChannelEndiannessEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::DeviceStreamChannelEndiannessEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT<`
> `DeviceStreamChannelEndiannessEnums > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::DeviceStreamChannelEndiannessEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)` –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceStreamChannelTypeEnums** (**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceStreamChannelTypeEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)` –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT<`
>   `DeviceStreamChannelTypeEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::DeviceStreamChannelTypeEnums`
> `const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceStreamChannelTypeEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::DeviceStreamChannelTypeEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::DeviceStreamChannelTypeEnums:
>
> > **Parameters self**   (*Spinnaker::GenApi::IEnumerationT<*
> > *DeviceStreamChannelTypeEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::DeviceStreamChannelTypeEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceTLTypeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceTLTypeEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self**   (*Spinnaker::GenApi::IEnumerationT< DeviceTLTypeEnums >*
> >   ***) –

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

> **Parameters Value** (*enum Spinnaker::DeviceTLTypeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceTLTypeEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::DeviceTLTypeEnums:
>
> > **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::DeviceTLTypeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *DeviceTLTypeEnums > **) –

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

> **Parameters**
>
> * **Value** (*enum Spinnaker::DeviceTLTypeEnums*) –
> * **Verify** (*bool*) –
> * **Value)** (*SetValue(self,*) –
> * **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceTapGeometryEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceTapGeometryEnums)> class.

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

>> **Parameters**
>>
>> * **Verify** (*bool*) –
>> * **IgnoreCache** (*bool*) –
>> * **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>> * **Verify** –
>> * **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>> * **self** (*Spinnaker::GenApi::IEnumerationT<*
>>   *DeviceTapGeometryEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::DeviceTapGeometryEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceTapGeometryEnums

>> **Parameters**
>>
>> * **Verify** (*bool*) –
>> * **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceTapGeometryEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceTapGeometryEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
>> *DeviceTapGeometryEnums > **) –

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

> **Parameters**
>
> * **Value** (*enum Spinnaker::DeviceTapGeometryEnums*) –
> * **Verify** (*bool*) –

- **Value)** (*SetValue(self,)*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_DeviceTemperatureSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceTemperatureSelectorEnums)> class.

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

> **Parameters**

>> - **Verify** (*bool*) –

>> - **IgnoreCache** (*bool*) –

>> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

>> - **Verify** –

>> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>> - **self** (*Spinnaker::GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceTemperatureSelectorEnums

> **Parameters**

>> - **Verify** (*bool*) –

>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::DeviceTemperatureSelectorEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::DeviceTemperatureSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< DeviceTemperatureSelectorEnums > **) –

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

> **Parameters**

>> - **Value** (*enum Spinnaker::DeviceTemperatureSelectorEnums*) –

>> - **Verify** (*bool*) –

>> - **Value)** (*SetValue(self,)*) –

>> - **Value** –

thisown
> The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceTypeEnum)> class.

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

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

                    • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

                    • **Verify** –

                    • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

                    • **self** (*Spinnaker::GenApi::IEnumerationT< DeviceTypeEnum > **) –

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

            Parameters Value (*enum Spinnaker::DeviceTypeEnum const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceTypeEnum

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

        :param GetValue(self, Verify=False) -> Spinnaker::DeviceTypeEnum:

            Parameters Verify (*bool*) –

        :param GetValue(self) -> Spinnaker::DeviceTypeEnum:

            Parameters self (*Spinnaker::GenApi::IEnumerationT< DeviceTypeEnum > **) –

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

            Parameters

                    • **Value** (*enum Spinnaker::DeviceTypeEnum*) –

                    • **Verify** (*bool*) –

                    • **Value)** (*SetValue(self,)*) –

                    • **Value** –

    thisown

        The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(DeviceTypeEnums)> class.

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

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (`Spinnaker::GenApi::IEnumerationT< DeviceTypeEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::DeviceTypeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::DeviceTypeEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::DeviceTypeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::DeviceTypeEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< DeviceTypeEnums > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::DeviceTypeEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_EncoderModeEnums** (**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT< EncoderModeEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::EncoderModeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderModeEnums

> Parameters
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::EncoderModeEnums:
>
> > Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::EncoderModeEnums:
>
> > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< EncoderModeEnums > **) –

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

> Parameters
>
> > • **Value** (*enum Spinnaker::EncoderModeEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_EncoderOutputModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderOutputModeEnums)> class.
>
> **GetCurrentEntry** (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< EncoderOutputModeEnums > **) –
>
> **GetEntry** (*self*, *Value*) → IEnumEntry
>
> > Parameters **Value** (*enum Spinnaker::EncoderOutputModeEnums const*) –
>
> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderOutputModeEnums
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> >
> > :param GetValue(self, Verify=False) -> Spinnaker::EncoderOutputModeEnums:
> >
> > > Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::EncoderOutputModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *EncoderOutputModeEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::EncoderOutputModeEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_EncoderResetActivationEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderResetActivationEnums)> class.

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

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > - **Verify** –
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT<*
> >   *EncoderResetActivationEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters Value** (*enum Spinnaker::EncoderResetActivationEnums*
> > *const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderResetActivationEnums

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::EncoderResetActivationEnums:

> > **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::EncoderResetActivationEnums:

> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> > *EncoderResetActivationEnums > **) –

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

> > **Parameters**
> >
> > - **Value** (*enum Spinnaker::EncoderResetActivationEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue*(self,)* –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_EncoderResetSourceEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderResetSourceEnums)> class.

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

        Parameters

        - **Verify** (*bool*) –

        - **IgnoreCache** (*bool*) –

        - **Verify=False) -> IEnumEntry** (GetCurrentEntry*(self,)* –

        - **Verify** –

        - **-> IEnumEntry** (GetCurrentEntry*(self)*) –

        - **self** (*Spinnaker::GenApi::IEnumerationT<
          EncoderResetSourceEnums > **) –

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

        Parameters **Value** (*enum Spinnaker::EncoderResetSourceEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderResetSourceEnums

        Parameters

        - **Verify** (*bool*) –

        - **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::EncoderResetSourceEnums:

        Parameters **Verify** (*bool*) –

    :param GetValue(self) -> Spinnaker::EncoderResetSourceEnums:

        Parameters **self** (*Spinnaker::GenApi::IEnumerationT<
          EncoderResetSourceEnums > **) –

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

        Parameters

        - **Value** (*enum Spinnaker::EncoderResetSourceEnums*) –

        - **Verify** (*bool*) –

        - **Value)** (SetValue*(self,)* –

        - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_EncoderSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderSelectorEnums)> class.

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

        Parameters

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
- **Verify** –
- **-> IEnumEntry** (`GetCurrentEntry(self)`) –
- **self** (*Spinnaker::GenApi::IEnumerationT< EncoderSelectorEnums > **) –

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

        Parameters Value (*enum Spinnaker::EncoderSelectorEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderSelectorEnums

        Parameters

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

    :param GetValue(self, Verify=False) -> Spinnaker::EncoderSelectorEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::EncoderSelectorEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< EncoderSelectorEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::EncoderSelectorEnums*) –
- **Verify** (*bool*) –
- **Value)** (`SetValue(self,`) –
- **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_EncoderSourceAEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderSourceAEnums)> class.

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

        Parameters

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

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< EncoderSourceAEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::EncoderSourceAEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderSourceAEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::EncoderSourceAEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::EncoderSourceAEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< EncoderSourceAEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::EncoderSourceAEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_EncoderSourceBEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderSourceBEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< EncoderSourceBEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::EncoderSourceBEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderSourceBEnums

>   Parameters

>   >   • **Verify** (*bool*) –

>   >   • **IgnoreCache** (*bool*) –

>   :param GetValue(self, Verify=False) -> Spinnaker::EncoderSourceBEnums:

>   Parameters **Verify** (*bool*) –

>   :param GetValue(self) -> Spinnaker::EncoderSourceBEnums:

>   Parameters **self**   (*Spinnaker::GenApi::IEnumerationT<*
>   >   *EncoderSourceBEnums > **) –

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

>   Parameters

>   >   • **Value** (*enum Spinnaker::EncoderSourceBEnums*) –

>   >   • **Verify** (*bool*) –

>   >   • **Value)** (`SetValue(self,`) –

>   >   • **Value** –

thisown
>   The membership flag

class PySpin.**IEnumerationT_EncoderStatusEnums**(**args*, ***kwargs*)
>   Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

>   Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EncoderStatusEnums)> class.

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

>   >   Parameters

>   >   >   • **Verify** (*bool*) –

>   >   >   • **IgnoreCache** (*bool*) –

>   >   >   • **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

>   >   >   • **Verify** –

>   >   >   • **-> IEnumEntry** (`GetCurrentEntry(self)`) –

>   >   >   • **self** (*Spinnaker::GenApi::IEnumerationT< EncoderStatusEnums >*
>   >   >   ***) –

>   **GetEntry**(*self*, *Value*) → IEnumEntry

>   >   Parameters **Value** (*enum Spinnaker::EncoderStatusEnums const*) –

>   **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EncoderStatusEnums

>   >   Parameters

>   >   >   • **Verify** (*bool*) –

>   >   >   • **IgnoreCache** (*bool*) –

>   >   :param GetValue(self, Verify=False) -> Spinnaker::EncoderStatusEnums:

>   >   Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::EncoderStatusEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< EncoderStatusEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::EncoderStatusEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (`SetValue(self,`) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_EventNotificationEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EventNotificationEnums)> class.

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

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> > - **Verify** –
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< EventNotificationEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters Value** (*enum Spinnaker::EventNotificationEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EventNotificationEnums

> > **Parameters**
> >
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::EventNotificationEnums:

> > **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::EventNotificationEnums:

> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< EventNotificationEnums > **) –

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

> > **Parameters**
> >
> > - **Value** (*enum Spinnaker::EventNotificationEnums*) –
> > - **Verify** (*bool*) –

- **Value)** ([SetValue](self,)) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_EventSelectorEnums**(*args*, ***kwargs*)
Bases: [*PySpin.IEnumeration*](), [*PySpin.IEnumReference*]()

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(EventSelectorEnums)> class.

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

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** ([GetCurrentEntry](self,)) –

- **Verify** –

- **-> IEnumEntry** ([GetCurrentEntry](self)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< EventSelectorEnums > **) –

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

        Parameters **Value** (*enum Spinnaker::EventSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::EventSelectorEnums

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::EventSelectorEnums:

        Parameters **Verify** (*bool*) –

    :param GetValue(self) -> Spinnaker::EventSelectorEnums:

        Parameters **self** (*Spinnaker::GenApi::IEnumerationT< EventSelectorEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::EventSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue](self,)) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_ExposureActiveModeEnums**(*args*, ***kwargs*)
Bases: [*PySpin.IEnumeration*](), [*PySpin.IEnumReference*]()

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ExposureActiveModeEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< ExposureActiveModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ExposureActiveModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureActiveModeEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::ExposureActiveModeEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::ExposureActiveModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ExposureActiveModeEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::ExposureActiveModeEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ExposureAutoEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ExposureAutoEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (`Spinnaker::GenApi::IEnumerationT< ExposureAutoEnums > *`) –

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

>   **Parameters Value** (`enum Spinnaker::ExposureAutoEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureAutoEnums

>   **Parameters**
>
>   - **Verify** (`bool`) –
>
>   - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::ExposureAutoEnums:

>   **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::ExposureAutoEnums:

>   **Parameters self**   (`Spinnaker::GenApi::IEnumerationT< ExposureAutoEnums > *`) –

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

>   **Parameters**
>
>   - **Value** (`enum Spinnaker::ExposureAutoEnums`) –
>
>   - **Verify** (`bool`) –
>
>   - **Value)** (`SetValue(self,`) –
>
>   - **Value** –

thisown
>   The membership flag

class PySpin.**IEnumerationT_ExposureModeEnums** (**args*, ***kwargs*)
>   Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

>   Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ExposureModeEnums)> class.

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

>   **Parameters**
>
>   - **Verify** (`bool`) –
>
>   - **IgnoreCache** (`bool`) –
>
>   - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
>   - **Verify** –
>
>   - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
>   - **self** (`Spinnaker::GenApi::IEnumerationT< ExposureModeEnums > *`) –

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

>   **Parameters Value** (`enum Spinnaker::ExposureModeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureModeEnums

>   **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ExposureModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ExposureModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ExposureModeEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::ExposureModeEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_ExposureTimeModeEnums** (**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ExposureTimeModeEnums)> class.

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

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< ExposureTimeModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ExposureTimeModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureTimeModeEnums

> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ExposureTimeModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ExposureTimeModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ExposureTimeModeEnums > **) –

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

> Parameters
>
> > - **Value** (`enum Spinnaker::ExposureTimeModeEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** ([SetValue(self,](`SetValue(self,)`) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ExposureTimeSelectorEnums**(**args*, ***kwargs*)
> Bases: *[PySpin.IEnumeration](#)*, *[PySpin.IEnumReference](#)*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ExposureTimeSelectorEnums)> class.

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

> Parameters
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –
> >
> > - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](`GetCurrentEntry(self,)`) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** ([GetCurrentEntry(self)](`GetCurrentEntry(self)`)) –
> >
> > - **self** (`Spinnaker::GenApi::IEnumerationT< ExposureTimeSelectorEnums > *`) –

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

> > Parameters **Value** (`enum Spinnaker::ExposureTimeSelectorEnums const`) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ExposureTimeSelectorEnums

> Parameters
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –

> :param GetValue(self, Verify=False) -> Spinnaker::ExposureTimeSelectorEnums:

> > Parameters **Verify** (`bool`) –

> :param GetValue(self) -> Spinnaker::ExposureTimeSelectorEnums:

> > Parameters **self** (`Spinnaker::GenApi::IEnumerationT< ExposureTimeSelectorEnums > *`) –

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

> Parameters
>
> > - **Value** (`enum Spinnaker::ExposureTimeSelectorEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** ([SetValue(self,](`SetValue(self,)`) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_FileOpenModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(FileOpenModeEnums)> class.

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

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

>>> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

>>> - **Verify** –

>>> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

>>> - **self** (*Spinnaker::GenApi::IEnumerationT< FileOpenModeEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::FileOpenModeEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::FileOpenModeEnums

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::FileOpenModeEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::FileOpenModeEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< FileOpenModeEnums > **) –

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

>> **Parameters**

>>> - **Value** (*enum Spinnaker::FileOpenModeEnums*) –

>>> - **Verify** (*bool*) –

>>> - **Value)** (*SetValue(self,*) –

>>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_FileOperationSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(FileOperationSelectorEnums)> class.

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

>> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< FileOperationSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::FileOperationSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::FileOperationSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::FileOperationSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::FileOperationSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< FileOperationSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::FileOperationSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_FileOperationStatusEnums** (**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(FileOperationStatusEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  FileOperationStatusEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::FileOperationStatusEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::FileOperationStatusEnums

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::FileOperationStatusEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::FileOperationStatusEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *FileOperationStatusEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::FileOperationStatusEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (SetValue(self,) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_FileSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(FileSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –
> - **Verify** –
> - **-> IEnumEntry** (GetCurrentEntry(self)) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< FileSelectorEnums >*
>   ***) –

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

> **Parameters Value** (*enum Spinnaker::FileSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::FileSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::FileSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::FileSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *FileSelectorEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::FileSelectorEnums*) –
- **Verify** (*bool*) –
- **Value)** ([SetValue(self,)](#)) –
- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GUIXMLLocationEnum**(**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GUIXMLLocationEnum)> class.

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

> **Parameters**

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –
- **Verify** –
- **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –
- **self** (*Spinnaker::GenApi::IEnumerationT< GUIXMLLocationEnum >*
  *) –

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

> **Parameters Value** (*enum Spinnaker::GUIXMLLocationEnum const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GUIXMLLocationEnum

> **Parameters**

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

:param GetValue(self, Verify=False) -> Spinnaker::GUIXMLLocationEnum:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::GUIXMLLocationEnum:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *GUIXMLLocationEnum > **) –

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

> Parameters
>> • **Value** (*enum Spinnaker::GUIXMLLocationEnum*) –
>>
>> • **Verify** (*bool*) –
>>
>> • **Value)** (`SetValue(self,)`) –
>>
>> • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GainAutoBalanceEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GainAutoBalanceEnums)> class.

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

>> Parameters
>>> • **Verify** (*bool*) –
>>>
>>> • **IgnoreCache** (*bool*) –
>>>
>>> • **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>>>
>>> • **Verify** –
>>>
>>> • **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>>>
>>> • **self** (*Spinnaker::GenApi::IEnumerationT< GainAutoBalanceEnums > **) –

> **GetEntry** (*self*, *Value*) → IEnumEntry

>> Parameters **Value** (*enum Spinnaker::GainAutoBalanceEnums const*) –

> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GainAutoBalanceEnums

>> Parameters
>>> • **Verify** (*bool*) –
>>>
>>> • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::GainAutoBalanceEnums:

>> Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::GainAutoBalanceEnums:

>> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< GainAutoBalanceEnums > **) –

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

>> Parameters
>>> • **Value** (*enum Spinnaker::GainAutoBalanceEnums*) –
>>>
>>> • **Verify** (*bool*) –
>>>
>>> • **Value)** (`SetValue(self,)`) –
>>>
>>> • **Value** –

thisown
> The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GainAutoEnums)> class.

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

            Parameters

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
- **Verify** –
- **-> IEnumEntry** (`GetCurrentEntry(self)`) –
- **self** (*Spinnaker::GenApi::IEnumerationT< GainAutoEnums > **) –

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

            Parameters Value (*enum Spinnaker::GainAutoEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GainAutoEnums

            Parameters

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

    :param GetValue(self, Verify=False) -> Spinnaker::GainAutoEnums:

            Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::GainAutoEnums:

            Parameters self (*Spinnaker::GenApi::IEnumerationT< GainAutoEnums > **) –

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

            Parameters

- **Value** (*enum Spinnaker::GainAutoEnums*) –
- **Verify** (*bool*) –
- **Value)** (`SetValue(self,`) –
- **Value** –

    thisown

        The membership flag

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

    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GainSelectorEnums)> class.

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

            Parameters

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

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< GainSelectorEnums >* *) –

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

> **Parameters Value** (*enum Spinnaker::GainSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GainSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::GainSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::GainSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GainSelectorEnums >* *) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::GainSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GenICamXMLLocationEnum** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GenICamXMLLocationEnum)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< GenICamXMLLocationEnum >* *) –

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

> **Parameters Value** (*enum Spinnaker::GenICamXMLLocationEnum const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GenICamXMLLocationEnum

> Parameters
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::GenICamXMLLocationEnum:
>
> > Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::GenICamXMLLocationEnum:
>
> > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< GenICamXMLLocationEnum > ★*) –

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

> Parameters
>
> > • **Value** (*enum Spinnaker::GenICamXMLLocationEnum*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** ([SetValue(self,](#)) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevCCPEnum**(**args*, ***kwargs*)
> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevCCPEnum)> class.
>
> **GetCurrentEntry**(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,](#)) –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< GevCCPEnum > ★*) –
>
> **GetEntry**(*self*, *Value*) → IEnumEntry
>
> > Parameters **Value** (*enum Spinnaker::GevCCPEnum const*) –
>
> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevCCPEnum
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::GevCCPEnum:
>
> > Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::GevCCPEnum:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GevCCPEnum > ⋆*) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::GevCCPEnum*) –
> - **Verify** (*bool*) –
> - **Value)** ([SetValue(self,)](#)) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevCCPEnums** (**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevCCPEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –
> - **Verify** –
> - **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< GevCCPEnums > ⋆*) –

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

> **Parameters Value** (*enum Spinnaker::GevCCPEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevCCPEnums

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::GevCCPEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::GevCCPEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GevCCPEnums > ⋆*)
> –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::GevCCPEnums*) –
> - **Verify** (*bool*) –
> - **Value)** ([SetValue(self,)](#)) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevCurrentPhysicalLinkConfigurationEnums**(**args*,
> ***kwargs*)

> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevCurrentPhysicalLinkConfigurationEnums)> class.

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

>> **Parameters**

>>> • **Verify**(*bool*)–

>>> • **IgnoreCache**(*bool*)–

>>> • **Verify=False) -> IEnumEntry**([GetCurrentEntry](#)*(self,)*)–

>>> • **Verify**–

>>> • **-> IEnumEntry**([GetCurrentEntry](#)*(self)*)–

>>> • **self**   (*Spinnaker::GenApi::IEnumerationT<*
>>>   *GevCurrentPhysicalLinkConfigurationEnums > **)–

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value**(*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums*
>>   *const*)–

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevCurrentPhysicalLinkConfigurationEnums

>> **Parameters**

>>> • **Verify**(*bool*)–

>>> • **IgnoreCache**(*bool*)–

> :param GetValue(self, Verify=False) -> Spinnaker::GevCurrentPhysicalLinkConfigurationEnums:

>> **Parameters Verify**(*bool*)–

> :param GetValue(self) -> Spinnaker::GevCurrentPhysicalLinkConfigurationEnums:

>> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT<*
>>   *GevCurrentPhysicalLinkConfigurationEnums > **)–

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

>> **Parameters**

>>> • **Value**(*enum Spinnaker::GevCurrentPhysicalLinkConfigurationEnums*)
>>>   –

>>> • **Verify**(*bool*)–

>>> • **Value)**([SetValue](#)*(self,)*)–

>>> • **Value**–

thisown
> The membership flag

class PySpin.**IEnumerationT_GevGVCPExtendedStatusCodesSelectorEnums**(**args*,
> ***kwargs*)

> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevGVCPExtendedStatusCodesSelectorEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (GetCurrentEntry(self)) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT<*
> >   *GevGVCPExtendedStatusCodesSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums*
> *const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums:
>
> **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums:
>
> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *GevGVCPExtendedStatusCodesSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::GevGVCPExtendedStatusCodesSelectorEnums*)
> >   –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (SetValue(self,) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevGVSPExtendedIDModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevGVSPExtendedIDModeEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums > **) –

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

> **Parameters Value** (`enum Spinnaker::GevGVSPExtendedIDModeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevGVSPExtendedIDModeEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::GevGVSPExtendedIDModeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::GevGVSPExtendedIDModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GevGVSPExtendedIDModeEnums > **) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::GevGVSPExtendedIDModeEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevIEEE1588ClockAccuracyEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevIEEE1588ClockAccuracyEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< GevIEEE1588ClockAccuracyEnums > **) –

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

> > > **Parameters Value**   (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevIEEE1588ClockAccuracyEnums

> > **Parameters**

> > > * **Verify** (*bool*) –

> > > * **IgnoreCache** (*bool*) –

> > :param GetValue(self, Verify=False) -> Spinnaker::GevIEEE1588ClockAccuracyEnums:

> > **Parameters Verify** (*bool*) –

> > :param GetValue(self) -> Spinnaker::GevIEEE1588ClockAccuracyEnums:

> > > **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< GevIEEE1588ClockAccuracyEnums > **) –

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

> > **Parameters**

> > > * **Value** (*enum Spinnaker::GevIEEE1588ClockAccuracyEnums*) –

> > > * **Verify** (*bool*) –

> > > * **Value)** (*SetValue(self,)* –

> > > * **Value** –

> **thisown**
> > The membership flag

class PySpin.**IEnumerationT_GevIEEE1588ModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevIEEE1588ModeEnums)> class.

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

> > **Parameters**

> > > * **Verify** (*bool*) –

> > > * **IgnoreCache** (*bool*) –

> > > * **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –

> > > * **Verify** –

> > > * **-> IEnumEntry** (*GetCurrentEntry(self))* –

> > > * **self** (*Spinnaker::GenApi::IEnumerationT< GevIEEE1588ModeEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

> > **Parameters Value**(*enum Spinnaker::GevIEEE1588ModeEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevIEEE1588ModeEnums

> > **Parameters**

> > > * **Verify** (*bool*) –

> > > * **IgnoreCache** (*bool*) –

> > :param GetValue(self, Verify=False) -> Spinnaker::GevIEEE1588ModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::GevIEEE1588ModeEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *GevIEEE1588ModeEnums > ***) –*

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::GevIEEE1588ModeEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue (self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevIEEE1588StatusEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevIEEE1588StatusEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry (self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry (self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT<*
> > *GevIEEE1588StatusEnums > ***) –*

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

> **Parameters Value** (*enum Spinnaker::GevIEEE1588StatusEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevIEEE1588StatusEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::GevIEEE1588StatusEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::GevIEEE1588StatusEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *GevIEEE1588StatusEnums > ***) –*

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::GevIEEE1588StatusEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_GevIPConfigurationStatusEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevIPConfigurationStatusEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<
>   GevIPConfigurationStatusEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::GevIPConfigurationStatusEnums
> const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevIPConfigurationStatusEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::GevIPConfigurationStatusEnums:
>
> **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::GevIPConfigurationStatusEnums:
>
> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<
> GevIPConfigurationStatusEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::GevIPConfigurationStatusEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

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

   Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

   Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevPhysicalLinkConfigurationEnums)> class.

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

   > **Parameters**
   >
   > - **Verify** (*bool*) –
   > - **IgnoreCache** (*bool*) –
   > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
   > - **Verify** –
   > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
   > - **self** (*Spinnaker::GenApi::IEnumerationT< GevPhysicalLinkConfigurationEnums > **) –

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

   > **Parameters Value** (*enum Spinnaker::GevPhysicalLinkConfigurationEnums const*) –

   GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevPhysicalLinkConfigurationEnums

   > **Parameters**
   >
   > - **Verify** (*bool*) –
   > - **IgnoreCache** (*bool*) –

   :param GetValue(self, Verify=False) -> Spinnaker::GevPhysicalLinkConfigurationEnums:

   > **Parameters Verify** (*bool*) –

   :param GetValue(self) -> Spinnaker::GevPhysicalLinkConfigurationEnums:

   > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GevPhysicalLinkConfigurationEnums > **) –

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

   > **Parameters**
   >
   > - **Value** (*enum Spinnaker::GevPhysicalLinkConfigurationEnums*) –
   > - **Verify** (*bool*) –
   > - **Value)** (`SetValue(self,`) –
   > - **Value** –

   thisown
   > The membership flag

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

   Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

   Proxy of C++ Spinnaker::GenApi::IEnumerationT<(GevSupportedOptionSelectorEnums)> class.

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

   > **Parameters**
   >
   > - **Verify** (*bool*) –

- **IgnoreCache**(*bool*) –

- **Verify=False) -> IEnumEntry**(GetCurrentEntry*(self,)*) –

- **Verify** –

- **-> IEnumEntry**(GetCurrentEntry*(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< GevSupportedOptionSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::GevSupportedOptionSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::GevSupportedOptionSelectorEnums

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::GevSupportedOptionSelectorEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::GevSupportedOptionSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< GevSupportedOptionSelectorEnums > **) –

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

> **Parameters**
>
> - **Value**(*enum Spinnaker::GevSupportedOptionSelectorEnums*) –
>
> - **Verify**(*bool*) –
>
> - **Value)**(SetValue*(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ImageComponentSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ImageComponentSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –
>
> - **Verify=False) -> IEnumEntry**(GetCurrentEntry*(self,)*) –
>
> - **Verify** –
>
> - **-> IEnumEntry**(GetCurrentEntry*(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< ImageComponentSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ImageComponentSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageComponentSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ImageComponentSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::ImageComponentSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< ImageComponentSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::ImageComponentSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

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

> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ImageCompressionJPEGFormatOptionEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageCompressionJPEGFormatOptionEnums

> **Parameters**
>
> - **Verify** (*bool*) –

- **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ImageCompressionJPEGFormatOptionEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::ImageCompressionJPEGFormatOptionEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< ImageCompressionJPEGFormatOptionEnums > **) –

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

> **Parameters**
>
> - **Value**(*enum Spinnaker::ImageCompressionJPEGFormatOptionEnums*) –
>
> - **Verify**(*bool*) –
>
> - **Value)** (SetValue*(self,)* –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ImageCompressionModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ImageCompressionModeEnums)> class.

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

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –
>
> - **Verify=False) -> IEnumEntry**(GetCurrentEntry*(self,)* –
>
> - **Verify** –
>
> - **-> IEnumEntry**(GetCurrentEntry*(self))* –
>
> - **self**   (*Spinnaker::GenApi::IEnumerationT< ImageCompressionModeEnums > **) –

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

> **Parameters Value**(*enum Spinnaker::ImageCompressionModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageCompressionModeEnums

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::ImageCompressionModeEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::ImageCompressionModeEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< ImageCompressionModeEnums > **) –

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

> **Parameters**
>
> > - **Value** (`enum Spinnaker::ImageCompressionModeEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** ([`SetValue(self,`](#)) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_ImageCompressionRateOptionEnums**(**args*, ***kwargs*)
> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(ImageCompressionRateOptionEnums)> class.

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

> **Parameters**
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –
> >
> > - **Verify=False) -> IEnumEntry** ([`GetCurrentEntry(self,`](#)) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** ([`GetCurrentEntry(self)`](#)) –
> >
> > - **self** (`Spinnaker::GenApi::IEnumerationT< ImageCompressionRateOptionEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::ImageCompressionRateOptionEnums const`) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::ImageCompressionRateOptionEnums

> **Parameters**
>
> > - **Verify** (`bool`) –
> >
> > - **IgnoreCache** (`bool`) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::ImageCompressionRateOptionEnums:
>
> **Parameters Verify** (`bool`) –
>
> :param GetValue(self) -> Spinnaker::ImageCompressionRateOptionEnums:
>
> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< ImageCompressionRateOptionEnums > *`) –

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

> **Parameters**
>
> > - **Value** (`enum Spinnaker::ImageCompressionRateOptionEnums`) –
> >
> > - **Verify** (`bool`) –
> >
> > - **Value)** ([`SetValue(self,`](#)) –
> >
> > - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_LUTSelectorEnums**(*args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LUTSelectorEnums)> class.

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

        Parameters

            • **Verify**(*bool*) –

            • **IgnoreCache**(*bool*) –

            • **Verify=False) -> IEnumEntry**(*GetCurrentEntry(self,)*) –

            • **Verify** –

            • **-> IEnumEntry**(*GetCurrentEntry(self)*) –

            • **self**(*Spinnaker::GenApi::IEnumerationT< LUTSelectorEnums > **)
                –

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

        Parameters Value(*enum Spinnaker::LUTSelectorEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LUTSelectorEnums

        Parameters

            • **Verify**(*bool*) –

            • **IgnoreCache**(*bool*) –

        :param GetValue(self, Verify=False) -> Spinnaker::LUTSelectorEnums:

        Parameters Verify(*bool*) –

        :param GetValue(self) -> Spinnaker::LUTSelectorEnums:

        Parameters self(*Spinnaker::GenApi::IEnumerationT< LUTSelectorEnums
            > **) –

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

        Parameters

            • **Value**(*enum Spinnaker::LUTSelectorEnums*) –

            • **Verify**(*bool*) –

            • **Value)**(*SetValue(self,)*) –

            • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_LineFormatEnums**(*args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LineFormatEnums)> class.

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

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< LineFormatEnums > **) –

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

> Parameters **Value** (`enum Spinnaker::LineFormatEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LineFormatEnums

> Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::LineFormatEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::LineFormatEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< LineFormatEnums > **) –

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

> Parameters

- **Value** (`enum Spinnaker::LineFormatEnums`) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LineInputFilterSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LineInputFilterSelectorEnums)> class.

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

> Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< LineInputFilterSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LineInputFilterSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LineInputFilterSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::LineInputFilterSelectorEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::LineInputFilterSelectorEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< LineInputFilterSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::LineInputFilterSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,)*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LineModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LineModeEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< LineModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LineModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LineModeEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::LineModeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::LineModeEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< LineModeEnums >` `*`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::LineModeEnums`) –
> - **Verify** (`bool`) –
> - **Value)** (`SetValue(self,`) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LineSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LineSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> - **Verify** –
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> - **self** (`Spinnaker::GenApi::IEnumerationT< LineSelectorEnums >` `*`) –

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

> **Parameters Value** (`enum Spinnaker::LineSelectorEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LineSelectorEnums

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::LineSelectorEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::LineSelectorEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< LineSelectorEnums >` `*`) –

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

> **Parameters**

- **Value**(*enum Spinnaker::LineSelectorEnums*) –

- **Verify**(*bool*) –

- **Value)** (SetValue(*self,*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LineSourceEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LineSourceEnums)> class.

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

> **Parameters**

>> - **Verify**(*bool*) –

>> - **IgnoreCache**(*bool*) –

>> - **Verify=False) -> IEnumEntry**(GetCurrentEntry(*self,*) –

>> - **Verify** –

>> - **-> IEnumEntry**(GetCurrentEntry(*self*)) –

>> - **self**(*Spinnaker::GenApi::IEnumerationT< LineSourceEnums > **) –

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

> **Parameters Value**(*enum Spinnaker::LineSourceEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LineSourceEnums

> **Parameters**

>> - **Verify**(*bool*) –

>> - **IgnoreCache**(*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::LineSourceEnums:

> **Parameters Verify**(*bool*) –

> :param GetValue(self) -> Spinnaker::LineSourceEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< LineSourceEnums > **) –

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

> **Parameters**

>> - **Value**(*enum Spinnaker::LineSourceEnums*) –

>> - **Verify**(*bool*) –

>> - **Value)** (SetValue(*self,*) –

>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTInputActivationEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LogicBlockLUTInputActivationEnums)> class.

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

        Parameters

- **Verify** (*bool*) –
- **IgnoreCache** (*bool*) –
- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
- **Verify** –
- **-> IEnumEntry** (`GetCurrentEntry(self)`) –
- **self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums > **) –

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

        Parameters Value (*enum Spinnaker::LogicBlockLUTInputActivationEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockLUTInputActivationEnums

        Parameters

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

    :param GetValue(self, Verify=False) -> Spinnaker::LogicBlockLUTInputActivationEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::LogicBlockLUTInputActivationEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputActivationEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::LogicBlockLUTInputActivationEnums*) –
- **Verify** (*bool*) –
- **Value)** (`SetValue(self,`) –
- **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTInputSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LogicBlockLUTInputSelectorEnums)> class.

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

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry(self)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockLUTInputSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::LogicBlockLUTInputSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::LogicBlockLUTInputSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::LogicBlockLUTInputSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (SetValue(self,) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTInputSourceEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LogicBlockLUTInputSourceEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (GetCurrentEntry(self)) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockLUTInputSourceEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::LogicBlockLUTInputSourceEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::LogicBlockLUTInputSourceEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTInputSourceEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::LogicBlockLUTInputSourceEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LogicBlockLUTSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LogicBlockLUTSelectorEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< LogicBlockLUTSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockLUTSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::LogicBlockLUTSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::LogicBlockLUTSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *LogicBlockLUTSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::LogicBlockLUTSelectorEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_LogicBlockSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(LogicBlockSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> - **Verify** –
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
> *LogicBlockSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::LogicBlockSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::LogicBlockSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::LogicBlockSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::LogicBlockSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *LogicBlockSelectorEnums > **) –

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

Parameters

- **Value** (*enum Spinnaker::LogicBlockSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue(self,)](#)) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_POEStatusEnum**(**args*, ***kwargs*)
    Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(POEStatusEnum)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** ([GetCurrentEntry(self,)](#)) –

- **Verify** –

- **-> IEnumEntry** ([GetCurrentEntry(self)](#)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< POEStatusEnum > **) –

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

Parameters **Value** (*enum Spinnaker::POEStatusEnum const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::POEStatusEnum

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::POEStatusEnum:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::POEStatusEnum:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT< POEStatusEnum > **) –

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

Parameters

- **Value** (*enum Spinnaker::POEStatusEnum*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue(self,)](#)) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_PixelColorFilterEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(PixelColorFilterEnums)> class.

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

        Parameters

            • **Verify** (*bool*) –

            • **IgnoreCache** (*bool*) –

            • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

            • **Verify** –

            • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

            • **self** (*Spinnaker::GenApi::IEnumerationT< PixelColorFilterEnums > **) –

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

        Parameters Value (*enum Spinnaker::PixelColorFilterEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::PixelColorFilterEnums

        Parameters

            • **Verify** (*bool*) –

            • **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::PixelColorFilterEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::PixelColorFilterEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< PixelColorFilterEnums > **) –

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

        Parameters

            • **Value** (*enum Spinnaker::PixelColorFilterEnums*) –

            • **Verify** (*bool*) –

            • **Value)** (*SetValue(self,*) –

            • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_PixelFormatEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(PixelFormatEnums)> class.

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

        Parameters

            • **Verify** (*bool*) –

            • **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (`Spinnaker::GenApi::IEnumerationT< PixelFormatEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::PixelFormatEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::PixelFormatEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::PixelFormatEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::PixelFormatEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< PixelFormatEnums > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::PixelFormatEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_PixelFormatInfoSelectorEnums** (**args*, ***kwargs*)
> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(PixelFormatInfoSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > *`) –

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

> **Parameters Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::PixelFormatInfoSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::PixelFormatInfoSelectorEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::PixelFormatInfoSelectorEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< PixelFormatInfoSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::PixelFormatInfoSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_PixelSizeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(PixelSizeEnums)> class.
>
> **GetCurrentEntry**(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > **Parameters**
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< PixelSizeEnums > **) –
>
> **GetEntry**(*self*, *Value*) → IEnumEntry
>
> > **Parameters Value** (*enum Spinnaker::PixelSizeEnums const*) –
>
> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::PixelSizeEnums
>
> > **Parameters**
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> >
> > :param GetValue(self, Verify=False) -> Spinnaker::PixelSizeEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::PixelSizeEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< PixelSizeEnums > **) –

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

> Parameters
>
> - **Value** (*enum Spinnaker::PixelSizeEnums*) –
> - **Verify** (*bool*) –
> - **Value)** ([SetValue (self,)](#)) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_RegionDestinationEnums** (**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](#), [*PySpin.IEnumReference*](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(RegionDestinationEnums)> class.

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

> Parameters
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** ([GetCurrentEntry (self,)](#)) –
> - **Verify** –
> - **-> IEnumEntry** ([GetCurrentEntry (self)](#)) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< RegionDestinationEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::RegionDestinationEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::RegionDestinationEnums

> Parameters
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::RegionDestinationEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::RegionDestinationEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< RegionDestinationEnums > **) –

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

> Parameters
>
> - **Value** (*enum Spinnaker::RegionDestinationEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue (self,)*) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_RegionModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(RegionModeEnums)> class.

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

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

>>> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry (self,)*) –

>>> - **Verify** –

>>> - **-> IEnumEntry** (*GetCurrentEntry (self)*) –

>>> - **self** (*Spinnaker::GenApi::IEnumerationT< RegionModeEnums > **) –

> **GetEntry** (*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::RegionModeEnums const*) –

> **GetValue** (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::RegionModeEnums

>> **Parameters**

>>> - **Verify** (*bool*) –

>>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::RegionModeEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::RegionModeEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< RegionModeEnums > **) –

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

>> **Parameters**

>>> - **Value** (*enum Spinnaker::RegionModeEnums*) –

>>> - **Verify** (*bool*) –

>>> - **Value)** (*SetValue (self,)*) –

>>> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_RegionSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(RegionSelectorEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< RegionSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::RegionSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::RegionSelectorEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::RegionSelectorEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::RegionSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< RegionSelectorEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::RegionSelectorEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (`SetValue(self,)`) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_RgbTransformLightSourceEnums** (**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(RgbTransformLightSourceEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
> >
> > - **Verify** –

- **-> IEnumEntry** (GetCurrentEntry(self)) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
  RgbTransformLightSourceEnums > *) –

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

> **Parameters Value** (*enum Spinnaker::RgbTransformLightSourceEnums*
> *const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::RgbTransformLightSourceEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::RgbTransformLightSourceEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::RgbTransformLightSourceEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *RgbTransformLightSourceEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::RgbTransformLightSourceEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (SetValue(self,) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateReferenceSelectorEnums** (**args*,
**kwargs)

Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dCoordinateReferenceSelectorEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (GetCurrentEntry(self)) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT<*
>   *Scan3dCoordinateReferenceSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dCoordinateReferenceSelectorEnums

> Parameters
>
> > • **Verify**(*bool*) –
> >
> > • **IgnoreCache**(*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::Scan3dCoordinateReferenceSelectorEnums:
>
> > Parameters **Verify**(*bool*) –
>
> :param GetValue(self) -> Spinnaker::Scan3dCoordinateReferenceSelectorEnums:
>
> > Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
> > *Scan3dCoordinateReferenceSelectorEnums > **) –

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

> Parameters
>
> > • **Value**(*enum Spinnaker::Scan3dCoordinateReferenceSelectorEnums*)
> > –
> >
> > • **Verify**(*bool*) –
> >
> > • **Value)** (SetValue*(self,)* –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dCoordinateSelectorEnums)> class.

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

> Parameters
>
> > • **Verify**(*bool*) –
> >
> > • **IgnoreCache**(*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (GetCurrentEntry*(self,)* –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (GetCurrentEntry*(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT<*
> > *Scan3dCoordinateSelectorEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums*
> *const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dCoordinateSelectorEnums

> Parameters
>
> > • **Verify**(*bool*) –
> >
> > • **IgnoreCache**(*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::Scan3dCoordinateSelectorEnums:

Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::Scan3dCoordinateSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *Scan3dCoordinateSelectorEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::Scan3dCoordinateSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown

> The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateSystemEnums** (**args*, ***kwargs*)

> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dCoordinateSystemEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT<*
> > *Scan3dCoordinateSystemEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums*
> *const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dCoordinateSystemEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::Scan3dCoordinateSystemEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::Scan3dCoordinateSystemEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *Scan3dCoordinateSystemEnums > **) –

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

> **Parameters**

- **Value** (*enum Spinnaker::Scan3dCoordinateSystemEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue](self,)) –

- **Value** –

thisown
: The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateSystemReferenceEnums**(**args*,
      **kwargs)

    Bases: [*PySpin.IEnumeration*](), [*PySpin.IEnumReference*]()

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dCoordinateSystemReferenceEnums)> class.

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

          Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** ([GetCurrentEntry](self,)) –

- **Verify** –

- **-> IEnumEntry** ([GetCurrentEntry](self))) –

- **self** (*Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemReferenceEnums > **) –

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

          Parameters Value (*enum Spinnaker::Scan3dCoordinateSystemReferenceEnums const*) –

    GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dCoordinateSystemReferenceEnums

          Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::Scan3dCoordinateSystemReferenceEnums:

          Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::Scan3dCoordinateSystemReferenceEnums:

          Parameters self (*Spinnaker::GenApi::IEnumerationT< Scan3dCoordinateSystemReferenceEnums > **) –

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

          Parameters

- **Value** (*enum Spinnaker::Scan3dCoordinateSystemReferenceEnums*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue](self,)) –

- **Value** –

thisown
: The membership flag

class PySpin.**IEnumerationT_Scan3dCoordinateTransformSelectorEnums**(**args*,
**kwargs)

> Bases: [`PySpin.IEnumeration`](), [`PySpin.IEnumReference`]()

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dCoordinateTransformSelectorEnums)> class.

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

>> **Parameters**

>>> • **Verify** (*bool*) –

>>> • **IgnoreCache** (*bool*) –

>>> • **Verify=False) -> IEnumEntry** ([`GetCurrentEntry(self,`]()) –

>>> • **Verify** –

>>> • **-> IEnumEntry** ([`GetCurrentEntry(self)`]()) –

>>> • **self** (*Spinnaker::GenApi::IEnumerationT<
>>> Scan3dCoordinateTransformSelectorEnums > **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::Scan3dCoordinateTransformSelectorEnums
>> const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dCoordinateTransformSelectorEnums

>> **Parameters**

>>> • **Verify** (*bool*) –

>>> • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::Scan3dCoordinateTransformSelectorEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::Scan3dCoordinateTransformSelectorEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<
>> Scan3dCoordinateTransformSelectorEnums > **) –

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

>> **Parameters**

>>> • **Value** (*enum Spinnaker::Scan3dCoordinateTransformSelectorEnums*)
>>> –

>>> • **Verify** (*bool*) –

>>> • **Value)** ([`SetValue(self,`]()) –

>>> • **Value** –

> **thisown**
>> The membership flag

class PySpin.**IEnumerationT_Scan3dDistanceUnitEnums**(**args*, ***kwargs*)

> Bases: [`PySpin.IEnumeration`](), [`PySpin.IEnumReference`]()

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dDistanceUnitEnums)> class.

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

>> **Parameters**

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry *(self,)*) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry *(self)*) –

- **self**   (*Spinnaker::GenApi::IEnumerationT< Scan3dDistanceUnitEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::Scan3dDistanceUnitEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dDistanceUnitEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::Scan3dDistanceUnitEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::Scan3dDistanceUnitEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< Scan3dDistanceUnitEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::Scan3dDistanceUnitEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (SetValue *(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_Scan3dOutputModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(Scan3dOutputModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry *(self,)*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (GetCurrentEntry *(self)*) –
>
> - **self**   (*Spinnaker::GenApi::IEnumerationT< Scan3dOutputModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::Scan3dOutputModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::Scan3dOutputModeEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::Scan3dOutputModeEnums:
>
> **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::Scan3dOutputModeEnums:
>
> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< Scan3dOutputModeEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::Scan3dOutputModeEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SensorDigitizationTapsEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SensorDigitizationTapsEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< SensorDigitizationTapsEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SensorDigitizationTapsEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SensorDigitizationTapsEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SensorDigitizationTapsEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::SensorDigitizationTapsEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT<`
> `SensorDigitizationTapsEnums > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::SensorDigitizationTapsEnums`) –
> - **Verify** (`bool`) –
> - **Value)** (`SetValue(self,`) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SensorShutterModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SensorShutterModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> - **Verify** –
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> - **self** (`Spinnaker::GenApi::IEnumerationT<`
>   `SensorShutterModeEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::SensorShutterModeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SensorShutterModeEnums

> **Parameters**
>
> - **Verify** (`bool`) –
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::SensorShutterModeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::SensorShutterModeEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT<`
> `SensorShutterModeEnums > *`) –

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

> **Parameters**

- **Value** (*enum Spinnaker::SensorShutterModeEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue(self,) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_SensorTapsEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SensorTapsEnums)> class.

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

        Parameters

            - **Verify** (*bool*) –

            - **IgnoreCache** (*bool*) –

            - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

            - **Verify** –

            - **-> IEnumEntry** (GetCurrentEntry(self)) –

            - **self** (*Spinnaker::GenApi::IEnumerationT< SensorTapsEnums > **) –

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

        Parameters Value (*enum Spinnaker::SensorTapsEnums const*) –

    GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SensorTapsEnums

        Parameters

            - **Verify** (*bool*) –

            - **IgnoreCache** (*bool*) –

        :param GetValue(self, Verify=False) -> Spinnaker::SensorTapsEnums:

        Parameters Verify (*bool*) –

        :param GetValue(self) -> Spinnaker::SensorTapsEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< SensorTapsEnums > **) –

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

        Parameters

            - **Value** (*enum Spinnaker::SensorTapsEnums*) –

            - **Verify** (*bool*) –

            - **Value)** (SetValue(self,) –

            - **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_SequencerConfigurationModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerConfigurationModeEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self**   (*Spinnaker::GenApi::IEnumerationT<
> >   SequencerConfigurationModeEnums > **) –

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

> **Parameters Value**   (*enum Spinnaker::SequencerConfigurationModeEnums
>   const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerConfigurationModeEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::SequencerConfigurationModeEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::SequencerConfigurationModeEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT<
>   SequencerConfigurationModeEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::SequencerConfigurationModeEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_SequencerConfigurationValidEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerConfigurationValidEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry *(self,)* –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry *(self))* –

- **self**   (*Spinnaker::GenApi::IEnumerationT< SequencerConfigurationValidEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SequencerConfigurationValidEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerConfigurationValidEnums

> **Parameters**

> - **Verify** (*bool*) –

> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SequencerConfigurationValidEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SequencerConfigurationValidEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< SequencerConfigurationValidEnums > **) –

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

> **Parameters**

> - **Value** (*enum Spinnaker::SequencerConfigurationValidEnums*) –

> - **Verify** (*bool*) –

> - **Value)** (SetValue *(self,)* –

> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SequencerFeatureSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerFeatureSelectorEnums)> class.

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

> **Parameters**

> - **Verify** (*bool*) –

> - **IgnoreCache** (*bool*) –

> - **Verify=False) -> IEnumEntry** (GetCurrentEntry *(self,)* –

> - **Verify** –

> - **-> IEnumEntry** (GetCurrentEntry *(self))* –

> - **self**   (*Spinnaker::GenApi::IEnumerationT< SequencerFeatureSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SequencerFeatureSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerFeatureSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SequencerFeatureSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SequencerFeatureSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SequencerFeatureSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::SequencerFeatureSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SequencerModeEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerModeEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< SequencerModeEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SequencerModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SequencerModeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::SequencerModeEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< SequencerModeEnums > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::SequencerModeEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown

> The membership flag

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

> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerSetValidEnums)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT< SequencerSetValidEnums > *`) –

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

> **Parameters Value** (`enum Spinnaker::SequencerSetValidEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerSetValidEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::SequencerSetValidEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::SequencerSetValidEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< SequencerSetValidEnums > *`) –

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

> **Parameters**

- **Value**(*enum Spinnaker::SequencerSetValidEnums*) –

- **Verify**(*bool*) –

- **Value)** (*SetValue(self,)* –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SequencerTriggerActivationEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerTriggerActivationEnums)> class.

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

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –
>
> - **Verify=False) -> IEnumEntry**(*GetCurrentEntry(self,)* –
>
> - **Verify** –
>
> - **-> IEnumEntry**(*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< SequencerTriggerActivationEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SequencerTriggerActivationEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerTriggerActivationEnums

> **Parameters**
>
> - **Verify**(*bool*) –
>
> - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SequencerTriggerActivationEnums:

> **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::SequencerTriggerActivationEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SequencerTriggerActivationEnums > **) –

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

> **Parameters**
>
> - **Value**(*enum Spinnaker::SequencerTriggerActivationEnums*) –
>
> - **Verify**(*bool*) –
>
> - **Value)** (*SetValue(self,)* –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SequencerTriggerSourceEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SequencerTriggerSourceEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< SequencerTriggerSourceEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SequencerTriggerSourceEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SequencerTriggerSourceEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::SequencerTriggerSourceEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::SequencerTriggerSourceEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SequencerTriggerSourceEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::SequencerTriggerSourceEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,*) –
> >
> > - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_SerialPortBaudRateEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SerialPortBaudRateEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –

- **IgnoreCache**(*bool*) –

- **Verify=False) -> IEnumEntry**(GetCurrentEntry(self,) –

- **Verify** –

- **-> IEnumEntry**(GetCurrentEntry(self)) –

- **self**   (*Spinnaker::GenApi::IEnumerationT<
  SerialPortBaudRateEnums > **) –

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

>   **Parameters Value**(*enum Spinnaker::SerialPortBaudRateEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortBaudRateEnums

>   **Parameters**
>
>   - **Verify**(*bool*) –
>
>   - **IgnoreCache**(*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SerialPortBaudRateEnums:

>   **Parameters Verify**(*bool*) –

:param GetValue(self) -> Spinnaker::SerialPortBaudRateEnums:

>   **Parameters self**   (*Spinnaker::GenApi::IEnumerationT<
  SerialPortBaudRateEnums > **) –

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

>   **Parameters**
>
>   - **Value**(*enum Spinnaker::SerialPortBaudRateEnums*) –
>
>   - **Verify**(*bool*) –
>
>   - **Value)**(SetValue(self,) –
>
>   - **Value** –

thisown
>   The membership flag

class PySpin.**IEnumerationT_SerialPortParityEnums**(**args*, ***kwargs*)
>   Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

>   Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SerialPortParityEnums)> class.

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

>   **Parameters**
>
>   - **Verify**(*bool*) –
>
>   - **IgnoreCache**(*bool*) –
>
>   - **Verify=False) -> IEnumEntry**(GetCurrentEntry(self,) –
>
>   - **Verify** –
>
>   - **-> IEnumEntry**(GetCurrentEntry(self)) –
>
>   - **self**   (*Spinnaker::GenApi::IEnumerationT<
  SerialPortParityEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SerialPortParityEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortParityEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SerialPortParityEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SerialPortParityEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SerialPortParityEnums > **) –

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

> **Parameters**
>
> > • **Value** (*enum Spinnaker::SerialPortParityEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SerialPortSelectorEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SerialPortSelectorEnums)> class.

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

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< SerialPortSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SerialPortSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SerialPortSelectorEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SerialPortSelectorEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
SerialPortSelectorEnums > *) –

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

Parameters

- **Value** (*enum Spinnaker::SerialPortSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown
The membership flag

class PySpin.**IEnumerationT_SerialPortSourceEnums** (**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SerialPortSourceEnums)> class.

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

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT<*
SerialPortSourceEnums > *) –

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

Parameters **Value** (*enum Spinnaker::SerialPortSourceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortSourceEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SerialPortSourceEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SerialPortSourceEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT<*
SerialPortSourceEnums > *) –

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

Parameters

- **Value** (*enum Spinnaker::SerialPortSourceEnums*) –

- **Verify** (*bool*) –

- **Value)** (SetValue(self,) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SerialPortStopBitsEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SerialPortStopBitsEnums)> class.

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

> **Parameters**

> - **Verify** (*bool*) –

> - **IgnoreCache** (*bool*) –

> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –

> - **Verify** –

> - **-> IEnumEntry** (GetCurrentEntry(self)) –

> - **self** (*Spinnaker::GenApi::IEnumerationT< SerialPortStopBitsEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SerialPortStopBitsEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SerialPortStopBitsEnums

> **Parameters**

> - **Verify** (*bool*) –

> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::SerialPortStopBitsEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::SerialPortStopBitsEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SerialPortStopBitsEnums > **) –

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

> **Parameters**

> - **Value** (*enum Spinnaker::SerialPortStopBitsEnums*) –

> - **Verify** (*bool*) –

> - **Value)** (SetValue(self,) –

> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SoftwareSignalSelectorEnums**(**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SoftwareSignalSelectorEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
> > - **Verify** –
> > - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
> > - **self** (*Spinnaker::GenApi::IEnumerationT< SoftwareSignalSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SoftwareSignalSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SoftwareSignalSelectorEnums

> **Parameters**
>
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SoftwareSignalSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SoftwareSignalSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SoftwareSignalSelectorEnums > **) –

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

> **Parameters**
>
> > - **Value** (*enum Spinnaker::SoftwareSignalSelectorEnums*) –
> > - **Verify** (*bool*) –
> > - **Value)** (`SetValue(self,`) –
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_SourceSelectorEnums**(**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(SourceSelectorEnums)> class.

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

> **Parameters**
>
> > - **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< SourceSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::SourceSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::SourceSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::SourceSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::SourceSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< SourceSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::SourceSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_StreamBufferHandlingModeEnum** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(StreamBufferHandlingModeEnum)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< StreamBufferHandlingModeEnum > **) –

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

> Parameters **Value** (*enum Spinnaker::StreamBufferHandlingModeEnum const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::StreamBufferHandlingModeEnum

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::StreamBufferHandlingModeEnum:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::StreamBufferHandlingModeEnum:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< StreamBufferHandlingModeEnum > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::StreamBufferHandlingModeEnum*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_StreamDefaultBufferCountModeEnum**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(StreamDefaultBufferCountModeEnum)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< StreamDefaultBufferCountModeEnum > **) –

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

> Parameters **Value** (*enum Spinnaker::StreamDefaultBufferCountModeEnum const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::StreamDefaultBufferCountModeEnum

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::StreamDefaultBufferCountModeEnum:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::StreamDefaultBufferCountModeEnum:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT<`
> `StreamDefaultBufferCountModeEnum > *`) –

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

> **Parameters**
>
> - **Value** (`enum Spinnaker::StreamDefaultBufferCountModeEnum`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)` –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_StreamTypeEnum** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(StreamTypeEnum)> class.

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

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)` –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT< StreamTypeEnum > *`) –

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

> **Parameters Value** (`enum Spinnaker::StreamTypeEnum const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::StreamTypeEnum

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::StreamTypeEnum:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::StreamTypeEnum:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< StreamTypeEnum >`
> `*`) –

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

> **Parameters**

- **Value** (*enum Spinnaker::StreamTypeEnum*) –

- **Verify** (*bool*) –

- **Value)** (SetValue(self,)) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TestPatternEnums**(**args*, ***kwargs*)
    Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TestPatternEnums)> class.

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

    Parameters

    - **Verify** (*bool*) –

    - **IgnoreCache** (*bool*) –

    - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,)) –

    - **Verify** –

    - **-> IEnumEntry** (GetCurrentEntry(self)) –

    - **self** (*Spinnaker::GenApi::IEnumerationT< TestPatternEnums > **)
        –

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

    Parameters Value (*enum Spinnaker::TestPatternEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TestPatternEnums

    Parameters

    - **Verify** (*bool*) –

    - **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::TestPatternEnums:

    Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::TestPatternEnums:

    Parameters self (*Spinnaker::GenApi::IEnumerationT< TestPatternEnums
        > **) –

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

    Parameters

    - **Value** (*enum Spinnaker::TestPatternEnums*) –

    - **Verify** (*bool*) –

    - **Value)** (SetValue(self,)) –

    - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TestPatternGeneratorSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TestPatternGeneratorSelectorEnums)> class.

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

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

                    • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)* –

                    • **Verify** –

                    • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

                    • **self**   (*Spinnaker::GenApi::IEnumerationT<
                      TestPatternGeneratorSelectorEnums > **) –

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

            Parameters Value (*enum Spinnaker::TestPatternGeneratorSelectorEnums
                const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TestPatternGeneratorSelectorEnums

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

        :param GetValue(self, Verify=False) -> Spinnaker::TestPatternGeneratorSelectorEnums:

            Parameters Verify (*bool*) –

        :param GetValue(self) -> Spinnaker::TestPatternGeneratorSelectorEnums:

            Parameters self   (*Spinnaker::GenApi::IEnumerationT<
                TestPatternGeneratorSelectorEnums > **) –

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

            Parameters

                    • **Value** (*enum Spinnaker::TestPatternGeneratorSelectorEnums*) –

                    • **Verify** (*bool*) –

                    • **Value)** (*SetValue(self,)* –

                    • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_TimerSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TimerSelectorEnums)> class.

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

            Parameters

                    • **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (GetCurrentEntry(*self,*) –

- **Verify** –

- **-> IEnumEntry** (GetCurrentEntry(*self*)) –

- **self** (*Spinnaker::GenApi::IEnumerationT< TimerSelectorEnums > **) –

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

> **Parameters Value** (*enum Spinnaker::TimerSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TimerSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TimerSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TimerSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TimerSelectorEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::TimerSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (SetValue(*self,*) –
>
> - **Value** –

thisown

> The membership flag

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

Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TimerStatusEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(*self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (GetCurrentEntry(*self*)) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< TimerStatusEnums > **) –

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

> **Parameters** **Value** (*enum Spinnaker::TimerStatusEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TimerStatusEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TimerStatusEnums:

> **Parameters** **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TimerStatusEnums:

> **Parameters** **self** (*Spinnaker::GenApi::IEnumerationT< TimerStatusEnums > **) –

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

> **Parameters**
>
> - **Value** (*enum Spinnaker::TimerStatusEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TimerTriggerActivationEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TimerTriggerActivationEnums)> class.

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

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< TimerTriggerActivationEnums > **) –

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

> **Parameters** **Value** (*enum Spinnaker::TimerTriggerActivationEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TimerTriggerActivationEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TimerTriggerActivationEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TimerTriggerActivationEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< TimerTriggerActivationEnums > **) –

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

> Parameters
>
> - **Value** (*enum Spinnaker::TimerTriggerActivationEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (SetValue(self,) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TimerTriggerSourceEnums** (**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](PySpin.IEnumeration), [*PySpin.IEnumReference*](PySpin.IEnumReference)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TimerTriggerSourceEnums)> class.

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

> Parameters
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** (GetCurrentEntry(self,) –
> - **Verify** –
> - **-> IEnumEntry** (GetCurrentEntry(self)) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< TimerTriggerSourceEnums > **) –

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

> Parameters **Value** (*enum Spinnaker::TimerTriggerSourceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TimerTriggerSourceEnums

> Parameters
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TimerTriggerSourceEnums:

> Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TimerTriggerSourceEnums:

> Parameters **self** (*Spinnaker::GenApi::IEnumerationT< TimerTriggerSourceEnums > **) –

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

> Parameters
>
> - **Value** (*enum Spinnaker::TimerTriggerSourceEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,`) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TransferComponentSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferComponentSelectorEnums)> class.

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

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< TransferComponentSelectorEnums > **) –

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

        Parameters Value (*enum Spinnaker::TransferComponentSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferComponentSelectorEnums

        Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferComponentSelectorEnums:

        Parameters Verify (*bool*) –

:param GetValue(self) -> Spinnaker::TransferComponentSelectorEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< TransferComponentSelectorEnums > **) –

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

        Parameters

- **Value** (*enum Spinnaker::TransferComponentSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,`) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TransferControlModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferControlModeEnums)> class.

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

> Parameters

> > • **Verify** (*bool*) –

> > • **IgnoreCache** (*bool*) –

> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

> > • **Verify** –

> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

> > • **self**   (*Spinnaker::GenApi::IEnumerationT< TransferControlModeEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> Parameters **Value** (*enum Spinnaker::TransferControlModeEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferControlModeEnums

> Parameters

> > • **Verify** (*bool*) –

> > • **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::TransferControlModeEnums:

> > Parameters **Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::TransferControlModeEnums:

> > Parameters **self**   (*Spinnaker::GenApi::IEnumerationT< TransferControlModeEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> Parameters

> > • **Value** (*enum Spinnaker::TransferControlModeEnums*) –

> > • **Verify** (*bool*) –

> > • **Value)** (*SetValue(self,*) –

> > • **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TransferOperationModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferOperationModeEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> Parameters

> > • **Verify** (*bool*) –

> > • **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self**   (*Spinnaker::GenApi::IEnumerationT< TransferOperationModeEnums > **) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TransferOperationModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferOperationModeEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferOperationModeEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TransferOperationModeEnums:

> **Parameters self**   (*Spinnaker::GenApi::IEnumerationT< TransferOperationModeEnums > **) –

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::TransferOperationModeEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (*SetValue(self,)*) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TransferQueueModeEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferQueueModeEnums)> class.

GetCurrentEntry (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
>
> - **self**   (*Spinnaker::GenApi::IEnumerationT< TransferQueueModeEnums > **) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (`enum Spinnaker::TransferQueueModeEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferQueueModeEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferQueueModeEnums:

> **Parameters Verify** (`bool`) –

:param GetValue(self) -> Spinnaker::TransferQueueModeEnums:

> **Parameters self** (`Spinnaker::GenApi::IEnumerationT< TransferQueueModeEnums > *`) –

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (`enum Spinnaker::TransferQueueModeEnums`) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown

> The membership flag

class PySpin.**IEnumerationT_TransferSelectorEnums** (**args*, ***kwargs*)

> Bases: [`PySpin.IEnumeration`](#), [`PySpin.IEnumReference`](#)

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferSelectorEnums)> class.

GetCurrentEntry (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (`Spinnaker::GenApi::IEnumerationT< TransferSelectorEnums > *`) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (`enum Spinnaker::TransferSelectorEnums const`) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferSelectorEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TransferSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TransferSelectorEnums > **) –

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**
>
> > • **Value** (*enum Spinnaker::TransferSelectorEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** ([*SetValue(self,*]) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TransferStatusSelectorEnums** (**args*, ***kwargs*)
> Bases: [*PySpin.IEnumeration*](), [*PySpin.IEnumReference*]()

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferStatusSelectorEnums)> class.

GetCurrentEntry (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** ([*GetCurrentEntry(self,*]) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** ([*GetCurrentEntry(self)*]) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< TransferStatusSelectorEnums > **) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TransferStatusSelectorEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferStatusSelectorEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferStatusSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TransferStatusSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TransferStatusSelectorEnums > **) –

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**

- **Value** (*enum Spinnaker::TransferStatusSelectorEnums*) –

- **Verify** (*bool*) –

- **Value)** (`SetValue(self,`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TransferTriggerActivationEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferTriggerActivationEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerActivationEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TransferTriggerActivationEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferTriggerActivationEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::TransferTriggerActivationEnums:

> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::TransferTriggerActivationEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerActivationEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::TransferTriggerActivationEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_TransferTriggerModeEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferTriggerModeEnums)> class.

    GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

                    • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

                    • **Verify** –

                    • **-> IEnumEntry** (*GetCurrentEntry(self)*) –

                    • **self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerModeEnums > **) –

    GetEntry(*self*, *Value*) → IEnumEntry

            Parameters Value (*enum Spinnaker::TransferTriggerModeEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferTriggerModeEnums

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::TransferTriggerModeEnums:

            Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::TransferTriggerModeEnums:

            Parameters self (*Spinnaker::GenApi::IEnumerationT< TransferTriggerModeEnums > **) –

    SetValue(*self*, *Value*, *Verify=True*)

            Parameters

                    • **Value** (*enum Spinnaker::TransferTriggerModeEnums*) –

                    • **Verify** (*bool*) –

                    • **Value)** (*SetValue(self,)*) –

                    • **Value** –

    thisown
        The membership flag

class PySpin.**IEnumerationT_TransferTriggerSelectorEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferTriggerSelectorEnums)> class.

    GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

            Parameters

                    • **Verify** (*bool*) –

                    • **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry**(`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry**(`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerSelectorEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TransferTriggerSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferTriggerSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TransferTriggerSelectorEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TransferTriggerSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerSelectorEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::TransferTriggerSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown

> The membership flag

class PySpin.**IEnumerationT_TransferTriggerSourceEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TransferTriggerSourceEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry**(`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry**(`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerSourceEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TransferTriggerSourceEnums const*)
> –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TransferTriggerSourceEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::TransferTriggerSourceEnums:
>
> > **Parameters Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::TransferTriggerSourceEnums:
>
> > **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TransferTriggerSourceEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> > • **Value** (*enum Spinnaker::TransferTriggerSourceEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,)*) –
> >
> > • **Value** –

thisown
>   The membership flag

class PySpin.**IEnumerationT_TriggerActivationEnums**(**args*, ***kwargs*)
>   Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TriggerActivationEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
> >
> > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
> >
> > • **Verify** –
> >
> > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::IEnumerationT< TriggerActivationEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TriggerActivationEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerActivationEnums

> **Parameters**
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::TriggerActivationEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TriggerActivationEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT< TriggerActivationEnums > **) –

SetValue (*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerActivationEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,*) –

- **Value** –

thisown

The membership flag

class PySpin.**IEnumerationT_TriggerModeEnums** (**args*, ***kwargs*)

Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TriggerModeEnums)> class.

GetCurrentEntry (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –

- **Verify** –

- **-> IEnumEntry** (*GetCurrentEntry(self)*) –

- **self** (*Spinnaker::GenApi::IEnumerationT< TriggerModeEnums > **) –

GetEntry (*self*, *Value*) → IEnumEntry

Parameters **Value** (*enum Spinnaker::TriggerModeEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerModeEnums

Parameters

- **Verify** (*bool*) –

- **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TriggerModeEnums:

Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TriggerModeEnums:

Parameters **self** (*Spinnaker::GenApi::IEnumerationT< TriggerModeEnums > **) –

SetValue (*self*, *Value*, *Verify=True*)

Parameters

- **Value** (*enum Spinnaker::TriggerModeEnums*) –

- **Verify** (*bool*) –

- **Value)** (*SetValue(self,)*) –

- **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TriggerOverlapEnums**(**args*, ***kwargs*)
    Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

    Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TriggerOverlapEnums)> class.

    GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

        Parameters

            - **Verify** (*bool*) –

            - **IgnoreCache** (*bool*) –

            - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –

            - **Verify** –

            - **-> IEnumEntry** (*GetCurrentEntry(self)*) –

            - **self** (*Spinnaker::GenApi::IEnumerationT< TriggerOverlapEnums > **) –

    GetEntry(*self*, *Value*) → IEnumEntry

        Parameters Value (*enum Spinnaker::TriggerOverlapEnums const*) –

    GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerOverlapEnums

        Parameters

            - **Verify** (*bool*) –

            - **IgnoreCache** (*bool*) –

    :param GetValue(self, Verify=False) -> Spinnaker::TriggerOverlapEnums:

        Parameters Verify (*bool*) –

    :param GetValue(self) -> Spinnaker::TriggerOverlapEnums:

        Parameters self (*Spinnaker::GenApi::IEnumerationT< TriggerOverlapEnums > **) –

    SetValue(*self*, *Value*, *Verify=True*)

        Parameters

            - **Value** (*enum Spinnaker::TriggerOverlapEnums*) –

            - **Verify** (*bool*) –

            - **Value)** (*SetValue(self,)*) –

            - **Value** –

thisown
    The membership flag

class PySpin.**IEnumerationT_TriggerSelectorEnums**(**args*, ***kwargs*)
  Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

  Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TriggerSelectorEnums)> class.

  GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**

> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> - **Verify** –
> - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> - **self** (*Spinnaker::GenApi::IEnumerationT< TriggerSelectorEnums > **) –

  GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TriggerSelectorEnums const*) –

  GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerSelectorEnums

> **Parameters**

> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

  :param GetValue(self, Verify=False) -> Spinnaker::TriggerSelectorEnums:

> **Parameters Verify** (*bool*) –

  :param GetValue(self) -> Spinnaker::TriggerSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TriggerSelectorEnums > **) –

  SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**

> - **Value** (*enum Spinnaker::TriggerSelectorEnums*) –
> - **Verify** (*bool*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

  thisown
    The membership flag

class PySpin.**IEnumerationT_TriggerSourceEnums**(**args*, ***kwargs*)
  Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

  Proxy of C++ Spinnaker::GenApi::IEnumerationT<(TriggerSourceEnums)> class.

  GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**

> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –

- **Verify** –

- **-> IEnumEntry** (`GetCurrentEntry(self)`) –

- **self** (*Spinnaker::GenApi::IEnumerationT< TriggerSourceEnums >* *) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::TriggerSourceEnums const*) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::TriggerSourceEnums

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –

:param GetValue(self, Verify=False) -> Spinnaker::TriggerSourceEnums:

> **Parameters Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::TriggerSourceEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< TriggerSourceEnums > * *) –

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::TriggerSourceEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_U3VCurrentSpeedEnums** (**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(U3VCurrentSpeedEnums)> class.

GetCurrentEntry (*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< U3VCurrentSpeedEnums > * *) –

GetEntry (*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::U3VCurrentSpeedEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::U3VCurrentSpeedEnums

> Parameters
>
> > • **Verify** (*bool*) –
> >
> > • **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::U3VCurrentSpeedEnums:
>
> > Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::U3VCurrentSpeedEnums:
>
> > Parameters **self** (*Spinnaker::GenApi::IEnumerationT< U3VCurrentSpeedEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> Parameters
>
> > • **Value** (*enum Spinnaker::U3VCurrentSpeedEnums*) –
> >
> > • **Verify** (*bool*) –
> >
> > • **Value)** (*SetValue(self,*) –
> >
> > • **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_UserOutputSelectorEnums**(**args*, ***kwargs*)
> Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*
>
> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(UserOutputSelectorEnums)> class.
>
> **GetCurrentEntry**(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> > >
> > > • **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,*) –
> > >
> > > • **Verify** –
> > >
> > > • **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> > >
> > > • **self** (*Spinnaker::GenApi::IEnumerationT< UserOutputSelectorEnums > **) –
>
> **GetEntry**(*self*, *Value*) → IEnumEntry
>
> > Parameters **Value** (*enum Spinnaker::UserOutputSelectorEnums const*) –
>
> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::UserOutputSelectorEnums
>
> > Parameters
> >
> > > • **Verify** (*bool*) –
> > >
> > > • **IgnoreCache** (*bool*) –
> >
> > :param GetValue(self, Verify=False) -> Spinnaker::UserOutputSelectorEnums:
> >
> > > Parameters **Verify** (*bool*) –

:param GetValue(self) -> Spinnaker::UserOutputSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
> *UserOutputSelectorEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::UserOutputSelectorEnums*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_UserSetDefaultEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

> Proxy of C++ Spinnaker::GenApi::IEnumerationT<(UserSetDefaultEnums)> class.

> **GetCurrentEntry**(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

>> **Parameters**
>>
>> - **Verify** (*bool*) –
>>
>> - **IgnoreCache** (*bool*) –
>>
>> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,`) –
>>
>> - **Verify** –
>>
>> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>>
>> - **self** (*Spinnaker::GenApi::IEnumerationT< UserSetDefaultEnums*
>>   *> **) –

> **GetEntry**(*self*, *Value*) → IEnumEntry

>> **Parameters Value** (*enum Spinnaker::UserSetDefaultEnums const*) –

> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::UserSetDefaultEnums

>> **Parameters**
>>
>> - **Verify** (*bool*) –
>>
>> - **IgnoreCache** (*bool*) –

> :param GetValue(self, Verify=False) -> Spinnaker::UserSetDefaultEnums:

>> **Parameters Verify** (*bool*) –

> :param GetValue(self) -> Spinnaker::UserSetDefaultEnums:

>> **Parameters self** (*Spinnaker::GenApi::IEnumerationT<*
>> *UserSetDefaultEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::UserSetDefaultEnums*) –
>
> - **Verify** (*bool*) –

- **Value)** (`SetValue(self,)`) –

- **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_UserSetFeatureSelectorEnums**(**args*, ***kwargs*)
> Bases: `PySpin.IEnumeration`, `PySpin.IEnumReference`

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(UserSetFeatureSelectorEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –
>
> - **Verify=False) -> IEnumEntry** (`GetCurrentEntry(self,)`) –
>
> - **Verify** –
>
> - **-> IEnumEntry** (`GetCurrentEntry(self)`) –
>
> - **self** (*Spinnaker::GenApi::IEnumerationT< UserSetFeatureSelectorEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::UserSetFeatureSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::UserSetFeatureSelectorEnums

> **Parameters**
>
> - **Verify** (`bool`) –
>
> - **IgnoreCache** (`bool`) –

> :param GetValue(self, Verify=False) -> Spinnaker::UserSetFeatureSelectorEnums:

> **Parameters Verify** (`bool`) –

> :param GetValue(self) -> Spinnaker::UserSetFeatureSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< UserSetFeatureSelectorEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::UserSetFeatureSelectorEnums*) –
>
> - **Verify** (`bool`) –
>
> - **Value)** (`SetValue(self,)`) –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_UserSetSelectorEnums**(**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(UserSetSelectorEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
> >
> > - **Verify=False) -> IEnumEntry** (*GetCurrentEntry(self,)*) –
> >
> > - **Verify** –
> >
> > - **-> IEnumEntry** (*GetCurrentEntry(self)*) –
> >
> > - **self** (*Spinnaker::GenApi::IEnumerationT< UserSetSelectorEnums > **) –

GetEntry(*self*, *Value*) → IEnumEntry

> Parameters **Value** (*enum Spinnaker::UserSetSelectorEnums const*) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::UserSetSelectorEnums

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –
>
> :param GetValue(self, Verify=False) -> Spinnaker::UserSetSelectorEnums:
>
> Parameters **Verify** (*bool*) –
>
> :param GetValue(self) -> Spinnaker::UserSetSelectorEnums:
>
> Parameters **self**   (*Spinnaker::GenApi::IEnumerationT< UserSetSelectorEnums > **) –

SetValue(*self*, *Value*, *Verify=True*)

> Parameters
>
> > - **Value** (*enum Spinnaker::UserSetSelectorEnums*) –
> >
> > - **Verify** (*bool*) –
> >
> > - **Value)** (*SetValue(self,)*) –
> >
> > - **Value** –

thisown
> The membership flag

class PySpin.**IEnumerationT_WhiteClipSelectorEnums**(**args*, ***kwargs*)
Bases: *PySpin.IEnumeration*, *PySpin.IEnumReference*

Proxy of C++ Spinnaker::GenApi::IEnumerationT<(WhiteClipSelectorEnums)> class.

GetCurrentEntry(*self*, *Verify=False*, *IgnoreCache=False*) → IEnumEntry

> Parameters
>
> > - **Verify** (*bool*) –
> >
> > - **IgnoreCache** (*bool*) –

- **Verify=False) -> IEnumEntry**(GetCurrentEntry(self,))–

- **Verify** –

- **-> IEnumEntry**(GetCurrentEntry(self))–

- **self** (*Spinnaker::GenApi::IEnumerationT< WhiteClipSelectorEnums > **)–

GetEntry(*self*, *Value*) → IEnumEntry

> **Parameters Value** (*enum Spinnaker::WhiteClipSelectorEnums const*)–

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → Spinnaker::WhiteClipSelectorEnums

> **Parameters**
>
> - **Verify** (*bool*)–
>
> - **IgnoreCache** (*bool*)–

:param GetValue(self, Verify=False) -> Spinnaker::WhiteClipSelectorEnums:

> **Parameters Verify** (*bool*)–

:param GetValue(self) -> Spinnaker::WhiteClipSelectorEnums:

> **Parameters self** (*Spinnaker::GenApi::IEnumerationT< WhiteClipSelectorEnums > **)–

SetValue(*self*, *Value*, *Verify=True*)

> **Parameters**
>
> - **Value** (*enum Spinnaker::WhiteClipSelectorEnums*)–
>
> - **Verify** (*bool*)–
>
> - **Value)** (SetValue(self,))–
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IFloat**(**args*, ***kwargs*)
Bases: *PySpin.IValue*

Proxy of C++ Spinnaker::GenApi::IFloat class.

GetDisplayNotation(*self*) → Spinnaker::GenApi::EDisplayNotation

> **Parameters self** (*Spinnaker::GenApi::IFloat const **)–

GetDisplayPrecision(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IFloat const **)–

GetInc(*self*) → double

> **Parameters self** (*Spinnaker::GenApi::IFloat **)–

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

> **Parameters self** (*Spinnaker::GenApi::IFloat **)–

GetListOfValidValues(*self*, *bounded=True*) → double_autovector_t

> **Parameters**

- **bounded** (*bool*) –

- **-> double_autovector_t** (`GetListOfValidValues(self)`) –

- **self** (*Spinnaker::GenApi::IFloat **) –

GetMax (*self*) → double

> Parameters **self** (*Spinnaker::GenApi::IFloat **) –

GetMin (*self*) → double

> Parameters **self** (*Spinnaker::GenApi::IFloat **) –

GetRepresentation (*self*) → Spinnaker::GenApi::ERepresentation

> Parameters **self** (*Spinnaker::GenApi::IFloat **) –

GetUnit (*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::IFloat const **) –

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → double

> Parameters
>
> - **Verify** (*bool*) –
>
> - **IgnoreCache** (*bool*) –
>
> - **Verify=False) -> double** (`GetValue(self,)` –
>
> - **Verify** –
>
> - **-> double** (`GetValue(self)`) –
>
> - **self** (*Spinnaker::GenApi::IFloat **) –

HasInc (*self*) → bool

> Parameters **self** (*Spinnaker::GenApi::IFloat **) –

ImposeMax (*self*, *Value*)

> Parameters **Value** (*double*) –

ImposeMin (*self*, *Value*)

> Parameters **Value** (*double*) –

SetValue (*self*, *Value*, *Verify=True*)

> Parameters
>
> - **Value** (*double*) –
>
> - **Verify** (*bool*) –
>
> - **Value)** (`SetValue(self,)` –
>
> - **Value** –

thisown
> The membership flag

class PySpin.**IImage** (**args*, ***kwargs*)
> Bases: object

> Proxy of C++ Spinnaker::IImage class.

CalculateStatistics (*self*, *pStatistics*)

Parameters **pStatistics** (*Spinnaker::ImageStatistics &*) –

CheckCRC (*self*) → bool

Parameters **self** (*Spinnaker::IImage const **) –

Convert (*self*, *format*, *algorithm*) → ImagePtr

Parameters

- **format** (*enum Spinnaker::PixelFormatEnums*) –

- **algorithm** (*enum Spinnaker::ColorProcessingAlgorithm*) –

- **format) -> ImagePtr** ([Convert](*self*,) –

- **format** –

DeepCopy (*self*, *pSrcImage*)

Parameters **pSrcImage** (*Spinnaker::ImagePtr const*) –

GetBitsPerPixel (*self*) → size_t

Parameters **self** (*Spinnaker::IImage const **) –

GetBufferSize (*self*) → size_t

Parameters **self** (*Spinnaker::IImage const **) –

GetChunkData (*self*) → ChunkData

Parameters **self** (*Spinnaker::IImage const **) –

GetChunkLayoutId (*self*) → uint64_t

Parameters **self** (*Spinnaker::IImage const **) –

GetColorProcessing (*self*) → Spinnaker::ColorProcessingAlgorithm

Parameters **self** (*Spinnaker::IImage const **) –

GetData (*self*)
    GetData(self) -> PyObject *

Parameters **self** (*Spinnaker::IImage **) –

GetFrameID (*self*) → uint64_t

Parameters **self** (*Spinnaker::IImage const **) –

GetHeight (*self*) → size_t

Parameters **self** (*Spinnaker::IImage const **) –

GetID (*self*) → uint64_t

Parameters **self** (*Spinnaker::IImage const **) –

GetImageSize (*self*) → size_t

Parameters **self** (*Spinnaker::IImage const **) –

GetImageStatus (*self*) → Spinnaker::ImageStatus

Parameters **self** (*Spinnaker::IImage const **) –

GetNDArray (*self*) → PyObject *

Parameters **self** (*Spinnaker::IImage **) –

GetNumChannels(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetPayloadType(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetPixelFormat(*self*) → Spinnaker::PixelFormatEnums

>   **Parameters self** (*Spinnaker::IImage const **) –

GetPixelFormatIntType(*self*) → Spinnaker::PixelFormatIntType

>   **Parameters self** (*Spinnaker::IImage const **) –

GetPixelFormatName(*self*) → gcstring

>   **Parameters self** (*Spinnaker::IImage const **) –

GetPrivateData(*self*) → void *

>   **Parameters self** (*Spinnaker::IImage const **) –

GetStride(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetTLPayloadType(*self*) → Spinnaker::PayloadTypeInfoIDs

>   **Parameters self** (*Spinnaker::IImage const **) –

GetTLPixelFormat(*self*) → uint64_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetTLPixelFormatNamespace(*self*) → Spinnaker::PixelFormatNamespaceID

>   **Parameters self** (*Spinnaker::IImage const **) –

GetTimeStamp(*self*) → uint64_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetValidPayloadSize(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetWidth(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetXOffset(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetXPadding(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetYOffset(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

GetYPadding(*self*) → size_t

>   **Parameters self** (*Spinnaker::IImage const **) –

HasCRC(*self*) → bool

>   **Parameters self** (*Spinnaker::IImage const **) –

IsInUse (*self*) → bool

> **Parameters self** (*Spinnaker::IImage **) –

IsIncomplete (*self*) → bool

> **Parameters self** (*Spinnaker::IImage const **) –

Release (*self*)

> **Parameters self** (*Spinnaker::IImage **) –

ResetImage (*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

> **Parameters**
>
> - **width** (*size_t*) –
> - **height** (*size_t*) –
> - **offsetX** (*size_t*) –
> - **offsetY** (*size_t*) –
> - **pixelFormat** (*enum Spinnaker::PixelFormatEnums*) –
> - **width, height, offsetX, offsetY, pixelFormat, pData)** (ResetImage (*self,*) –
> - **width** –
> - **height** –
> - **offsetX** –
> - **offsetY** –
> - **pixelFormat** –
> - **pData** (*void **) –

Save (*self*, *pFilename*, *format*)

> **Parameters**
>
> - **pFilename** (*char const **) –
> - **format** (*enum Spinnaker::ImageFileFormat*) –
> - **pFilename)** (Save (*self,*) –
> - **pFilename** –
> - **pFilename, pOption)** (Save (*self,*) –
> - **pFilename** –
> - **pOption** (*Spinnaker::BMPOption &*) –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –
> - **pFilename, pOption)** –
> - **pFilename** –
> - **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

thisown
> The membership flag

class `PySpin.`**`IImageEvent`**(*args*, ***kwargs*)
> Bases: *`PySpin.Event`*

Proxy of C++ Spinnaker::IImageEvent class.

OnImageEvent(*self*, *image*)

> **Parameters image** (*Spinnaker::ImagePtr*) –

thisown
> The membership flag

class `PySpin.`**`IInteger`**(*args*, ***kwargs*)
> Bases: *`PySpin.IValue`*

Proxy of C++ Spinnaker::GenApi::IInteger class.

GetInc(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IInteger **) –

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

> **Parameters self** (*Spinnaker::GenApi::IInteger **) –

GetListOfValidValues(*self*, *bounded=True*) → int64_autovector_t

> **Parameters**
>
> - **bounded** (*bool*) –
>
> - **-> int64_autovector_t** (*GetListOfValidValues(self)*) –
>
> - **self** (*Spinnaker::GenApi::IInteger **) –

GetMax(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IInteger **) –

GetMin(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IInteger **) –

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

> Parameters **self** (*Spinnaker::GenApi::IInteger **) –

GetUnit(*self*) → gcstring

> Parameters **self** (*Spinnaker::GenApi::IInteger **) –

GetValue(*self*, *Verify=False*, *IgnoreCache=False*) → int64_t

> Parameters
>
> - **Verify** (*bool*) –
> - **IgnoreCache** (*bool*) –
> - **Verify=False) -> int64_t** (*GetValue(self,*) –
> - **Verify** –
> - **-> int64_t** (*GetValue(self)*) –
> - **self** (*Spinnaker::GenApi::IInteger **) –

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*) –
> - **Value)** (*SetValue(self,*) –
> - **Value** –

thisown
> The membership flag

class PySpin.**IInterfaceEvent**(**args*, ***kwargs*)
> Bases: *PySpin.IArrivalEvent*, *PySpin.IRemovalEvent*

> Proxy of C++ Spinnaker::IInterfaceEvent class.

> **OnDeviceArrival**(*self*, *serialNumber*)

> > Parameters **serialNumber** (*uint64_t*) –

> **OnDeviceRemoval**(*self*, *serialNumber*)

> > Parameters **serialNumber** (*uint64_t*) –

> **thisown**
> > The membership flag

class PySpin.**ILoggingEvent**(**args*, ***kwargs*)
> Bases: *PySpin.Event*

> Proxy of C++ Spinnaker::ILoggingEvent class.

> **OnLogEvent**(*self*, *eventPtr*)

> Parameters **eventPtr** (*Spinnaker::LoggingEventDataPtr*) –

thisown
> The membership flag

class PySpin.**INode**(**args*, ***kwargs*)
> Bases: *PySpin.ISelector*, *PySpin.IReference*

> Proxy of C++ Spinnaker::GenApi::INode class.

> **DeregisterCallback**(*self*, *hCallback*) → bool

>> Parameters **hCallback** (*Spinnaker::GenApi::CallbackHandleType*) –

> **GetAlias**(*self*) → INode

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetCachingMode**(*self*) → Spinnaker::GenApi::ECachingMode

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetCastAlias**(*self*) → INode

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetChildren**(*self*, *Children*, *LinkType*)

>> Parameters

>>> • **Children** (*Spinnaker::GenApi::NodeList_t &*) –

>>> • **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –

>>> • **Children)** (GetChildren *(self,)* –

>>> • **Children** –

> **GetDescription**(*self*) → gcstring

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetDeviceName**(*self*) → gcstring

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetDisplayName**(*self*) → gcstring

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetDocuURL**(*self*) → gcstring

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetEventID**(*self*) → gcstring

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

> **GetName**(*self*, *FullQualified=False*) → gcstring

>> Parameters

>>> • **FullQualified** (*bool*) –

>>> • **-> gcstring** (GetName *(self)*) –

>>> • **self** (*Spinnaker::GenApi::INode const **) –

> **GetNameSpace**(*self*) → Spinnaker::GenApi::ENameSpace

>> Parameters **self** (*Spinnaker::GenApi::INode const **) –

GetNodeMap(*self*) → INodeMap

      Parameters self(*Spinnaker::GenApi::INode const **) –

GetParents(*self*, *Parents*)

      Parameters Parents(*Spinnaker::GenApi::NodeList_t &*) –

GetPollingTime(*self*) → int64_t

      Parameters self(*Spinnaker::GenApi::INode const **) –

GetPrincipalInterfaceType(*self*) → Spinnaker::GenApi::EInterfaceType

      Parameters self(*Spinnaker::GenApi::INode const **) –

GetProperty(*self*, *PropertyName*, *ValueStr*, *AttributeStr*) → bool

      Parameters

          • **PropertyName**(*Spinnaker::GenICam::gcstring const &*) –

          • **ValueStr**(*Spinnaker::GenICam::gcstring &*) –

          • **AttributeStr**(*Spinnaker::GenICam::gcstring &*) –

GetPropertyNames(*self*)

      Parameters self(*Spinnaker::GenApi::INode const **) –

GetToolTip(*self*) → gcstring

      Parameters self(*Spinnaker::GenApi::INode const **) –

GetVisibility(*self*) → Spinnaker::GenApi::EVisibility

      Parameters self(*Spinnaker::GenApi::INode const **) –

ImposeAccessMode(*self*, *ImposedAccessMode*)

      Parameters ImposedAccessMode(*enum Spinnaker::GenApi::EAccessMode*) –

ImposeVisibility(*self*, *ImposedVisibility*)

      Parameters ImposedVisibility(*enum Spinnaker::GenApi::EVisibility*) –

InvalidateNode(*self*)

      Parameters self(*Spinnaker::GenApi::INode **) –

IsAccessModeCacheable(*self*) → Spinnaker::GenApi::EYesNo

      Parameters self(*Spinnaker::GenApi::INode const **) –

IsCachable(*self*) → bool

      Parameters self(*Spinnaker::GenApi::INode const **) –

IsDeprecated(*self*) → bool

      Parameters self(*Spinnaker::GenApi::INode const **) –

IsFeature(*self*) → bool

      Parameters self(*Spinnaker::GenApi::INode const **) –

IsStreamable(*self*) → bool

      Parameters self(*Spinnaker::GenApi::INode const **) –

RegisterCallback(*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

        Parameters **pCallback** (*Spinnaker::GenApi::CNodeCallback **) –

thisown
> The membership flag

class 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 &*) –

>>> • **pPort) -> bool** ([Connect](self,)*(self,)*) –

>>> • **pPort** –

> **GetDeviceName** (*self*) → gcstring

>> **Parameters self** (*Spinnaker::GenApi::INodeMap **) –

> **GetNode** (*self*, *Name*) → INode

>> **Parameters Name** (*Spinnaker::GenICam::gcstring const &*) –

> **GetNodes** (*self*)

>> **Parameters self** (*Spinnaker::GenApi::INodeMap const **) –

> **GetNumNodes** (*self*) → uint64_t

>> **Parameters self** (*Spinnaker::GenApi::INodeMap const **) –

> **InvalidateNodes** (*self*)

>> **Parameters self** (*Spinnaker::GenApi::INodeMap const **) –

> **Poll** (*self*, *ElapsedTime*)

>> **Parameters ElapsedTime** (*int64_t*) –

thisown
> The membership flag

class PySpin.**INodeMapDyn**(**args*, ***kwargs*)
> Bases: *PySpin.INodeMap*

> Proxy of C++ Spinnaker::GenApi::INodeMapDyn class.

> **ClearAllNodes** (*self*)

>> **Parameters self** (*Spinnaker::GenApi::INodeMapDyn **) –

> **ExtractIndependentSubtree** (*self*, *XMLData*, *InjectXMLData*, *SubTreeRootNodeName*, *Extract-edSubtree*)

>> **Parameters**

>>> • **XMLData** (*Spinnaker::GenICam::gcstring const &*) –

>>> • **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*) –

>>> • **SubTreeRootNodeName** (*Spinnaker::GenICam::gcstring const &*) –

>>> • **ExtractedSubtree** (*Spinnaker::GenICam::gcstring &*) –

GetSupportedSchemaVersions(*self*)

>> **Parameters self** (*Spinnaker::GenApi::INodeMapDyn **) –

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*)

>> **Parameters**

>> - **XMLFileName** (*Spinnaker::GenICam::gcstring const &*) –

>> - **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*) –

>> - **OutputFileName** (*Spinnaker::GenICam::gcstring const &*) –

>> - **XMLValidation** (*uint32_t const*) –

>> - **XMLFileName, StyleSheetFileName, OutputFileName)** ([PreprocessXMLFromFile*(self,*)](#) –

>> - **XMLFileName** –

>> - **StyleSheetFileName** –

>> - **OutputFileName** –

PreprocessXMLFromZIPFile(*self*, *XMLFileName*, *StyleSheetFileName*, *OutputFileName*, *XML-Validation*)

> Parameters
>
> - **XMLFileName** (*Spinnaker::GenICam::gcstring const &*) –
> - **StyleSheetFileName** (*Spinnaker::GenICam::gcstring const &*) –
> - **OutputFileName** (*Spinnaker::GenICam::gcstring const &*) –
> - **XMLValidation** (*uint32_t const*) –
> - **XMLFileName, StyleSheetFileName, OutputFileName)** (*PreprocessXMLFromZIPFile(self,*) –
> - **XMLFileName** –
> - **StyleSheetFileName** –
> - **OutputFileName** –

thisown
> The membership flag

class 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 &*) –

> **thisown**
> > The membership flag

class PySpin.**IReference**(**args*, ***kwargs*)
> Bases: object

> Proxy of C++ Spinnaker::GenApi::IReference class.

> **SetReference**(*self*, *pBase*)
>
> > Parameters **pBase** (*INode **) –

> **thisown**
> > The membership flag

class PySpin.**IRegister**(**args*, ***kwargs*)
> Bases: *PySpin.IValue*

> Proxy of C++ Spinnaker::GenApi::IRegister class.

> **Get**(*self*, *pBuffer*, *Verify=False*, *IgnoreCache=False*)
>
> > Parameters
> >
> > - **pBuffer** (*uint8_t **) –
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **pBuffer, Verify=False)** (*Get(self,*) –

- **pBuffer** –

- **Verify** –

- **pBuffer)** `(Get(self,)` –

- **pBuffer** –

Gets a NumPy array representing the contents of the register, as 8-bit unsigned ints.

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

> **Parameters self** (*Spinnaker::GenApi::IRegister **) –

GetLength (*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IRegister **) –

Set (*self*, *pBuffer*, *Verify=True*)

> **Parameters**
>
> - **pBuffer** (*uint8_t const **) –
>
> - **Verify** (*bool*) –
>
> - **pBuffer)** `(Set(self,)` –
>
> - **pBuffer** –

Set the register's contents with the contents (as 8-bit unsigned ints) of the given array.

pBuffer: The NumPy array containing the data to set

Verify: Enables AccessMode and Range verification (default = true)

thisown
> The membership flag

class PySpin.**IRemovalEvent** (**args*, ***kwargs*)
> Bases: *PySpin.Event*

Proxy of C++ Spinnaker::IRemovalEvent class.

OnDeviceRemoval (*self*, *serialNumber*)

> **Parameters serialNumber** (*uint64_t*) –

thisown
> The membership flag

class PySpin.**ISelector** (**args*, ***kwargs*)
> Bases: *PySpin.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
>
>> Parameters **self** (*Spinnaker::GenApi::ISelector const **) –
>
> **thisown**
>> The membership flag

class PySpin.**ISelectorDigit**(**args*, ***kwargs*)
> Bases: object

> Proxy of C++ Spinnaker::GenApi::ISelectorDigit class.

> **GetSelectorList**(*self*, *Incremental=False*)
>
>> Parameters
>>
>>> • **Incremental** (*bool*) –
>>>
>>> • **GetSelectorList(self)** –
>>>
>>> • **self** (*Spinnaker::GenApi::ISelectorDigit **) –
>
> **Restore**(*self*)
>
>> Parameters **self** (*Spinnaker::GenApi::ISelectorDigit **) –
>
> **SetFirst**(*self*) → bool
>
>> Parameters **self** (*Spinnaker::GenApi::ISelectorDigit **) –
>
> **SetNext**(*self*, *Tick=True*) → bool
>
>> Parameters
>>
>>> • **Tick** (*bool*) –
>>>
>>> • **-> bool** (*SetNext(self)*) –
>>>
>>> • **self** (*Spinnaker::GenApi::ISelectorDigit **) –
>
> **ToString**(*self*) → gcstring
>
>> Parameters **self** (*Spinnaker::GenApi::ISelectorDigit **) –
>
> **thisown**
>> The membership flag

class PySpin.**IString**(**args*, ***kwargs*)
> Bases: *PySpin.IValue*

> Proxy of C++ Spinnaker::GenApi::IString class.

> **GetMaxLength**(*self*) → int64_t
>
>> Parameters **self** (*Spinnaker::GenApi::IString **) –
>
> **GetValue**(*self*, *Verify=False*, *IgnoreCache=False*) → gcstring
>
>> Parameters
>>
>>> • **Verify** (*bool*) –
>>>
>>> • **IgnoreCache** (*bool*) –
>>>
>>> • **Verify=False) -> gcstring** (*GetValue(self,)*) –
>>>
>>> • **Verify** –
>>>
>>> • **-> gcstring** (*GetValue(self)*) –

- **self** (*Spinnaker::GenApi::IString **) –

SetValue (*self*, *Value*, *Verify=True*)

    Parameters

- **Value** (*Spinnaker::GenICam::gcstring const &*) –

- **Verify** (*bool*) –

- **Value)** ([SetValue](*self,*)) –

- **Value** –

thisown

    The membership flag

class PySpin.**ISystem** (**args*, ***kwargs*)

    Bases: object

    Proxy of C++ Spinnaker::ISystem class.

    GetCameras (*self*, *updateInterfaces=True*, *updateCameras=True*) → CameraList

        Parameters

- **updateInterfaces** (*bool*) –

- **updateCameras** (*bool*) –

- **updateInterfaces=True) -> CameraList** ([GetCameras](*self,*)) –

- **updateInterfaces** –

- **-> CameraList** ([GetCameras](*self*)) –

- **self** (*Spinnaker::ISystem **) –

    GetInterfaces (*self*, *updateInterface=True*) → InterfaceList

        Parameters

- **updateInterface** (*bool*) –

- **-> InterfaceList** ([GetInterfaces](*self*)) –

- **self** (*Spinnaker::ISystem **) –

    GetLoggingEventPriorityLevel (*self*) → Spinnaker::SpinnakerLogLevel

        Parameters **self** (*Spinnaker::ISystem **) –

    IsInUse (*self*) → bool

        Parameters **self** (*Spinnaker::ISystem **) –

    RegisterInterfaceEvent (*self*, *evtToRegister*, *updateInterface=True*)

        Parameters

- **evtToRegister** (*Spinnaker::Event &*) –

- **updateInterface** (*bool*) –

- **evtToRegister)** ([RegisterInterfaceEvent](*self,*)) –

- **evtToRegister** –

    RegisterLoggingEvent (*self*, *handler*)

        Parameters **handler** (*Spinnaker::LoggingEvent &*) –

ReleaseInstance(*self*)

> **Parameters self** (*Spinnaker::ISystem **) –

SendActionCommand(*self, deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None, results=0*)

> **Parameters**
>
> - **deviceKey** (*unsigned int*) –
> - **groupKey** (*unsigned int*) –
> - **groupMask** (*unsigned int*) –
> - **actionTime** (*unsigned long long*) –
> - **pResultSize** (*unsigned int **) –
> - **results** (*Spinnaker::ActionCommandResult []*) –
> - **deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **pResultSize** –
> - **deviceKey, groupKey, groupMask, actionTime=0)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **deviceKey, groupKey, groupMask)** (SendActionCommand*(self,)* –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –

SetLoggingEventPriorityLevel(*self, level*)

> **Parameters level** (*enum Spinnaker::SpinnakerLogLevel*) –

UnregisterAllLoggingEvent(*self*)

> **Parameters self** (*Spinnaker::ISystem **) –

UnregisterInterfaceEvent(*self, evtToUnregister*)

> **Parameters evtToUnregister** (*Spinnaker::Event &*) –

UnregisterLoggingEvent(*self, handler*)

> **Parameters handler** (*Spinnaker::LoggingEvent &*) –

UpdateCameras(*self, updateInterfaces=True*) → bool

Parameters

> - **updateInterfaces** (*bool*) –
> - **-> bool** (*UpdateCameras(self)*) –
> - **self** (*Spinnaker::ISystem **) –

thisown
> The membership flag

class PySpin.**IValue**(**args*, ***kwargs*)
> Bases: *PySpin.INode*

> Proxy of C++ Spinnaker::GenApi::IValue class.

> **FromString**(*self*, *ValueStr*, *Verify=True*)

>> Parameters

>> - **ValueStr** (*Spinnaker::GenICam::gcstring const &*) –
>> - **Verify** (*bool*) –
>> - **ValueStr)** (*FromString(self,*) –
>> - **ValueStr** –

> **GetNode**(*self*) → INode

>> Parameters **self** (*Spinnaker::GenApi::IValue **) –

> **IsValueCacheValid**(*self*) → bool

>> Parameters **self** (*Spinnaker::GenApi::IValue const **) –

> **ToString**(*self*, *Verify=False*, *IgnoreCache=False*) → gcstring

>> Parameters

>> - **Verify** (*bool*) –
>> - **IgnoreCache** (*bool*) –
>> - **Verify=False) -> gcstring** (*ToString(self,*) –
>> - **Verify** –
>> - **-> gcstring** (*ToString(self)*) –
>> - **self** (*Spinnaker::GenApi::IValue **) –

thisown
> The membership flag

class PySpin.**Image**(**args*, ***kwargs*)
> Bases: *PySpin.IImage*

> The image object class.

> C++ includes: Image.h

> **CalculateChannelStatistics**(*self*, *channel*) → ChannelStatistics

>> Parameters **channel** (*enum Spinnaker::StatisticsChannel*) –

> Returns a ChannelStatistics instance for the current image on a given channel.

> channel: Channel to generate statistics on.

CheckCRC (*self*) → bool

> **Parameters self** (*Spinnaker::Image const **) –

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.

Convert (*self*, *format*, *algorithm*) → ImagePtr

> **Parameters**
>
> - **format** (*enum Spinnaker::PixelFormatEnums*) –
> - **algorithm** (*enum Spinnaker::ColorProcessingAlgorithm*) –
> - **format) -> ImagePtr** (Convert (*self*,) –
> - **format** –

ImagePtr Spinnaker::Image::Convert(Spinnaker::PixelFormatEnums format, ColorProcessingAlgorithm algorithm=DEFAULT) const

Converts the current image buffer to the specified output pixel format and stores the result in the specified image. The destination image does not need to be configured in any way before the call is made.

See: PixelFormatEnums

format: Output format of the converted image.

algorithm: processing algorithm for producing the converted image

The converted image.

static Create () → ImagePtr
Create(image) -> ImagePtr

> **Parameters**
>
> - **image** (*Spinnaker::ImagePtr const*) –
> - **height, offsetX, offsetY, pixelFormat, pData) -> ImagePtr** (Create (width,) –
> - **width** (*size_t*) –
> - **height** (*size_t*) –
> - **offsetX** (*size_t*) –
> - **offsetY** (*size_t*) –
> - **pixelFormat** (*enum Spinnaker::PixelFormatEnums*) –
> - **pData** (*void **) –

Creates a new Image object, either using a default constructor, copied from another ImagePtr, or using width, height, offset_x, offset_y, pixel format, and a NumPy array containing 8-bit unsigned ints representing the image data (replaces the void* pData argument).

DeepCopy (*self*, *pSrcImage*)

> **Parameters pSrcImage** (*Spinnaker::ImagePtr const*) –

void Spinnaker::Image::DeepCopy(const ImagePtr pSrcImage)

Performs a deep copy of the Image. After this operation, the image contents and member variables will be the same. The Images will not share a buffer. The Image's current buffer will not be released.

pSrcImage: The Image to copy the data from.

GetBitsPerPixel (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

const ChunkData& Spinnaker::Image::GetChunkData() const

Returns a pointer to a chunk data interface. No ownership is transfered, 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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

ColorProcessingAlgorithm Spinnaker::Image::GetColorProcessing() const

Gets the algorithm used to produce the image.

See: Convert()

The color processing algorithm used to produce the image.

static GetDefaultColorProcessing () → Spinnaker::ColorProcessingAlgorithm

GetFrameID (*self*) → uint64_t

> **Parameters self** (*Spinnaker::Image const **) –

uint64_t Spinnaker::Image::GetFrameID() const

Gets the frame ID for this image.

The frame ID.

GetHeight (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

uint64_t Spinnaker::Image::GetID() const

Gets a unique ID for this image. Each image in a steam will have a unique ID to help identify it.

The 64 bit unique id for this image.

GetImageSize (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

size_t Spinnaker::Image::GetImageSize() const

Returns the size of the image

The image size in bytes.

GetImageStatus (*self*) → Spinnaker::ImageStatus

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

GetPayloadType (*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

GetPixelFormatName(*self*) → gcstring

> **Parameters self** (*Spinnaker::Image const **) –

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 *

> **Parameters self** (*Spinnaker::Image const **) –

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.

GetStride(*self*) → size_t

> **Parameters self** (*Spinnaker::Image const **) –

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::PayloadTypeInfoIDs

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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::PixelFormatNamespaceID

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

> **Parameters self** (*Spinnaker::Image const **) –

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

>   **Parameters self** (*Spinnaker::Image const **) –

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

>   **Parameters self** (*Spinnaker::Image const **) –

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

>   **Parameters self** (*Spinnaker::Image const **) –

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

>   **Parameters self** (*Spinnaker::Image const **) –

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.

IsInUse (*self*) → bool

>   **Parameters self** (*Spinnaker::Image **) –

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

>   **Parameters self** (*Spinnaker::Image const **) –

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.

Release (*self*)

> **Parameters self** (*Spinnaker::Image **) –

void Spinnaker::Image::Release()

ResetImage (*self*, *width*, *height*, *offsetX*, *offsetY*, *pixelFormat*)

> **Parameters**
>
> - **width** (*size_t*) –
> - **height** (*size_t*) –
> - **offsetX** (*size_t*) –
> - **offsetY** (*size_t*) –
> - **pixelFormat** (*enum Spinnaker::PixelFormatEnums*) –
> - **width, height, offsetX, offsetY, pixelFormat, pData)**
>   (ResetImage(self,) –
> - **width** –
> - **height** –
> - **offsetX** –
> - **offsetY** –
> - **pixelFormat** –
> - **pData** (*void **) –

void Spinnaker::Image::ResetImage(size_t width, size_t height, size_t offsetX, size_t offsetY, Spinnaker::PixelFormatEnums pixelFormat, void *pData)

Sets new dimensions of the image object.

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.

pData: Pointer to the image buffer.

Save (*self*, *pFilename*, *format*)

> **Parameters**
>
> - **pFilename** (*char const **) –
> - **format** (*enum Spinnaker::ImageFileFormat*) –
> - **pFilename)** (Save(self,) –
> - **pFilename** –
> - **pFilename, pOption)** (Save(self,) –
> - **pFilename** –
> - **pOption** (*Spinnaker::BMPOption &*) –
> - **pFilename, pOption)** –
> - **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

- **pFilename, pOption)** –

- **pFilename** –

- **pOption** –

void Spinnaker::Image::Save(const char *****pFilename, BMPOption &pOption)

Saves the image to the specified file name with the options specified.

pFilename: Filename to save image with.

pOption: Options to use while saving image.

> ## static **SetDefaultColorProcessing**(*defaultMethod*)
>
> > **Parameters defaultMethod** (*enum Spinnaker::ColorProcessingAlgorithm*) –

thisown
> The membership flag

class PySpin.**ImageEvent**
Bases: *PySpin.IImageEvent*

A handler for capturing image arrival events.

C++ includes: ImageEvent.h

OnImageEvent(*self*, *image*)

> **Parameters image** (*Spinnaker::ImagePtr*) –

virtual void Spinnaker::ImageEvent::OnImageEvent(ImagePtr image)=0

Image event callback

image: The ImagePtr object

thisown
> The membership flag

class PySpin.**ImagePtr**(*args*)

Bases: PySpin._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

thisown

The membership flag

PySpin.**Image_Create**(*args*)

Create() -> ImagePtr Create(image) -> ImagePtr

Parameters

- **image**(*Spinnaker::ImagePtr const*) –
- **height, offsetX, offsetY, pixelFormat, pData) -> ImagePtr** (Image_Create(width,) –
- **width**(*size_t*) –
- **height**(*size_t*) –
- **offsetX**(*size_t*) –
- **offsetY**(*size_t*) –
- **pixelFormat**(*enum Spinnaker::PixelFormatEnums*) –
- **pData**(*void **) –

Creates a new Image object, either using a default constructor, copied from another ImagePtr, or using width, height, offset_x, offset_y, pixel format, and a NumPy array containing 8-bit unsigned ints representing the image data (replaces the void* pData argument).

PySpin.**Image_GetDefaultColorProcessing**() → Spinnaker::ColorProcessingAlgorithm

PySpin.**Image_GetImageStatusDescription**(*status*) → char const *

Parameters status(*enum Spinnaker::ImageStatus*) –

PySpin.**Image_SetDefaultColorProcessing**(*defaultMethod*)

Parameters defaultMethod(*enum Spinnaker::ColorProcessingAlgorithm*) –

class PySpin.**IntRegNode**(*args*, **kwargs*)

Bases: *PySpin.IntegerNode*, *PySpin.RegisterNode*

Interface for string properties.

C++ includes: IntRegNode.h

SetReference(*self*, *pBase*)

Parameters pBase(*Spinnaker::GenApi::INode **) –

virtual void Spinnaker::GenApi::IntRegNode::SetReference(INode *pBase)

overload SetReference for Value

thisown

The membership flag

class PySpin.**IntegerNode**(*args*, **kwargs*)

Bases: *PySpin.IInteger*, *PySpin.ValueNode*

Interface for string properties.

C++ includes: IntegerNode.h

GetFloatAlias(*self*) → IFloat

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual IFloat* Spinnaker::GenApi::IntegerNode::GetFloatAlias()

> gets the interface of an alias node.

GetInc(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual int64_t Spinnaker::GenApi::IntegerNode::GetInc()

> Get increment

GetIncMode(*self*) → Spinnaker::GenApi::EIncMode

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual EIncMode Spinnaker::GenApi::IntegerNode::GetIncMode()

> Get increment mode

GetListOfValidValues(*self*, *bounded=True*) → int64_autovector_t

> **Parameters**

> > • **bounded** (*bool*) –

> > • **-> int64_autovector_t** (GetListOfValidValues(self)) –

> > • **self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual int64_autovector_t Spinnaker::GenApi::IntegerNode::GetListOfValidValues(bool bounded=true)

> Get list of valid value

GetMax(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual int64_t Spinnaker::GenApi::IntegerNode::GetMax()

> Get maximum value allowed

GetMin(*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual int64_t Spinnaker::GenApi::IntegerNode::GetMin()

> Get minimum value allowed

GetRepresentation(*self*) → Spinnaker::GenApi::ERepresentation

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

> virtual ERepresentation Spinnaker::GenApi::IntegerNode::GetRepresentation()

> Get recommended representation

GetUnit(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::IntegerNode **) –

virtual GenICam::gcstring Spinnaker::GenApi::IntegerNode::GetUnit()

Get the physical unit name

GetValue (*self*, *Verify=False*, *IgnoreCache=False*) → int64_t

> **Parameters**
>
>> • **Verify** (*bool*) –
>>
>> • **IgnoreCache** (*bool*) –
>>
>> • **Verify=False) -> int64_t** (GetValue *(self,)* –
>>
>> • **Verify** –
>>
>> • **-> int64_t** (GetValue *(self)*) –
>>
>> • **self** (*Spinnaker::GenApi::IntegerNode **) –

virtual int64_t Spinnaker::GenApi::IntegerNode::GetValue(bool Verify=false, bool IgnoreCache=false)

Get node value

Verify: Enables Range verification (default = false). The AccessMode is always checked

IgnoreCache: If true the value is read ignoring any caches (default = false)

The value read

ImposeMax (*self*, *Value*)

> **Parameters Value** (*int64_t*) –

virtual void Spinnaker::GenApi::IntegerNode::ImposeMax(int64_t Value)

Restrict maximum value

ImposeMin (*self*, *Value*)

> **Parameters Value** (*int64_t*) –

virtual void Spinnaker::GenApi::IntegerNode::ImposeMin(int64_t Value)

Restrict minimum value

SetReference (*self*, *pBase*)

> **Parameters pBase** (*Spinnaker::GenApi::INode **) –

virtual void Spinnaker::GenApi::IntegerNode::SetReference(INode *pBase)

overload SetReference for Integer

SetValue (*self*, *Value*, *Verify=True*)

> **Parameters**
>
>> • **Value** (*int64_t*) –
>>
>> • **Verify** (*bool*) –
>>
>> • **Value)** (SetValue *(self,)* –
>>
>> • **Value** –

virtual void Spinnaker::GenApi::IntegerNode::SetValue(int64_t Value, bool Verify=true)

Set node value

Value: The value to set

Verify: Enables AccessMode and Range verification (default = true)

thisown
>   The membership flag

class PySpin.**Interface**(**args*, ***kwargs*)
>   Bases: object

An interface object which holds a list of cameras.

C++ includes: Interface.h

GetCameras(*self*, *updateCameras=True*) → CameraList

>   **Parameters**
>
>   - **updateCameras** (*bool*) –
>
>   - **-> CameraList** (GetCameras *(self)*) –
>
>   - **self** (*Spinnaker::Interface const **) –

CameraList Spinnaker::Interface::GetCameras(bool updateCameras=true) const

Returns a list of cameras available on this interface. This call returns either usb3 vision or gige vision cameras depending on the underlying transport layer of this interface. The camera list object will reference count the cameras that it holds. It is important that the CameraList is destroyed or is cleared before System::ReleaseInstance() can be called or an InterfaceList that holds this interface can be cleared.

See: System::ReleaseInstance()

See: InterfaceList::Clear()

See: CameraList::Clear()

updateCameras: A flag used to issue an updateCameras() call internally before getting the camera list

An CameraList object that contains a list of cameras on this interface.

GetTLNodeMap(*self*) → INodeMap

>   **Parameters self** (*Spinnaker::Interface const **) –

GenApi::INodeMap& Spinnaker::Interface::GetTLNodeMap() const

Gets a nodeMap that is generated from a GenICam XML file for the GenTL interface Module.

A reference to a INodeMap object.

IsInUse(*self*) → bool

>   **Parameters self** (*Spinnaker::Interface const **) –

bool Spinnaker::Interface::IsInUse() const

Checks if the interface is in use by any camera objects

Returns true if the interface is in use and false otherwise.

RegisterEvent(*self*, *evtToRegister*)

>   **Parameters evtToRegister** (*Spinnaker::Event &*) –

void Spinnaker::Interface::RegisterEvent(Event &evtToRegister)

Registers an event for the interface

evtToRegister: The event to register for the interface

SendActionCommand(*self*, *deviceKey*, *groupKey*, *groupMask*, *actionTime=0*, *pResultSize=None*, *results=0*)

> **Parameters**
>
> - **deviceKey** (`unsigned int`) –
> - **groupKey** (`unsigned int`) –
> - **groupMask** (`unsigned int`) –
> - **actionTime** (`unsigned long long`) –
> - **pResultSize** (`unsigned int *`) –
> - **results** (`Spinnaker::ActionCommandResult []`) –
> - **deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None)** (SendActionCommand`(self,)`) –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **pResultSize** –
> - **deviceKey, groupKey, groupMask, actionTime=0)** (SendActionCommand`(self,)`) –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **deviceKey, groupKey, groupMask)** (SendActionCommand`(self,)`) –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –

void Spinnaker::Interface::SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL) const

Broadcast an Action Command to all devices on interface

deviceKey: The Action Command's device key

groupKey: The Action Command's group key

groupMask: The Action Command's group mask

actionTime: (Optional) Time when to assert a future action. Zero means immediate action.

pResultSize: (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.

results: (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected

---

number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

TLInterface
> Interface_TLInterface_get(self) -> TransportLayerInterface

> > **Parameters self** (*Spinnaker::Interface **) –

UnregisterEvent(*self*, *evtToUnregister*)

> > **Parameters evtToUnregister** (*Spinnaker::Event &*) –

> void Spinnaker::Interface::UnregisterEvent(Event &evtToUnregister)

> Unregisters an event for the interface

> evtToUnregister: The event to unregister from the interface

UpdateCameras(*self*) → bool

> > **Parameters self** (*Spinnaker::Interface **) –

> bool Spinnaker::Interface::UpdateCameras()

> Updates the list of cameras on this interface. This function needs to be called before any cameras can be discovered using GetCameras(). System::GetCameras() will automatically call this function for each interface it enumerates. If the list changed after the last time System::GetCameras() or UpdateCameras() was called then the return value will be true, otherwise it is false.

> See: System::GetCameras()

> See: GetCameras()

> true if cameras changed on interface and false otherwise.

thisown
> The membership flag

class PySpin.**InterfaceEvent**
> Bases: *PySpin.IInterfaceEvent*

> A handler to device arrival and removal events on all interfaces.

> C++ includes: InterfaceEvent.h

> **OnDeviceArrival**(*self*, *serialNumber*)

> > > **Parameters serialNumber** (*uint64_t*) –

> > virtual void Spinnaker::InterfaceEvent::OnDeviceArrival(uint64_t serialNumber)=0

> > Device arrival event callback.

> **OnDeviceRemoval**(*self*, *serialNumber*)

> > > **Parameters serialNumber** (*uint64_t*) –

> > virtual void Spinnaker::InterfaceEvent::OnDeviceRemoval(uint64_t serialNumber)=0

> > Callback to the device removal event.

> > serialNumber: The serial number of the removed device

> **thisown**
> > The membership flag

class PySpin.**InterfaceList**(**args*)

    Bases: object

A list of the available interfaces on the system.

C++ includes: InterfaceList.h

Clear(*self*)

        Parameters self (*Spinnaker::InterfaceList **) –

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 (*int*) –

InterfacePtr Spinnaker::InterfaceList::GetByIndex(int index) const

Returns a pointer to an Interface object at the "index".

index: The index at which to retrieve the Interface object

A pointer to an Interface object.

GetSize(*self*) → int

        Parameters self (*Spinnaker::InterfaceList const **) –

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.

thisown

    The membership flag

class PySpin.**InterfacePtr**(**args*)

    Bases: PySpin._SWIG_IFacePtr

A reference tracked pointer to the interface object.

C++ includes: InterfacePtr.h

thisown

    The membership flag

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 Spinnaker::GenApi::IsAvailable(const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is Available

PySpin.**IsCacheable**(*CachingMode*) → bool

> Parameters **CachingMode** (*enum Spinnaker::GenApi::ECachingMode*) –

bool Spinnaker::GenApi::IsCacheable(ECachingMode CachingMode)

Tests Cacheability

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 Spinnaker::GenApi::IsImplemented(const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is Implemented

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 Spinnaker::GenApi::IsReadable(const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is readable

PySpin.**IsVisible**(*Visibility*, *MaxVisiblity*) → bool

   Parameters

- **Visibility** (*enum Spinnaker::GenApi::EVisibility*) –

- **MaxVisiblity** (*enum Spinnaker::GenApi::EVisibility*) –

bool Spinnaker::GenApi::IsVisible(EVisibility Visibility, EVisibility MaxVisiblity)

Tests Visibility CAVE : this relies on the EVisibility enum's coding

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 Spinnaker::GenApi::IsWritable(const Spinnaker::GenApi::CPointer< T, B > &ptr)

Checks if a node is Writable

class PySpin.**JPEGOption**
> Bases: `object`

Options for saving JPEG image.

C++ includes: SpinnakerDefs.h

progressive
> JPEGOption_progressive_get(self) -> bool

> > Parameters **self** (*Spinnaker::JPEGOption* *) –

quality
> JPEGOption_quality_get(self) -> unsigned int

> > Parameters **self** (*Spinnaker::JPEGOption* *) –

reserved
> JPEGOption_reserved_get(self) -> unsigned int [16]

> > Parameters **self** (*Spinnaker::JPEGOption* *) –

thisown
> The membership flag

class PySpin.**JPG2Option**
> Bases: `object`

Options for saving JPEG2000 image.

C++ includes: SpinnakerDefs.h

quality
> JPG2Option_quality_get(self) -> unsigned int

> > Parameters **self** (*Spinnaker::JPG2Option* *) –

reserved
> JPG2Option_reserved_get(self) -> unsigned int [16]

> > Parameters **self** (*Spinnaker::JPG2Option* *) –

thisown
> The membership flag

class PySpin.**LoggingEvent**
> Bases: *PySpin.ILoggingEvent*

An event handler for capturing the device logging event.

C++ includes: LoggingEvent.h

OnLogEvent(*self*, *eventPtr*)

> > Parameters **eventPtr** (*Spinnaker::LoggingEventDataPtr*) –

> virtual void Spinnaker::LoggingEvent::OnLogEvent(LoggingEventDataPtr eventPtr)=0

> The callback for the log event.

> eventPtr: The logging event pointer

thisown
> The membership flag

class PySpin.**LoggingEventData**(**args*, ***kwargs*)

Bases: object

The LoggingEventData object.

C++ includes: LoggingEventData.h

GetCategoryName(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetCategoryName()

Gets the logging event category name.

The category name

GetLogMessage(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetLogMessage()

Gets the logging event message.

The log message

GetNDC(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetNDC()

Gets the logging event's Nested Diagnostic Context (NDC).

The log event's NDC

GetPriority(*self*) → int const

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const int Spinnaker::LoggingEventData::GetPriority()

Gets the logging event priority.

The log priority

GetPriorityName(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetPriorityName()

Gets the logging event priority name.

The priority name of the log

GetThreadName(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetThreadName()

Gets the logging event thread name.

The thread name

GetTimestamp(*self*) → char const *

> **Parameters self** (*Spinnaker::LoggingEventData **) –

const char* Spinnaker::LoggingEventData::GetTimestamp()

Gets the logging event time stamp.

The time stamp of the log

thisown
The membership flag

class PySpin.**LoggingEventDataPtr**(**args*)
Bases: PySpin._SWIG_LogPtr

A reference tracked pointer to the LoggingEvent object.

C++ includes: LoggingEventDataPtr.h

thisown
The membership flag

class PySpin.**MJPGOption**
Bases: object

Options for saving MJPG files.

C++ includes: SpinnakerDefs.h

frameRate
MJPGOption_frameRate_get(self) -> float

Parameters self (*Spinnaker::MJPGOption **) –

quality
MJPGOption_quality_get(self) -> unsigned int

Parameters self (*Spinnaker::MJPGOption **) –

reserved
MJPGOption_reserved_get(self) -> unsigned int [256]

Parameters self (*Spinnaker::MJPGOption **) –

thisown
The membership flag

class PySpin.**Node**(**args*, ***kwargs*)
Bases: *PySpin.INode*

class common to all nodes

C++ includes: Node.h

DeregisterCallback(*self*, *hCallback*) → bool

Parameters hCallback (*Spinnaker::GenApi::CallbackHandleType*) –

virtual bool Spinnaker::GenApi::Node::DeregisterCallback(CallbackHandleType hCallback)

De register change callback Destroys CNodeCallback object true if the callback handle was valid

GetAccessMode(*self*) → Spinnaker::GenApi::EAccessMode

Parameters self (*Spinnaker::GenApi::Node const **) –

virtual EAccessMode Spinnaker::GenApi::Node::GetAccessMode() const

Base interface overrides.

Get the access mode of the node

GetAlias (*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

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

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual ECachingMode Spinnaker::GenApi::Node::GetCachingMode() const

Get Caching Mode

GetCastAlias (*self*) → INode

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

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*)

> **Parameters**
>
> > • **LinkType** (*enum Spinnaker::GenApi::ELinkType*) –
> >
> > • **GetChildren(self)** –
> >
> > • **self** (*Spinnaker::GenApi::Node const **) –

virtual void Spinnaker::GenApi::Node::GetChildren(GenApi::NodeList_t &Children, ELinkType Link-
Type=ctReadingChildren) const

Get all nodes this node directly depends on.

Children: List of children nodes

LinkType: The link type

GetDescription (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDescription() const

Get a long description of the node

GetDeviceName (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDeviceName() const

Get a name of the device

GetDisplayName (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDisplayName() const

Get a name string for display

GetDocuURL (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetDocuURL() const

Gets a URL pointing to the documentation of that feature

GetEventID (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetEventID() const

Get the EventId of the node

GetName (*self*, *FullQualified=False*) → gcstring

> **Parameters**
>
> > • **FullQualified** (*bool*) –
> >
> > • **-> gcstring** (*GetName(self)*) –
> >
> > • **self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetName(bool FullQualified=false) const

Get node name

GetNameSpace (*self*) → Spinnaker::GenApi::ENameSpace

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenApi::ENameSpace Spinnaker::GenApi::Node::GetNameSpace() const

Get name space

GetNodeHandle (*self*) → std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

std::shared_ptr<Node::NodeImpl> Spinnaker::GenApi::Node::GetNodeHandle() const

Get Node handle

GetNodeMap (*self*) → INodeMap

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual INodeMap* Spinnaker::GenApi::Node::GetNodeMap() const

Retrieves the central node map

GetParents (*self*)

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual void Spinnaker::GenApi::Node::GetParents(GenApi::NodeList_t &Parents) const

Gets all nodes this node is directly depending on.

Parents: List of parent nodes

GetPollingTime (*self*) → int64_t

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual int64_t Spinnaker::GenApi::Node::GetPollingTime() const

recommended polling time (for not cacheable nodes)

GetPrincipalInterfaceType (*self*) → Spinnaker::GenApi::EInterfaceType

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

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 bool Spinnaker::GenApi::Node::GetProperty(const GenICam::gcstring &PropertyName, GenICam::gcstring &ValueStr, GenICam::gcstring &AttributeStr)

Retrieves a property plus an additional attribute by name If a property has multiple values/attribute they come with Tabs as delimiters

GetPropertyNames (*self*)

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual void Spinnaker::GenApi::Node::GetPropertyNames(GenICam::gcstring_vector &PropertyNames) const

Returns a list of the names all properties set during initialization

GetSelectedFeatures (*self*)

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual void Spinnaker::GenApi::Node::GetSelectedFeatures(FeatureList_t &) const

retrieve the group of selected features

GetSelectingFeatures (*self*)

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual void Spinnaker::GenApi::Node::GetSelectingFeatures(FeatureList_t &) const

retrieve the group of features selecting this node

GetToolTip (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual GenICam::gcstring Spinnaker::GenApi::Node::GetToolTip() const

Get a short description of the node

GetVisibility (*self*) → Spinnaker::GenApi::EVisibility

> **Parameters self** (*Spinnaker::GenApi::Node const **) –

virtual EVisibility Spinnaker::GenApi::Node::GetVisibility() const

Get the recommended visibility of the node

ImposeAccessMode (*self*, *ImposedAccessMode*)

> **Parameters ImposedAccessMode** (*enum Spinnaker::GenApi::EAccessMode*) –

virtual void Spinnaker::GenApi::Node::ImposeAccessMode(EAccessMode ImposedAccessMode)

Imposes an access mode to the natural access mode of the node

ImposeVisibility (*self*, *ImposedVisibility*)

Parameters ImposedVisibility (*enum Spinnaker::GenApi::EVisibility*) –

virtual void Spinnaker::GenApi::Node::ImposeVisibility(EVisibility ImposedVisibility)

Imposes a visibility to the natural visibility of the node

InvalidateNode (*self*)

> **Parameters self** (*Spinnaker::GenApi::Node **) –
>
> 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

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> virtual EYesNo Spinnaker::GenApi::Node::IsAccessModeCacheable() const
>
> True if the AccessMode can be cached

IsCachable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> virtual bool Spinnaker::GenApi::Node::IsCachable() const
>
> Is the node value cacheable

IsDeprecated (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> virtual bool Spinnaker::GenApi::Node::IsDeprecated() const
>
> True if the node should not be used any more

IsFeature (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> 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

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> virtual bool Spinnaker::GenApi::Node::IsSelector() const
>
> Selector interface overrides.
>
> true if this feature selects a group of features

IsStreamable (*self*) → bool

> **Parameters self** (*Spinnaker::GenApi::Node const **) –
>
> virtual bool Spinnaker::GenApi::Node::IsStreamable() const
>
> True if the node is streamable

RegisterCallback (*self*, *pCallback*) → Spinnaker::GenApi::CallbackHandleType

> **Parameters pCallback** (*Spinnaker::GenApi::CNodeCallback **) –
>
> virtual CallbackHandleType Spinnaker::GenApi::Node::RegisterCallback(CNodeCallback *pCallback)
>
> Register change callback Takes ownership of the CNodeCallback object

SetNodeHandle(*self*, *pNodeHandle*)

> **Parameters pNodeHandle**(*std::shared_ptr< Spinnaker::GenApi::Node::NodeImpl >*) –

void Spinnaker::GenApi::Node::SetNodeHandle(std::shared_ptr< Node::NodeImpl > pNodeHandle)

Set Node handle

SetNodeMap(*self*, *pNodeMap*)

> **Parameters pNodeMap**(*Spinnaker::GenApi::INodeMap **) –

void Spinnaker::GenApi::Node::SetNodeMap(INodeMap *pNodeMap)

SetReference(*self*, *pBase*)

> **Parameters**
>
> - **pBase**(*Spinnaker::GenApi::ISelector **) –
> - **pBase)** (*SetReference(self,)* –
> - **pBase** –

virtual void Spinnaker::GenApi::Node::SetReference(ISelector *pBase)

thisown
> The membership flag

class 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.

node: INode passed to the function during the callback.

thisown
> The membership flag

class PySpin.**NodeMap**(**args*)
> Bases: *PySpin.INodeMap*, *PySpin.IDeviceInfo*

Smart pointer template for NodeMaps with create function.

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 &*) –
> - **pPort) -> bool**(*Connect(self,)* –
> - **pPort** –

virtual bool Spinnaker::GenApi::NodeMap::Connect(IPort *pPort) const

Connects a port to the standard port "Device"

Destroy (*self*)

> **Parameters self** (`Spinnaker::GenApi::NodeMap *`) –

void Spinnaker::GenApi::NodeMap::Destroy()

Destroys the node map

GetDeviceName (*self*) → gcstring

> **Parameters self** (`Spinnaker::GenApi::NodeMap *`) –

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetDeviceName()

Get device name

GetDeviceVersion (*self*, *Version*)

> **Parameters Version** (`Spinnaker::GenICam::Version_t &`) –

virtual void Spinnaker::GenApi::NodeMap::GetDeviceVersion(GenICam::Version_t &Version)

Get the version of the device description file

GetGenApiVersion (*self*, *Version*, *Build*)

> **Parameters**
>
> > • **Version** (`Spinnaker::GenICam::Version_t &`) –
> >
> > • **Build** (`uint16_t &`) –

virtual void Spinnaker::GenApi::NodeMap::GetGenApiVersion(GenICam::Version_t &Version, uint16_t &Build)

Get the version of the DLL's GenApi implementation

GetModelName (*self*) → gcstring

> **Parameters self** (`Spinnaker::GenApi::NodeMap *`) –

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetModelName()

Get the model name

GetNode (*self*, *key*) → INode

> **Parameters key** (`Spinnaker::GenICam::gcstring const &`) –

virtual INode* Spinnaker::GenApi::NodeMap::GetNode(const GenICam::gcstring &key) const

Retrieves the node from the central map by name

GetNodeMapHandle (*self*) → void *

> **Parameters self** (`Spinnaker::GenApi::NodeMap const *`) –

void* Spinnaker::GenApi::NodeMap::GetNodeMapHandle() const

GetNodes (*self*)

> **Parameters self** (`Spinnaker::GenApi::NodeMap const *`) –

virtual void Spinnaker::GenApi::NodeMap::GetNodes(NodeList_t &Nodes) const

Retrieves all nodes in the node map

GetNumNodes (*self*) → uint64_t

> **Parameters self** (*Spinnaker::GenApi::NodeMap const **) –

> virtual uint64_t Spinnaker::GenApi::NodeMap::GetNumNodes() const

> Get the number of nodes in the map

GetProductGuid (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

> virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetProductGuid()

> Get the GUID describing the product

GetSchemaVersion (*self*, *Version*)

> **Parameters Version** (*Spinnaker::GenICam::Version_t &*) –

> virtual void Spinnaker::GenApi::NodeMap::GetSchemaVersion(GenICam::Version_t &Version)

> Get the schema version number

GetStandardNameSpace (*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

> virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetStandardNameSpace()

> Get the standard name space

GetSupportedSchemaVersions (*self*)

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

> virtual void Spinnaker::GenApi::NodeMap::GetSupportedSchemaVersions(GenICam::gcstri 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

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetToolTip()

Get tool tip

GetVendorName(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVendorName()

Get the vendor name

GetVersionGuid(*self*) → gcstring

> **Parameters self** (*Spinnaker::GenApi::NodeMap **) –

virtual GenICam::gcstring Spinnaker::GenApi::NodeMap::GetVersionGuid()

Get the GUID describing the product version

InvalidateNodes(*self*)

> **Parameters self** (*Spinnaker::GenApi::NodeMap const **) –

virtual void Spinnaker::GenApi::NodeMap::InvalidateNodes() const

Invalidates all nodes

LoadXMLFromFile(*self*, *FileName*)

> **Parameters FileName** (*Spinnaker::GenICam::gcstring*) –

void Spinnaker::GenApi::NodeMap::LoadXMLFromFile(GenICam::gcstring FileName)

! Creates the object from the default DLL ! note Can only be used if the class TCameraParams was auto generated from a specific camera xml file void LoadDLL(void);

! Creates the object from a DLL whose name is deduced from vendor and model name void Load-DLL(GenICam::gcstring VendorName, GenICam::gcstring ModelName);

! Creates the object from a DLL with given file name void LoadDLL(GenICam::gcstring FileName); Creates the object from a XML 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, GenICam::gcstring InjectFileName)

Creates the object from a XML target and an inject file with given file name

LoadXMLFromString(*self*, *XMLData*)

> **Parameters XMLData** (*Spinnaker::GenICam::gcstring const &*) –

void Spinnaker::GenApi::NodeMap::LoadXMLFromString(const GenICam::gcstring &XMLData)

Creates the object from XML data given in a string

LoadXMLFromStringInject(*self*, *TargetXMLDataconst*, *InjectXMLData*)

> **Parameters**
>
> > • **TargetXMLDataconst** (*Spinnaker::GenICam::gcstring const &*) –

> • **InjectXMLData** (*Spinnaker::GenICam::gcstring const &*) –

void Spinnaker::GenApi::NodeMap::LoadXMLFromStringInject(const GenICam::gcstring &TargetXMLDataconst, const GenICam::gcstring &InjectXMLData)

Creates the object from XML data given in a string with injection

LoadXMLFromZIPData (*self*, *zipData*, *zipSize*)

> **Parameters**
>
> > • **zipData** (*void const **) –
> >
> > • **zipSize** (*size_t*) –

void Spinnaker::GenApi::NodeMap::LoadXMLFromZIPData(const void *zipData, size_t zipSize)

Creates the object from a ZIP'd XML file given in a string

LoadXMLFromZIPFile (*self*, *ZipFileName*)

> **Parameters ZipFileName** (*Spinnaker::GenICam::gcstring*) –

void Spinnaker::GenApi::NodeMap::LoadXMLFromZIPFile(GenICam::gcstring ZipFileName)

Creates the object from a ZIP'd XML file with given file name

Poll (*self*, *ElapsedTime*)

> **Parameters ElapsedTime** (*int64_t*) –

virtual void Spinnaker::GenApi::NodeMap::Poll(int64_t ElapsedTime)

Fires nodes which have a polling time

thisown
> The membership flag

PySpin.**NodeMap_ClearXMLCache** () → bool

class PySpin.**PGMOption**
> Bases: object

> Options for saving PGM images.

> C++ includes: SpinnakerDefs.h

binaryFile
> PGMOption_binaryFile_get(self) -> bool

> > **Parameters self** (*Spinnaker::PGMOption **) –

reserved
> PGMOption_reserved_get(self) -> unsigned int [16]

> > **Parameters self** (*Spinnaker::PGMOption **) –

thisown
> The membership flag

class PySpin.**PNGOption**
> Bases: object

> Options for saving PNG images.

> C++ includes: SpinnakerDefs.h

compressionLevel
> PNGOption_compressionLevel_get(self) -> unsigned int

Parameters **self** (*Spinnaker::PNGOption* *) –

interlaced
> PNGOption_interlaced_get(self) -> bool

> Parameters **self** (*Spinnaker::PNGOption* *) –

reserved
> PNGOption_reserved_get(self) -> unsigned int [16]

> Parameters **self** (*Spinnaker::PNGOption* *) –

thisown
> The membership flag

class PySpin.**PPMOption**
> Bases: object

> Options for saving PPM images.

> C++ includes: SpinnakerDefs.h

> **binaryFile**
> > PPMOption_binaryFile_get(self) -> bool

> > Parameters **self** (*Spinnaker::PPMOption* *) –

> **reserved**
> > PPMOption_reserved_get(self) -> unsigned int [16]

> > Parameters **self** (*Spinnaker::PPMOption* *) –

> **thisown**
> > The membership flag

class PySpin.**RegisterNode**(**args*, ***kwargs*)
> Bases: *PySpin.IRegister*, *PySpin.ValueNode*

> Interface for string properties.

> C++ includes: RegisterNode.h

> **Get** (*self*, *pBuffer*, *Verify=False*, *IgnoreCache=False*)

> > **Parameters**

> > - **pBuffer** (*uint8_t* *) –
> > - **Verify** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **pBuffer, Verify=False)** (Get(*self*,) –
> > - **pBuffer** –
> > - **Verify** –
> > - **pBuffer)** (Get(*self*,) –
> > - **pBuffer** –

> > virtual void Spinnaker::GenApi::RegisterNode::Get(uint8_t *pBuffer, int64_t Length, bool Verify=false, bool IgnoreCache=false)

> > Fills a buffer with the register's contents

> > pBuffer: The buffer receiving the data to read

Length: 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)

The value read

GetAddress (*self*) → int64_t

>> **Parameters self** (*Spinnaker::GenApi::RegisterNode **) –

virtual int64_t Spinnaker::GenApi::RegisterNode::GetAddress()

Retrieves the Address of the register

GetLength (*self*) → int64_t

>> **Parameters self** (*Spinnaker::GenApi::RegisterNode **) –

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*) –

>>> • **pBuffer)** (*Set(self,*) –

>>> • **pBuffer** –

virtual void Spinnaker::GenApi::RegisterNode::Set(const uint8_t *pBuffer, int64_t Length, bool Verify=true)

Set the register's contents

pBuffer: The buffer containing the data to set

Length: The number of bytes in pBuffer

Verify: Enables AccessMode and Range verification (default = true)

SetReference (*self*, *pBase*)

>> **Parameters pBase** (*Spinnaker::GenApi::INode **) –

virtual void Spinnaker::GenApi::RegisterNode::SetReference(INode *pBase)

overload SetReference for Register

thisown
>> The membership flag

PySpin.**RegisterNodeCallback** (*pNode*, *f*)

>> **Parameters**

>>> • **pNode** (*Spinnaker::GenApi::INode **) –

>>> • **f** (*NodeCallback &*) –

class PySpin.**RemovalEvent**
>> Bases: *PySpin.IRemovalEvent*

An event handler for capturing the device removal event.

C++ includes: RemovalEvent.h

OnDeviceRemoval(*self*, *serialNumber*)

>> **Parameters serialNumber** (`uint64_t`) –

> virtual void Spinnaker::RemovalEvent::OnDeviceRemoval(uint64_t serialNumber)=0

> Device removal event callback.

> serialNumber: The serial number of the device removed

thisown
> The membership flag

PySpin.**ReplaceEnvironmentVariables**(*Buffer*, *ReplaceBlankBy20=False*)

> **Parameters**

>> - **Buffer** (`Spinnaker::GenICam::gcstring &`) –
>> - **ReplaceBlankBy20** (`bool`) –
>> - **ReplaceEnvironmentVariables(Buffer)** –
>> - **Buffer** –

SPINNAKER_API void Spinnaker::GenICam::ReplaceEnvironmentVariables(gcstring &Buffer, bool Replace-BlankBy20=false)

Replaces in a string and replace ' ' with %20

PySpin.**SetGenICamCLProtocolFolder**(*path*)

> **Parameters path** (`Spinnaker::GenICam::gcstring const &`) –

SPINNAKER_API void Spinnaker::GenICam::SetGenICamCLProtocolFolder(const gcstring &path)

Stores the path of the CLProtocol folder

PySpin.**SetGenICamCacheFolder**(*path*)

> **Parameters path** (`Spinnaker::GenICam::gcstring const &`) –

SPINNAKER_API void Spinnaker::GenICam::SetGenICamCacheFolder(const gcstring &path)

Stores the path of the GenICam cache folder

PySpin.**SetGenICamLogConfig**(*path*)

> **Parameters path** (`Spinnaker::GenICam::gcstring const &`) –

SPINNAKER_API void Spinnaker::GenICam::SetGenICamLogConfig(const gcstring &path)

Stores the path of the GenICam logging properties file

PySpin.**SetMessageCallback**(*cb*)
Adds a callback to the updator to handle messages from the updator. 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.**SetProgressCallback**(*cb*)
Adds a callback to the updator 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.**SpinUpdate_SetMsgCallback**(*messageCallbackFunction*)

> Parameters **messageCallbackFunction** (*SpinUpdate::UpdatorMessageCallback*)
> –

PySpin.**SpinUpdate_SetProgCallback** (*progressCallbackFunction*)

> Parameters **progressCallbackFunction** (*SpinUpdate::UpdatorProgressCallback*)
> –

class PySpin.**StringNode** (**args*, ***kwargs*)

> Bases: *PySpin.IString*, *PySpin.ValueNode*
>
> Interface for string properties.
>
> C++ includes: StringNode.h
>
> **GetMaxLength** (*self*) → int64_t
>
> > Parameters **self** (*Spinnaker::GenApi::StringNode **) –
> >
> > 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** (*bool*) –
> > - **IgnoreCache** (*bool*) –
> > - **Verify=False) -> gcstring** (*GetValue (self,*) –
> > - **Verify** –
> > - **-> gcstring** (*GetValue (self)*) –
> > - **self** (*Spinnaker::GenApi::StringNode **) –
> >
> > virtual GenICam::gcstring Spinnaker::GenApi::StringNode::GetValue(bool Verify=false, bool Ignore-
> > Cache=false)
> >
> > Get node value
> >
> > Verify: Enables Range verification (default = false). The AccessMode is always checked
> >
> > IgnoreCache: If true the value is read ignoring any caches (default = false)
> >
> > The value read
>
> **SetReference** (*self*, *pBase*)
>
> > Parameters **pBase** (*Spinnaker::GenApi::INode **) –
> >
> > virtual void Spinnaker::GenApi::StringNode::SetReference(INode **pBase)
> >
> > overload SetReference for Value
>
> **SetValue** (*self*, *Value*, *Verify=True*)
>
> > Parameters
> >
> > - **Value** (*Spinnaker::GenICam::gcstring const &*) –
> > - **Verify** (*bool*) –
> > - **Value)** (*SetValue (self,*) –
> > - **Value** –

virtual void Spinnaker::GenApi::StringNode::SetValue(const GenICam::gcstring &Value, bool Verify=true)

Set node value

Value: The value to set

Verify: Enables AccessMode and Range verification (default = true)

thisown
> The membership flag

class PySpin.**StringRegNode**(*args*, ***kwargs*)
> Bases: *PySpin.StringNode*, *PySpin.RegisterNode*

Interface for string properties.

C++ includes: StringRegNode.h

SetReference(*self*, *pBase*)

> **Parameters pBase** (*Spinnaker::GenApi::INode **) –

virtual void Spinnaker::GenApi::StringRegNode::SetReference(INode *****pBase)

overload SetReference for Value

thisown
> The membership flag

class PySpin.**System**(*args*, ***kwargs*)
> Bases: *PySpin.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** (*bool*) –
> > - **updateCameras** (*bool*) –
> > - **updateInterfaces=True) -> CameraList** (*GetCameras(self,*) –
> > - **updateInterfaces** –
> > - **-> CameraList** (*GetCameras(self)*) –
> > - **self** (*Spinnaker::System **) –

CameraList Spinnaker::System::GetCameras(bool updateInterfaces=true, bool updateCameras=true)

Returns a list of cameras that are available on the system. This call returns both GigE Vision and Usb3 Vision cameras from all interfaces. The camera list object will reference count the cameras it returns. It is important that the camera list is destroyed or is cleared before calling system-> ReleaseInstance() or else the call to system-> ReleaseInstance() will result in an error message thrown that a reference to the camera is still held.

See: ReleaseInstance()

See: CameraList::Clear()

updateInterfaces: Determines whether or not updateInterfaceList() is called before getting cameras from available interfaces on the system

updateCameras: Determines whether or not UpdateCameras() is called before getting cameras from available interfaces on the system

An CameraList object that contains a list of all cameras.

static GetInstance() → SystemPtr

GetInterfaces(*self*, *updateInterface=True*) → InterfaceList

> **Parameters**
>
> > • **updateInterface**(*bool*) –
> >
> > • **-> InterfaceList**(GetInterfaces*(self)*) –
> >
> > • **self**(*Spinnaker::System **) –

InterfaceList Spinnaker::System::GetInterfaces(bool updateInterface=true)

Returns a list of interfaces available on the system. This call returns GigE and Usb2 and Usb3 interfaces.

updateInterface: Determines whether or not UpdateInterfaceList() is called before getting available interfaces

An InterfaceList object that contains a list of all interfaces.

GetLoggingEventPriorityLevel(*self*) → Spinnaker::SpinnakerLogLevel

> **Parameters self**(*Spinnaker::System **) –

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

IsInUse(*self*) → bool

> **Parameters self**(*Spinnaker::System **) –

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.

RegisterInterfaceEvent(*self*, *evtToRegister*, *updateInterface=True*)

> **Parameters**
>
> > • **evtToRegister**(*Spinnaker::Event &*) –
> >
> > • **updateInterface**(*bool*) –
> >
> > • **evtToRegister)**(RegisterInterfaceEvent*(self,)* –

- **evtToRegister** –

void Spinnaker::System::RegisterInterfaceEvent(Event &evtToRegister, bool updateInterface=true)

Registers events for all available interfaces that are found on the system

evtToRegister: The event to register for the available interfaces

updateInterface: Determines whether or not UpdateInterfaceList() is called before registering event for available interfaces on the system

RegisterLoggingEvent(*self*, *handler*)

> **Parameters handler** (*Spinnaker::LoggingEvent &*) –

void Spinnaker::System::RegisterLoggingEvent(LoggingEvent &handler)

Registers a logging event.

handler: The logging event handler to register

ReleaseInstance(*self*)

> **Parameters self** (*Spinnaker::System **) –

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*, *pResultSize=None*, *results=0*)

> **Parameters**
>
> - **deviceKey** (*unsigned int*) –
> - **groupKey** (*unsigned int*) –
> - **groupMask** (*unsigned int*) –
> - **actionTime** (*unsigned long long*) –
> - **pResultSize** (*unsigned int **) –
> - **results** (*Spinnaker::ActionCommandResult []*) –
> - **deviceKey, groupKey, groupMask, actionTime=0, pResultSize=None)** (*SendActionCommand(self,*) –
> - **deviceKey** –
> - **groupKey** –
> - **groupMask** –
> - **actionTime** –
> - **pResultSize** –
> - **deviceKey, groupKey, groupMask, actionTime=0)** (*SendActionCommand(self,*) –
> - **deviceKey** –

- **groupKey** –

- **groupMask** –

- **actionTime** –

- **deviceKey, groupKey, groupMask)** (SendActionCommand(self,) –

- **deviceKey** –

- **groupKey** –

- **groupMask** –

void Spinnaker::System::SendActionCommand(unsigned int deviceKey, unsigned int groupKey, unsigned int groupMask, unsigned long long actionTime=0, unsigned int *pResultSize=0, ActionCommandResult results[]=NULL)

Broadcast an Action Command to all devices on system

deviceKey: The Action Command's device key

groupKey: The Action Command's group key

groupMask: The Action Command's group mask

actionTime: (Optional) Time when to assert a future action. Zero means immediate action.

pResultSize: (Optional) The number of results in the results array. The value passed should be equal to the expected number of devices that acknowledge the command. Returns the number of received results.

results: (Optional) An Array with *pResultSize elements to hold the action command result status. The buffer is filled starting from index 0. If received results are less than expected number of devices that acknowledge the command, remaining results are not changed. If received results are more than expected number of devices that acknowledge the command, extra results are ignored and not appended to array. This parameter is ignored if pResultSize is 0. Thus this parameter can be NULL if pResultSize is 0 or NULL.

SetLoggingEventPriorityLevel(*self*, *level*)

> **Parameters level** (*enum Spinnaker::SpinnakerLogLevel*) –

void Spinnaker::System::SetLoggingEventPriorityLevel(SpinnakerLogLevel level)

Sets a threshold priority level for logging event. Logging events below such level will not trigger callbacks.

Spinnaker uses five levels of logging: Error - failures that are non- recoverable without user intervention.

Warning - failures that are recoverable without user intervention.

Notice - information about events such as camera arrival and removal, initialization and deinitialization, starting and stopping image acquisition, and feature modification.

Info - information about recurring events that are generated regularly such as information on individual images.

Debug - information that can be used to troubleshoot the system.

See: SpinnakerLogLevel

level: The threshold level

UnregisterAllLoggingEvent(*self*)

> **Parameters self** (*Spinnaker::System **) –

void Spinnaker::System::UnregisterAllLoggingEvent()

Unregisters all previously registered logging events.

UnregisterInterfaceEvent(*self*, *evtToUnregister*)

> Parameters **evtToUnregister** (*Spinnaker::Event &*) –

void Spinnaker::System::UnregisterInterfaceEvent(Event &evtToUnregister)

Unregisters events for all available interfaces that are found on the system

evtToUnregister: The event to unregister from the available interfaces

UnregisterLoggingEvent(*self*, *handler*)

> Parameters **handler** (*Spinnaker::LoggingEvent &*) –

void Spinnaker::System::UnregisterLoggingEvent(LoggingEvent &handler)

Unregisters a logging event.

handler: The logging event handler to unregister

UpdateCameras(*self*, *updateInterfaces=True*) → bool

> Parameters
>
> - **updateInterfaces** (*bool*) –
> - **-> bool** (*UpdateCameras(self)*) –
> - **self** (*Spinnaker::System **) –

bool Spinnaker::System::UpdateCameras(bool updateInterfaces=true)

Updates the list of cameras on the system. Note that System::GetCameras() internally calls UpdateCameras() for each interface it enumerates. If the list changed between this call and the last time UpdateCameras was called then the return value will be true, otherwise it is false.

See: GetCameras()

updateInterfaces: Determines whether or not UpdateInterfaceList() is called before updating cameras for available interfaces on the system

True if cameras changed on interface and false otherwise.

thisown
    The membership flag

class PySpin.**SystemPtr**(**args*)
    Bases: PySpin._SWIG_SysPtr

A reference tracked pointer to a system object.

C++ includes: SystemPtr.h

thisown
    The membership flag

PySpin.**System_GetInstance**() → SystemPtr

class PySpin.**TIFFOption**
    Bases: object

Options for saving TIFF images.

C++ includes: SpinnakerDefs.h

ADOBE_DEFLATE = 4

CCITTFAX3 = 5

CCITTFAX4 = 6

DEFLATE = 3

JPEG = 8

LZW = 7

NONE = 1

PACKBITS = 2

compression
    TIFFOption_compression_get(self) -> Spinnaker::TIFFOption::CompressionMethod

        Parameters self (*Spinnaker::TIFFOption **) –

reserved
    TIFFOption_reserved_get(self) -> unsigned int [16]

        Parameters self (*Spinnaker::TIFFOption **) –

thisown
    The membership flag

PySpin.**ThrowBadAlloc**()
    SPINNAKER_API void Spinnaker::GenICam::ThrowBadAlloc()

PySpin.**Tokenize**(*str*, *delimiters*)

    Parameters

        • **str** (*Spinnaker::GenICam::gcstring const &*) –

        • **delimiters** (*Spinnaker::GenICam::gcstring const &*) –

        • **Tokenize(str)** –

        • **str** –

    SPINNAKER_API void Spinnaker::GenICam::Tokenize(const gcstring &str, gcstring_vector &tokens, const gcstring &delimiters=" ")

    splits str input string into a list of tokens using the delimiter

class 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

    DeviceAccessStatus
        TransportLayerDevice_DeviceAccessStatus_get(self) -> IEnumerationT_DeviceAccessStatusEnum

            Parameters self (*Spinnaker::TransportLayerDevice **) –

    DeviceCurrentSpeed
        TransportLayerDevice_DeviceCurrentSpeed_get(self) -> IEnumerationT_DeviceCurrentSpeedEnum

            Parameters self (*Spinnaker::TransportLayerDevice **) –

    DeviceDisplayName
        TransportLayerDevice_DeviceDisplayName_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceDriverVersion
TransportLayerDevice_DeviceDriverVersion_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceEndianessMechanism
TransportLayerDevice_DeviceEndianessMechanism_get(self)   -> IEnumera-
tionT_DeviceEndianessMechanismEnum

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceID
TransportLayerDevice_DeviceID_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceInstanceId
TransportLayerDevice_DeviceInstanceId_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceLinkSpeed
TransportLayerDevice_DeviceLinkSpeed_get(self) -> IInteger

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceModelName
TransportLayerDevice_DeviceModelName_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceMulticastMonitorMode
TransportLayerDevice_DeviceMulticastMonitorMode_get(self) -> IBoolean

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceSerialNumber
TransportLayerDevice_DeviceSerialNumber_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceType
TransportLayerDevice_DeviceType_get(self) -> IEnumerationT_DeviceTypeEnum

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceUserID
TransportLayerDevice_DeviceUserID_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceVendorName
TransportLayerDevice_DeviceVendorName_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

DeviceVersion
TransportLayerDevice_DeviceVersion_get(self) -> IString

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

GUIXMLLocation
TransportLayerDevice_GUIXMLLocation_get(self) -> IEnumerationT_GUIXMLLocationEnum

Parameters **self** (*Spinnaker::TransportLayerDevice* *) –

GUIXMLPath
　　TransportLayerDevice_GUIXMLPath_get(self) -> IString

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GenICamXMLLocation
　　TransportLayerDevice_GenICamXMLLocation_get(self)　　　　-> 　　　　IEnumera-
　　tionT_GenICamXMLLocationEnum

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GenICamXMLPath
　　TransportLayerDevice_GenICamXMLPath_get(self) -> IString

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevCCP
　　TransportLayerDevice_GevCCP_get(self) -> IEnumerationT_GevCCPEnum

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceDiscoverMaximumPacketSize
　　TransportLayerDevice_GevDeviceDiscoverMaximumPacketSize_get(self) -> ICommand

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceGateway
　　TransportLayerDevice_GevDeviceGateway_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceIPAddress
　　TransportLayerDevice_GevDeviceIPAddress_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceMACAddress
　　TransportLayerDevice_GevDeviceMACAddress_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceMaximumPacketSize
　　TransportLayerDevice_GevDeviceMaximumPacketSize_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceMaximumRetryCount
　　TransportLayerDevice_GevDeviceMaximumRetryCount_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceModeIsBigEndian
　　TransportLayerDevice_GevDeviceModeIsBigEndian_get(self) -> IBoolean

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDevicePort
　　TransportLayerDevice_GevDevicePort_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceReadAndWriteTimeout
　　TransportLayerDevice_GevDeviceReadAndWriteTimeout_get(self) -> IInteger

　　　　Parameters self (*Spinnaker::TransportLayerDevice **) –

GevDeviceSubnetMask
> TransportLayerDevice_GevDeviceSubnetMask_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevVersionMajor
> TransportLayerDevice_GevVersionMajor_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerDevice **) –

GevVersionMinor
> TransportLayerDevice_GevVersionMinor_get(self) -> IInteger

>> **Parameters self** (*Spinnaker::TransportLayerDevice **) –

thisown
> The membership flag

class 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

> **ActionCommand**
> > TransportLayerInterface_ActionCommand_get(self) -> ICommand

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **AutoForceIP**
> > TransportLayerInterface_AutoForceIP_get(self) -> ICommand

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceAccessStatus**
> > TransportLayerInterface_DeviceAccessStatus_get(self) -> IEnumerationT_DeviceAccessStatusEnum

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceCount**
> > TransportLayerInterface_DeviceCount_get(self) -> IInteger

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceID**
> > TransportLayerInterface_DeviceID_get(self) -> IString

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceModelName**
> > TransportLayerInterface_DeviceModelName_get(self) -> IString

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceSelector**
> > TransportLayerInterface_DeviceSelector_get(self) -> IInteger

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceUnlock**
> > TransportLayerInterface_DeviceUnlock_get(self) -> IString

> > > **Parameters self** (*Spinnaker::TransportLayerInterface **) –

> **DeviceUpdateList**
> > TransportLayerInterface_DeviceUpdateList_get(self) -> ICommand

> Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

DeviceVendorName
> TransportLayerInterface_DeviceVendorName_get(self) -> IString

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevActionDeviceKey
> TransportLayerInterface_GevActionDeviceKey_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevActionGroupKey
> TransportLayerInterface_GevActionGroupKey_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevActionGroupMask
> TransportLayerInterface_GevActionGroupMask_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevActionTime
> TransportLayerInterface_GevActionTime_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevDeviceIPAddress
> TransportLayerInterface_GevDeviceIPAddress_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevDeviceMACAddress
> TransportLayerInterface_GevDeviceMACAddress_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevDeviceSubnetMask
> TransportLayerInterface_GevDeviceSubnetMask_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevInterfaceGateway
> TransportLayerInterface_GevInterfaceGateway_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevInterfaceIPAddress
> TransportLayerInterface_GevInterfaceIPAddress_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevInterfaceMACAddress
> TransportLayerInterface_GevInterfaceMACAddress_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

GevInterfaceSubnetMask
> TransportLayerInterface_GevInterfaceSubnetMask_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

IncompatibleDeviceCount
> TransportLayerInterface_IncompatibleDeviceCount_get(self) -> IInteger

> > Parameters **self** (*Spinnaker::TransportLayerInterface* *) –

IncompatibleDeviceID
    TransportLayerInterface_IncompatibleDeviceID_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceModelName
    TransportLayerInterface_IncompatibleDeviceModelName_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceSelector
    TransportLayerInterface_IncompatibleDeviceSelector_get(self) -> IInteger

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

IncompatibleDeviceVendorName
    TransportLayerInterface_IncompatibleDeviceVendorName_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceDisplayName
    TransportLayerInterface_InterfaceDisplayName_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceID
    TransportLayerInterface_InterfaceID_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

InterfaceType
    TransportLayerInterface_InterfaceType_get(self) -> IString

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

POEStatus
    TransportLayerInterface_POEStatus_get(self) -> IEnumerationT_POEStatusEnum

        Parameters **self** (*Spinnaker::TransportLayerInterface **) –

thisown
    The membership flag

class 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

GevFailedPacketCount
    TransportLayerStream_GevFailedPacketCount_get(self) -> IInteger

        Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevMaximumNumberResendBuffers
    TransportLayerStream_GevMaximumNumberResendBuffers_get(self) -> IInteger

        Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevMaximumNumberResendRequests
    TransportLayerStream_GevMaximumNumberResendRequests_get(self) -> IInteger

        Parameters **self** (*Spinnaker::TransportLayerStream **) –

GevPacketResendMode
    TransportLayerStream_GevPacketResendMode_get(self) -> IBoolean

Parameters self (*Spinnaker::TransportLayerStream **) –

GevPacketResendTimeout
TransportLayerStream_GevPacketResendTimeout_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

GevResendPacketCount
TransportLayerStream_GevResendPacketCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

GevResendRequestCount
TransportLayerStream_GevResendRequestCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

GevTotalPacketCount
TransportLayerStream_GevTotalPacketCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamBlockTransferSize
TransportLayerStream_StreamBlockTransferSize_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamBufferHandlingMode
TransportLayerStream_StreamBufferHandlingMode_get(self) -> IEnumerationT_StreamBufferHandlingModeEnum

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamBufferUnderrunCount
TransportLayerStream_StreamBufferUnderrunCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamCRCCheckEnable
TransportLayerStream_StreamCRCCheckEnable_get(self) -> IBoolean

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamDefaultBufferCount
TransportLayerStream_StreamDefaultBufferCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamDefaultBufferCountMax
TransportLayerStream_StreamDefaultBufferCountMax_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamDefaultBufferCountMode
TransportLayerStream_StreamDefaultBufferCountMode_get(self) -> IEnumerationT_StreamDefaultBufferCountModeEnum

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamFailedBufferCount
TransportLayerStream_StreamFailedBufferCount_get(self) -> IInteger

Parameters self (*Spinnaker::TransportLayerStream **) –

StreamID
TransportLayerStream_StreamID_get(self) -> IString

>
> Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamTotalBufferCount
> TransportLayerStream_StreamTotalBufferCount_get(self) -> IInteger
>
> > Parameters **self** (*Spinnaker::TransportLayerStream* *) –

StreamType
> TransportLayerStream_StreamType_get(self) -> IEnumerationT_StreamTypeEnum
>
> > Parameters **self** (*Spinnaker::TransportLayerStream* *) –

thisown
> The membership flag

PySpin.**UpdateFirmware**(*args*) → int

> > Parameters **args** (*char const* *) –

PySpin.**UpdateFirmwareConsole**(*argc*) → int

> > Parameters **argc** (*int*) –

PySpin.**UrlDecode**(*Input*) → gcstring

> > Parameters **Input** (*Spinnaker::GenICam::gcstring const &*) –
>
> SPINNAKER_API gcstring Spinnaker::GenICam::UrlDecode(const gcstring &Input)
>
> Replaces xx escapes by their char equivalent

PySpin.**UrlEncode**(*Input*) → gcstring

> > Parameters **Input** (*Spinnaker::GenICam::gcstring const &*) –
>
> SPINNAKER_API gcstring Spinnaker::GenICam::UrlEncode(const gcstring &Input)
>
> Converts to / and replaces all unsave characters by their xx equivalent

class PySpin.**ValueNode**(**args*, ***kwargs*)
> Bases: [`PySpin.IValue`](#), [`PySpin.Node`](#)
>
> Interface for value properties.
>
> C++ includes: ValueNode.h
>
> **FromString**(*self*, *ValueStr*, *Verify=True*)
>
> > Parameters
> >
> > - **ValueStr** (*Spinnaker::GenICam::gcstring const &*) –
> > - **Verify** (*bool*) –
> > - **ValueStr)** ([`FromString`](#)*(self,)*) –
> > - **ValueStr** –
> >
> > virtual void Spinnaker::GenApi::ValueNode::FromString(const GenICam::gcstring &ValueStr, bool Verify=true)
> >
> > Set content of the node as string
> >
> > ValueStr: The value to set
> >
> > Verify: Enables AccessMode and Range verification (default = true)
>
> **GetNode**(*self*) → INode
>
> > Parameters **self** (*Spinnaker::GenApi::ValueNode* *) –

> virtual INode* Spinnaker::GenApi::ValueNode::GetNode()

IsValueCacheValid(*self*) → bool

> > **Parameters self**(*Spinnaker::GenApi::ValueNode const **) –
>
> 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 **) –
>
> virtual void Spinnaker::GenApi::ValueNode::SetReference(INode *pBase)
>
> overload SetReference for Value

ToString(*self*, *Verify=False*, *IgnoreCache=False*) → gcstring

> > **Parameters**
> >
> > > - **Verify**(*bool*) –
> > > - **IgnoreCache**(*bool*) –
> > > - **Verify=False) -> gcstring**(`ToString(self,)`) –
> > > - **Verify** –
> > > - **-> gcstring**(`ToString(self)`) –
> > > - **self**(*Spinnaker::GenApi::ValueNode **) –
>
> virtual GenICam::gcstring Spinnaker::GenApi::ValueNode::ToString(bool Verify=false, bool Ignore-Cache=false)
>
> Get content of the node as string
>
> Verify: Enables Range verification (default = false). The AccessMode is always checked
>
> IgnoreCache: If true the value is read ignoring any caches (default = false)
>
> The value read

thisown
> The membership flag

class PySpin.**Version_t**
> Bases: object
>
> Version
>
> C++ includes: GCTypes.h
>
> **Major**
> > Version_t_Major_get(self) -> uint16_t
> >
> > > **Parameters self**(*Spinnaker::GenICam::Version_t **) –
>
> **Minor**
> > Version_t_Minor_get(self) -> uint16_t
> >
> > > **Parameters self**(*Spinnaker::GenICam::Version_t **) –
>
> **SubMinor**
> > Version_t_SubMinor_get(self) -> uint16_t
> >
> > > **Parameters self**(*Spinnaker::GenICam::Version_t **) –

thisown
> The membership flag

class `PySpin.`**`double_autovector_t`**`(*args)`
> Bases: `object`

Vector of doubles with reference counting.

C++ includes: Autovector.h

`size``(self)` → size_t

> > **Parameters** **self** (`Spinnaker::GenApi::double_autovector_t const *`) –

> size_t Spinnaker::GenApi::double_autovector_t::size() const

thisown
> The membership flag

class `PySpin.`**`gcstring`**`(*args)`
> Bases: `object`

Proxy of C++ Spinnaker::GenICam::gcstring class.

`append``(self, str)` → gcstring

> **Parameters**

> > - **str** (`Spinnaker::GenICam::gcstring const &`) –
> > - **count, ch) -> gcstring** (`append(self,`) –
> > - **count** (`size_t`) –
> > - **ch** (`char`) –

> virtual gcstring& Spinnaker::GenICam::gcstring::append(size_t count, char ch)

`assign``(self, str)` → gcstring

> **Parameters**

> > - **str** (`Spinnaker::GenICam::gcstring const &`) –
> > - **count, ch) -> gcstring** (`assign(self,`) –
> > - **count** (`size_t`) –
> > - **ch** (`char`) –
> > - **pc) -> gcstring** (`assign(self,`) –
> > - **pc** (`char const *`) –
> > - **pc, n) -> gcstring** (`assign(self,`) –
> > - **pc** –
> > - **n** (`size_t`) –

> virtual gcstring& Spinnaker::GenICam::gcstring::assign(const char *pc, size_t n)

`c_str``(self)` → char const *

> > **Parameters** **self** (`Spinnaker::GenICam::gcstring const *`) –

> virtual const char* Spinnaker::GenICam::gcstring::c_str(void) const

`compare``(self, str)` → int

> > > Parameters **str** (*Spinnaker::GenICam::gcstring const &*) –

> > virtual int Spinnaker::GenICam::gcstring::compare(const gcstring &str) const

> **empty** (*self*) → bool

> > > Parameters **self** (*Spinnaker::GenICam::gcstring const **) –

> > virtual bool Spinnaker::GenICam::gcstring::empty(void) const

> **find** (*self*, *ch*, *offset=0*) → size_t

> > **Parameters**

> > > - **ch** (*char*) –
> > > - **offset** (*size_t*) –
> > > - **ch) -> size_t** (find *(self,)* –
> > > - **ch** –
> > > - **str, offset=0) -> size_t** (find *(self,)* –
> > > - **str** (*Spinnaker::GenICam::gcstring const &*) –
> > > - **offset** –
> > > - **str) -> size_t** (find *(self,)* –
> > > - **str** –
> > > - **str, offset, count) -> size_t** (find *(self,)* –
> > > - **str** –
> > > - **offset** –
> > > - **count** (*size_t*) –
> > > - **pc, offset=0) -> size_t** (find *(self,)* –
> > > - **pc** (*char const **) –
> > > - **offset** –
> > > - **pc) -> size_t** (find *(self,)* –
> > > - **pc** –
> > > - **pc, offset, count) -> size_t** (find *(self,)* –
> > > - **pc** –
> > > - **offset** –
> > > - **count** –

> > virtual size_t Spinnaker::GenICam::gcstring::find(const char *pc, size_t offset, size_t count) const

> **find_first_not_of** (*self*, *str*, *offset=0*) → size_t

> > **Parameters**

> > > - **str** (*Spinnaker::GenICam::gcstring const &*) –
> > > - **offset** (*size_t*) –
> > > - **str) -> size_t** (find_first_not_of *(self,)* –
> > > - **str** –

virtual size_t Spinnaker::GenICam::gcstring::find_first_not_of(const gcstring &str, size_t offset=0) const

find_first_of(*self*, *str*, *offset=0*) → size_t

> **Parameters**

>> • **str** (*Spinnaker::GenICam::gcstring const &*) –

>> • **offset** (*size_t*) –

>> • **str) -> size_t** ([find_first_of](*self,*)) –

>> • **str** –

> virtual size_t Spinnaker::GenICam::gcstring::find_first_of(const gcstring &str, size_t offset=0) const

length(*self*) → size_t

> **Parameters self** (*Spinnaker::GenICam::gcstring const **) –

> virtual size_t Spinnaker::GenICam::gcstring::length(void) const

max_size(*self*) → size_t

> **Parameters self** (*Spinnaker::GenICam::gcstring const **) –

> virtual size_t Spinnaker::GenICam::gcstring::max_size() const

npos = 18446744073709551615

resize(*self*, *n*)

> **Parameters n** (*size_t*) –

> virtual void Spinnaker::GenICam::gcstring::resize(size_t n)

size(*self*) → size_t

> **Parameters self** (*Spinnaker::GenICam::gcstring const **) –

> virtual size_t Spinnaker::GenICam::gcstring::size(void) const

substr(*self*, *offset=0*, *count*) → gcstring

> **Parameters**

>> • **offset** (*size_t*) –

>> • **count** (*size_t*) –

>> • **offset=0) -> gcstring** ([substr](*self,*)) –

>> • **offset** –

>> • **-> gcstring** ([substr](*self*)) –

>> • **self** (*Spinnaker::GenICam::gcstring const **) –

> virtual gcstring Spinnaker::GenICam::gcstring::substr(size_t offset=0, size_t count=GCSTRING_NPOS) const

swap(*self*, *Right*)

> **Parameters Right** (*Spinnaker::GenICam::gcstring &*) –

> virtual void Spinnaker::GenICam::gcstring::swap(gcstring &Right)

thisown
> The membership flag

PySpin.**gcstring__npos**() → size_t

class PySpin.**int64_autovector_t**(**args*)

>   Bases: `object`
>
>   Vector of integers with reference counting.
>
>   C++ includes: Autovector.h
>
>   **size**(*self*) → size_t
>
> >   **Parameters self** (*Spinnaker::GenApi::int64_autovector_t const **) –
> >
> >   size_t Spinnaker::GenApi::int64_autovector_t::size() const
>
>   **thisown**
> >   The membership flag

class 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*) –
> > >
> > >   • **uiIndex) -> INode** (*at(self,*) –
> > >
> > >   • **uiIndex** –
>
>   **back**(*self*) → INode
> >   back(self) -> INode
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector const **) –
>
>   **begin**(*self*) → Spinnaker::GenApi::node_vector::iterator
> >   begin(self) -> Spinnaker::GenApi::node_vector::const_iterator
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector const **) –
>
>   **capacity**(*self*) → size_t
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector const **) –
>
>   **clear**(*self*)
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector **) –
>
>   **empty**(*self*) → bool
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector const **) –
>
>   **end**(*self*) → Spinnaker::GenApi::node_vector::iterator
> >   end(self) -> Spinnaker::GenApi::node_vector::const_iterator
>
> >   **Parameters self** (*Spinnaker::GenApi::node_vector const **) –
>
>   **erase**(*self*, *pos*) → Spinnaker::GenApi::node_vector::iterator
>
> >   **Parameters**

- **pos** (*Spinnaker::GenApi::node_vector::iterator*) –

- **uiIndex)** ([erase (self,)](#)) –

- **uiIndex** (*size_t*) –

front (*self*) → INode
> front(self) -> INode

>> **Parameters self** (*Spinnaker::GenApi::node_vector const **) –

insert (*self*, *pos*, *val*) → Spinnaker::GenApi::node_vector::iterator

> **Parameters**

- **pos** (*Spinnaker::GenApi::node_vector::iterator*) –

- **val** (*Spinnaker::GenApi::node_vector::T const &*) –

- **uiIndex, val)** ([insert (self,)](#)) –

- **uiIndex** (*size_t*) –

- **val** –

max_size (*self*) → size_t

>> **Parameters self** (*Spinnaker::GenApi::node_vector const **) –

pop_back (*self*)

>> **Parameters self** (*Spinnaker::GenApi::node_vector **) –

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

>> **Parameters self** (*Spinnaker::GenApi::node_vector const **) –

thisown
> The membership flag

class 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*) –

- **uiIndex) -> IValue** (at(*self,*)) –

- **uiIndex** –

back (*self*) → IValue
> back(self) -> IValue

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

begin (*self*) → Spinnaker::GenApi::value_vector::iterator
> begin(self) -> Spinnaker::GenApi::value_vector::const_iterator

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

capacity (*self*) → size_t

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

clear (*self*)

> > **Parameters self** (*Spinnaker::GenApi::value_vector **) –

empty (*self*) → bool

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

end (*self*) → Spinnaker::GenApi::value_vector::iterator
> end(self) -> Spinnaker::GenApi::value_vector::const_iterator

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

erase (*self*, *pos*) → Spinnaker::GenApi::value_vector::iterator

> > **Parameters**

> > > - **pos** (*Spinnaker::GenApi::value_vector::iterator*) –

> > > - **uiIndex)** (erase(*self,*)) –

> > > - **uiIndex** (*size_t*) –

front (*self*) → IValue
> front(self) -> IValue

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

insert (*self*, *pos*, *val*) → Spinnaker::GenApi::value_vector::iterator

> > **Parameters**

> > > - **pos** (*Spinnaker::GenApi::value_vector::iterator*) –

> > > - **val** (*Spinnaker::GenApi::value_vector::T const &*) –

> > > - **uiIndex, val)** (insert(*self,*)) –

> > > - **uiIndex** (*size_t*) –

> > > - **val** –

max_size (*self*) → size_t

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

pop_back (*self*)

> > **Parameters self** (*Spinnaker::GenApi::value_vector **) –

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

> > **Parameters self** (*Spinnaker::GenApi::value_vector const **) –

thisown
  The membership flag

PYTHON MODULE INDEX

p

H

I

W